Anti-Copying Copy Protector ; proposal

Information Security

DoItLater Leader : WonJin Yoon, SeoungYoon Kim, ChangMin Choi, SunJae Kim, BuRu Jang, TaeSeong Kim, WonTae Jeong, SuYeon Lee

Table of Contents

Motivation

Concept of our project

Technical explanation

Roles and timeline

Motivation

- Let's think of industrial Spies… or someone who leaks personal information
- People tries such a many ways to make leakage impossible. Like strict entrance policies or encryption

HOWEVER….

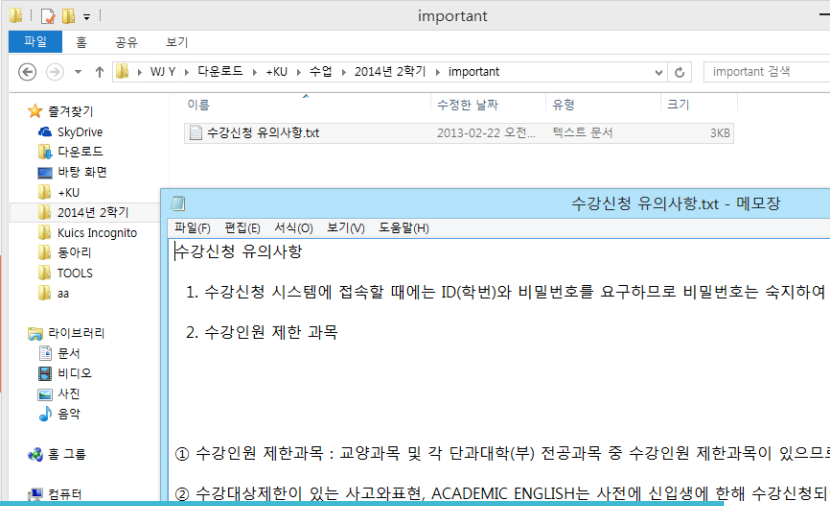
M

Motivation

HOWEVER….

- USBs are getting smaller and harder to be physically detected (like cufflinks USB or Card-style USB…)
- Encryption can be nice solution for information leakage, but it is of no use to inside betrayal.
- So we started to think from the very base of file leakage

“Anti – copying”



What is "Copy Protector"?

- Think of situation, that internal betrayer (maybe industrial spy) tries to copy important file
- First, malicious user access to file and opens it to figure out whether it is the file he intended.
- "Copy Protector" detects the opening of important file, but he does not touch anything so that the user can verify the file

What is "Copy Protector"?

- Then, the user tries to copy file, and it seems it is copied correctly
- But, "Copy Protector" intervene the copying process and prevent copying and make dummy file instead.
- Also malicious user cannot terminate "Copy Protector" due to the self-protection, which makes it impossible to be terminated without proper authentication.

Analysis on technique

- We will hook SSDT to detect opening the important file, to interrupt copying (writing) and to prevent "Copy Protector" process from closing

- Also we will use GUI for selecting the file to be protected and for authenticated copying

Role

Function Team

Make functions of
* Detecting reading
* Hooking writing

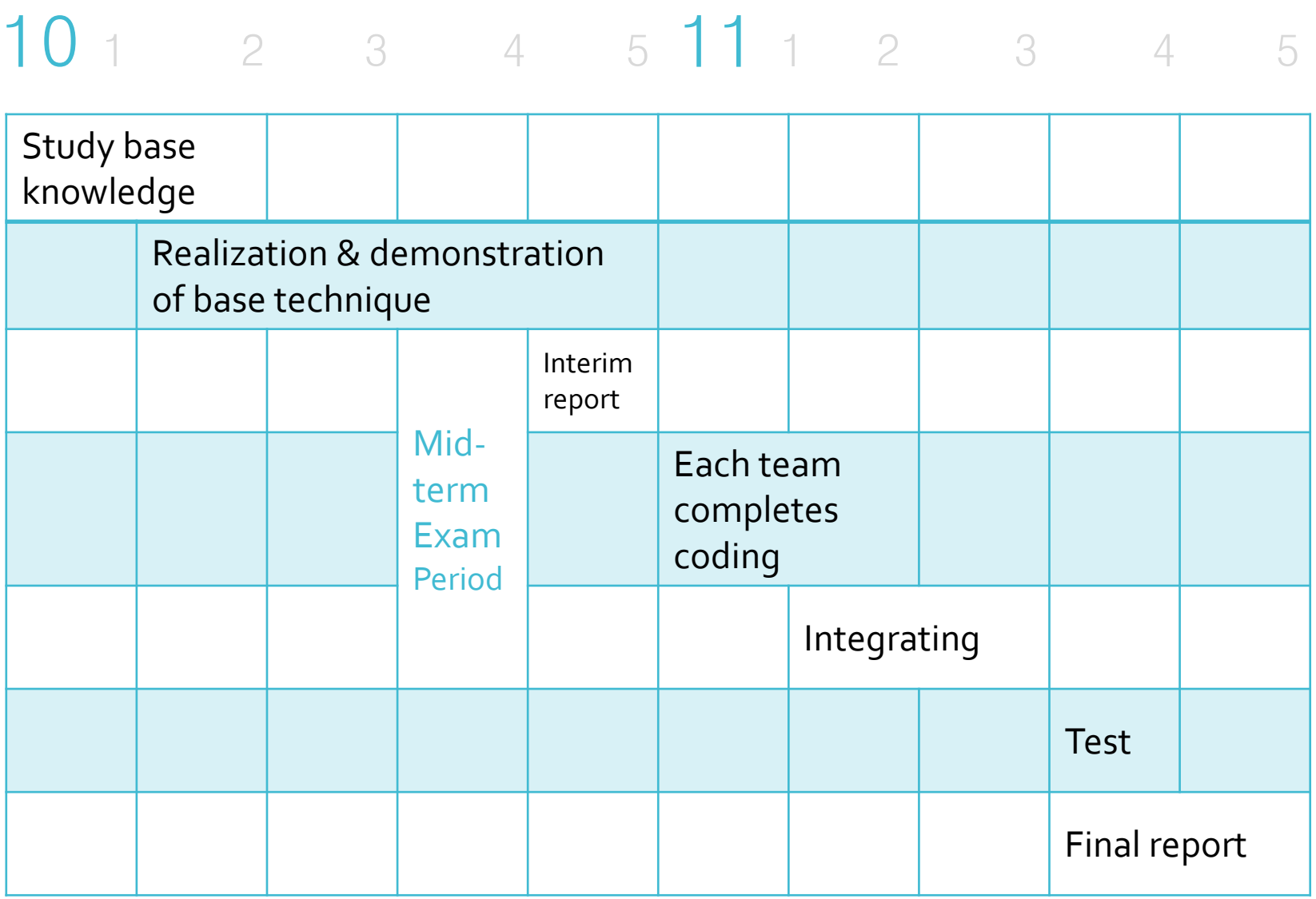
윤원진, 김승윤,
최창민, 김순재

Process team

GUI
Process Protection
(Anti-closing)

정원태, 이수연,
장부루, 김태승

Timeline


10
1
2
3
4
5
11
1
2
3
4
5
Study base knowledge
Realization & demonstration of base technique
Interim report
Mid-term Exam Period
Each team completes coding
Integrating
Test
Final report

End of proposal

- Thank you for your attention

Anti-Copying Copy Protector ; interim

Information Security

DoItLater Leader : WonJin Yoon, SeoungYoon Kim, ChangMin Choi, SunJae Kim, BuRu Jang, TaeSeong Kim, WonTae Jeong, SuYeon Lee

Brief Review Concept

- We wants to prevent important files from being copied with out permission.
- We wants to fake malicious user(especially internal betrayer) as file being copied correctly, so that he can't realize our program on site of criminal.
  Consequently, he does not even try to evade it.

Function Team

- From now, we will report Function Team

Function Team

Make functions of
* Detecting reading
* Hooking writing

윤원진, 김승윤,
최창민, 김순재

Process team

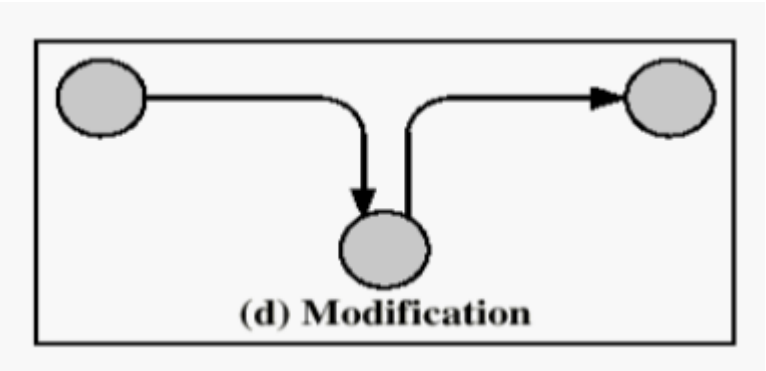
GUI
Process Protection
(Anti-closing)

정원태, 이수연,
장부루, 김태승

How can we realize (Techniques)

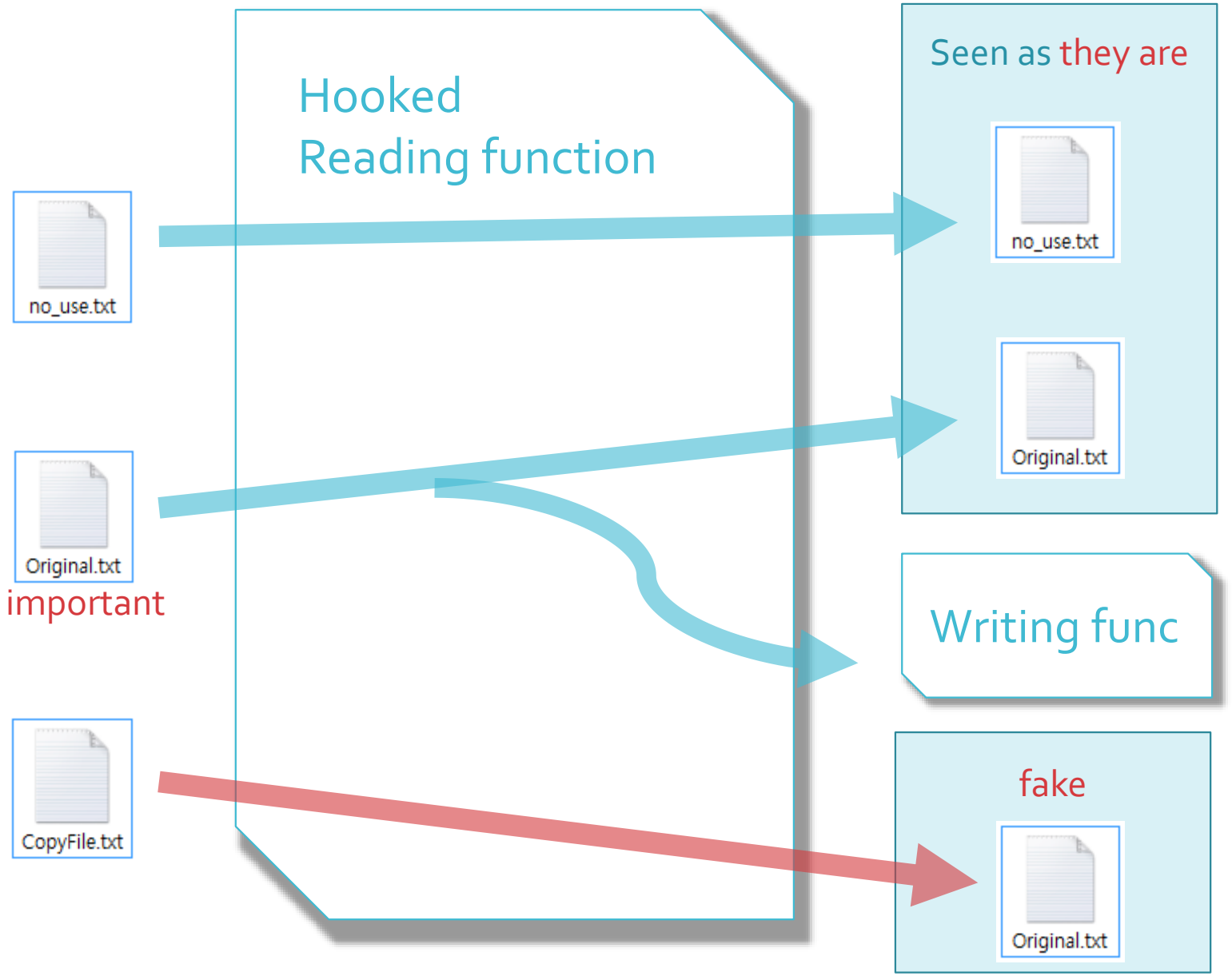
- We hooked SSDT and manipulate reading and writing process

*SSDT : System Service Descriptor Table.
A table that contains addresses of internal functions



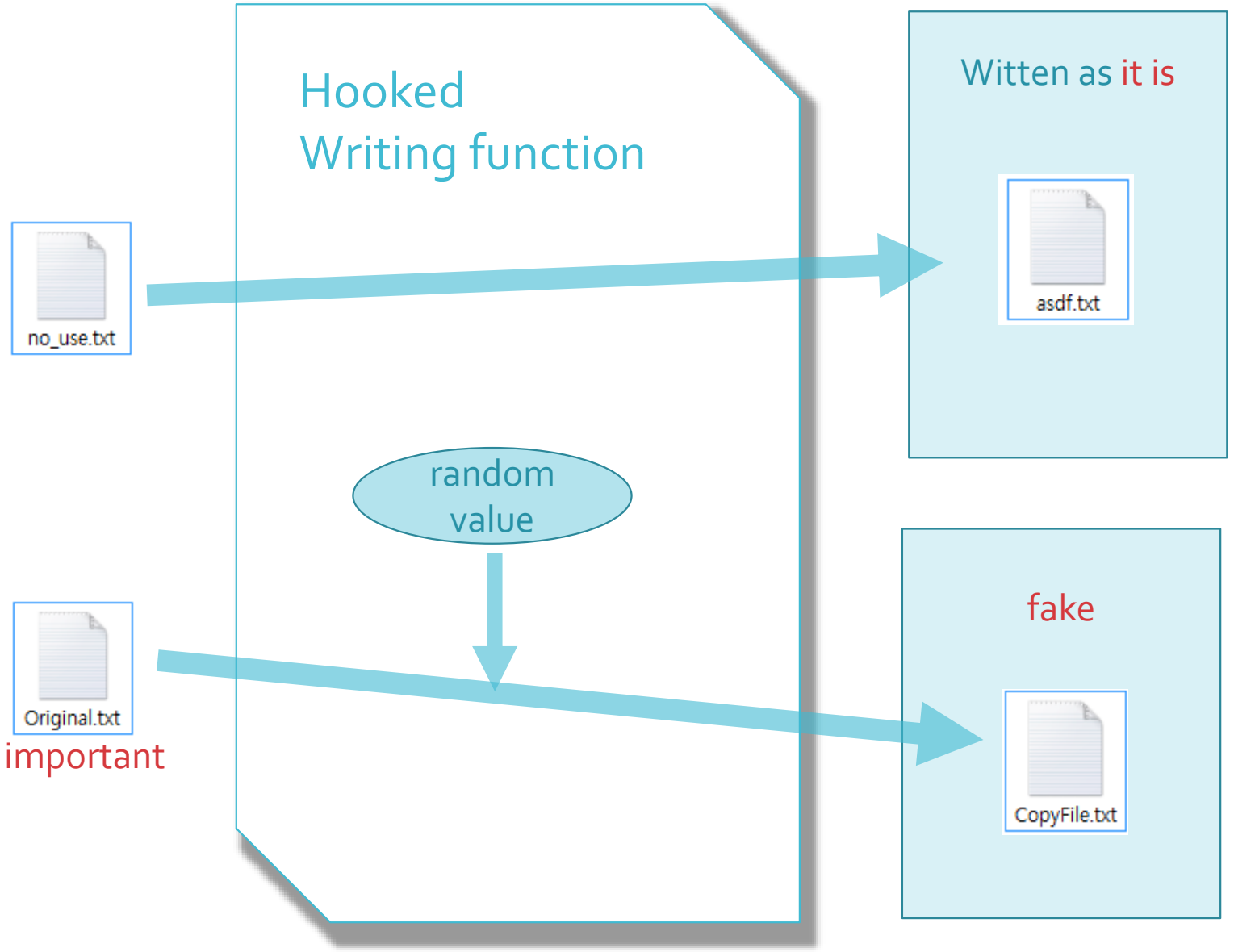
How can we realize (Techniques)

Reading



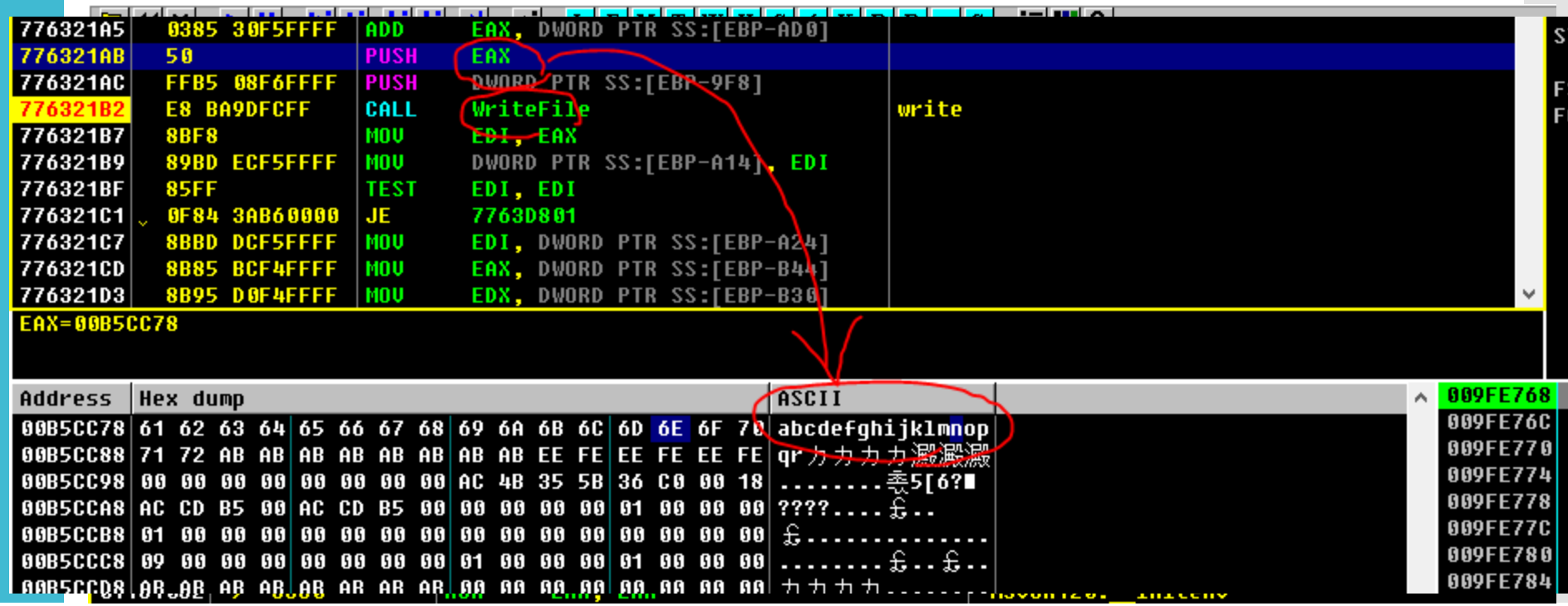
How can we realize (Techniques)

Writing



Technical Process Reverse engineering

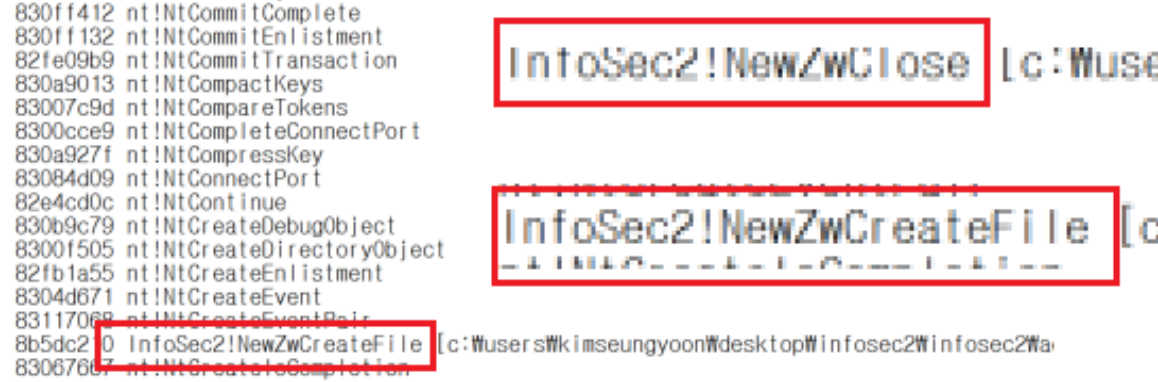
- We reverse engineered copy function to get full understanding of copying process.
- We step into and into until we can see the surface of Kernel mode.



Demonstration (SSDT Hooking)

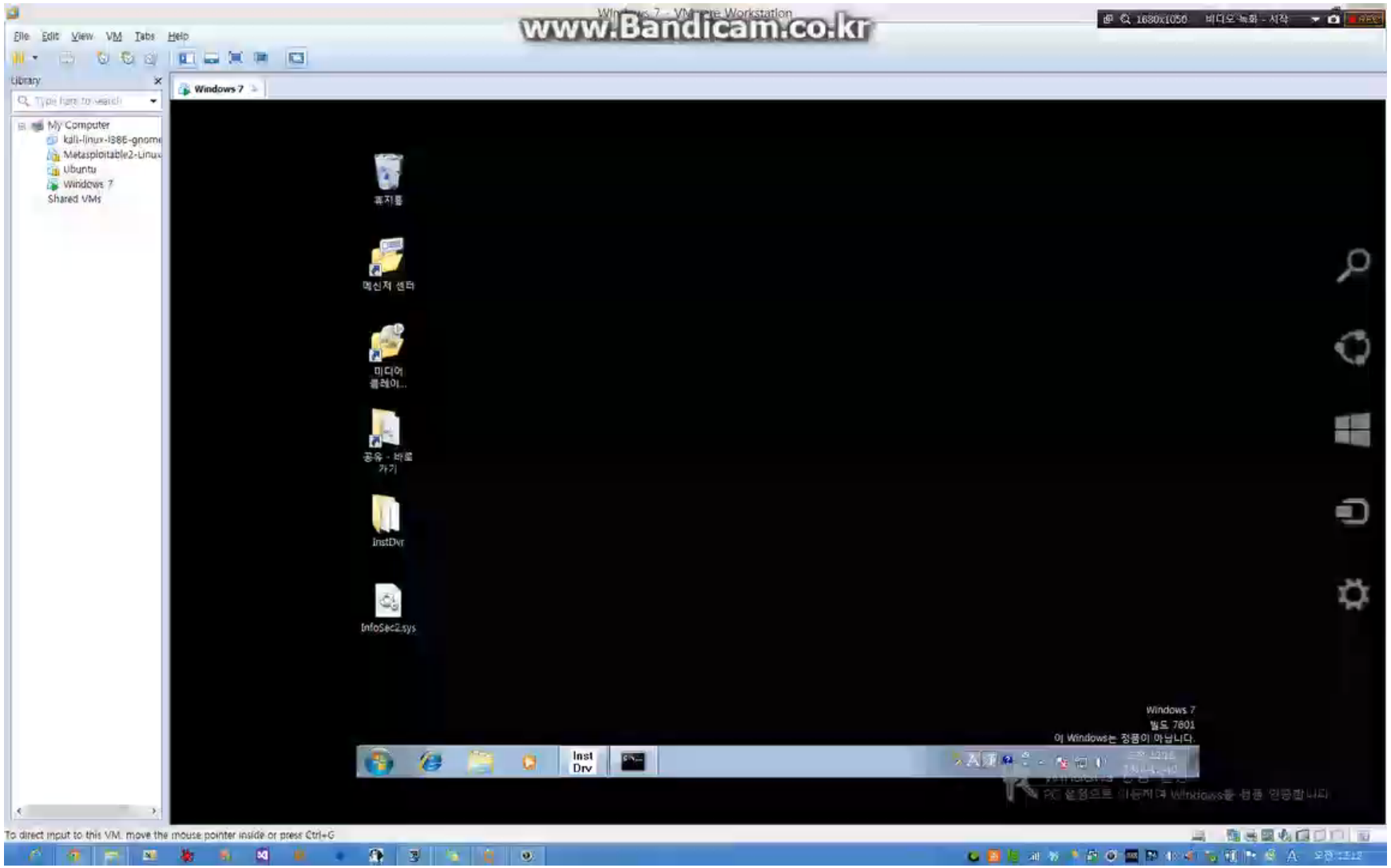
- We established serial port connection and used WinDbg to figure out hooking.

Before



After

Demonstration (Video)



Process Team

- From now, we will report Process Team

Function Team

Make functions of
* Detecting reading
* Hooking writing

윤원진, 김승윤,
최창민, 김순재

Process team

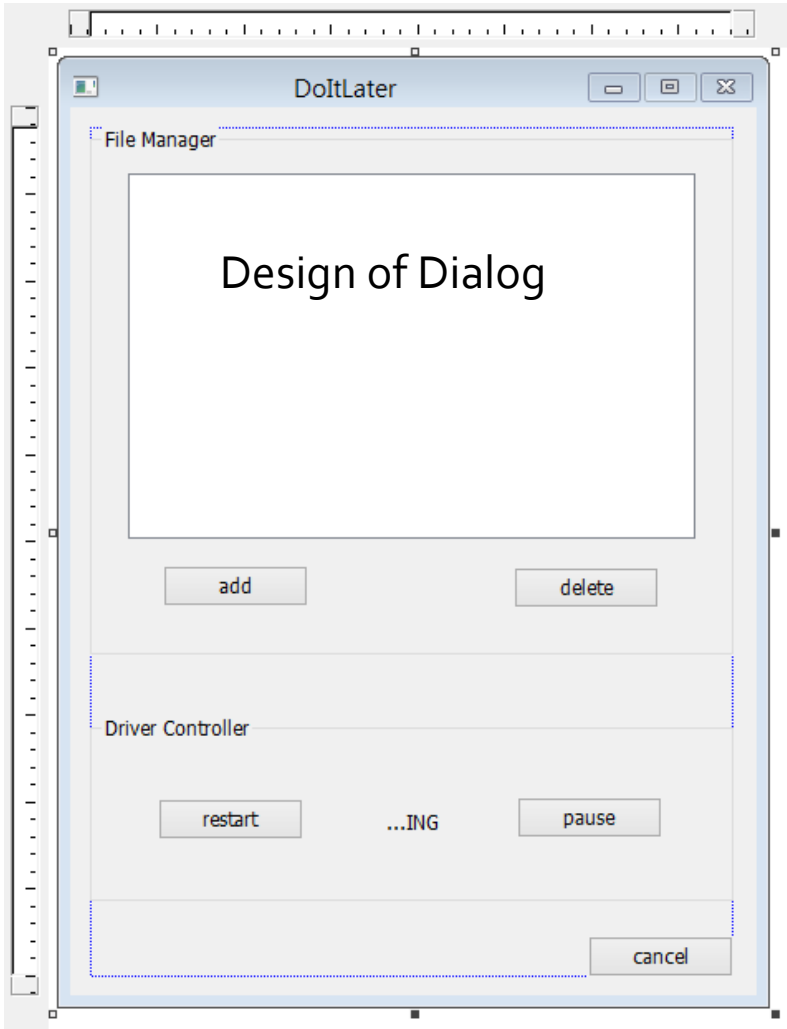
GUI
Process Protection
(Anti-closing)

정원태, 이수연,
장부루, 김태승

Process Team

- We takes care of 2 functions until interim report.
- First is GUI coding, and second is driver loading.

GUI coding

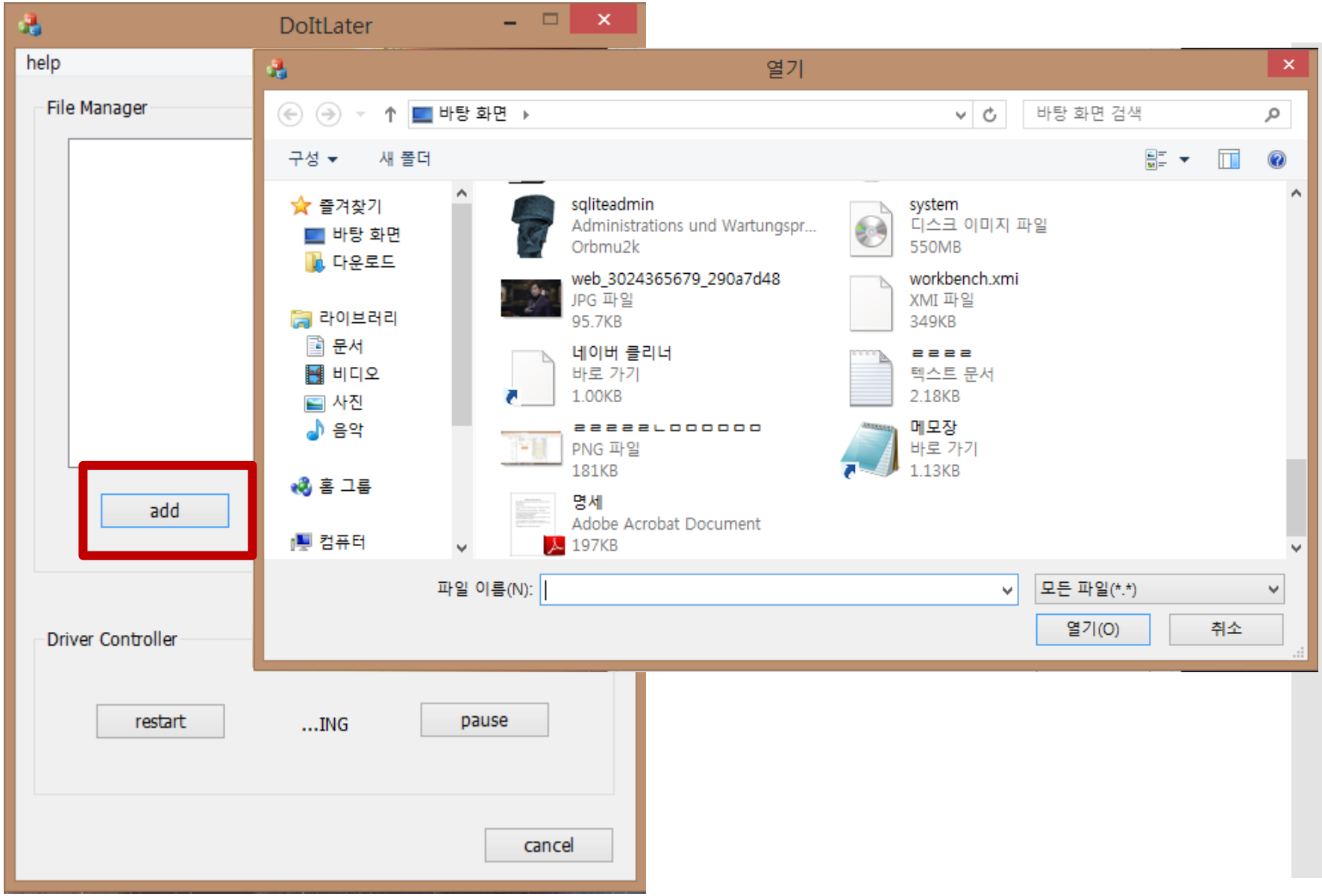



```
void CAhroobeDlg::OnBnClickedButton1()
{
    // TODO: 여기에 컨트롤 알림 처리기 코드를 추가합니다.

    static TCHAR BASED_CODE szFilter[] = _T("모든 파일(*.*)|*.*||");
    CFileDialog dlg(TRUE, _T("*.*"),_T(""), OFN_HIDEREADONLY, szFilter);
    if (IDOK == dlg.DoModal())
    {
        CString pathName = dlg.GetPathName();
        int SelectIndex = file_list.AddString(pathName);
        file_list.SetCurSel(SelectIndex);
        MessageBox(pathName);
    }

}
```

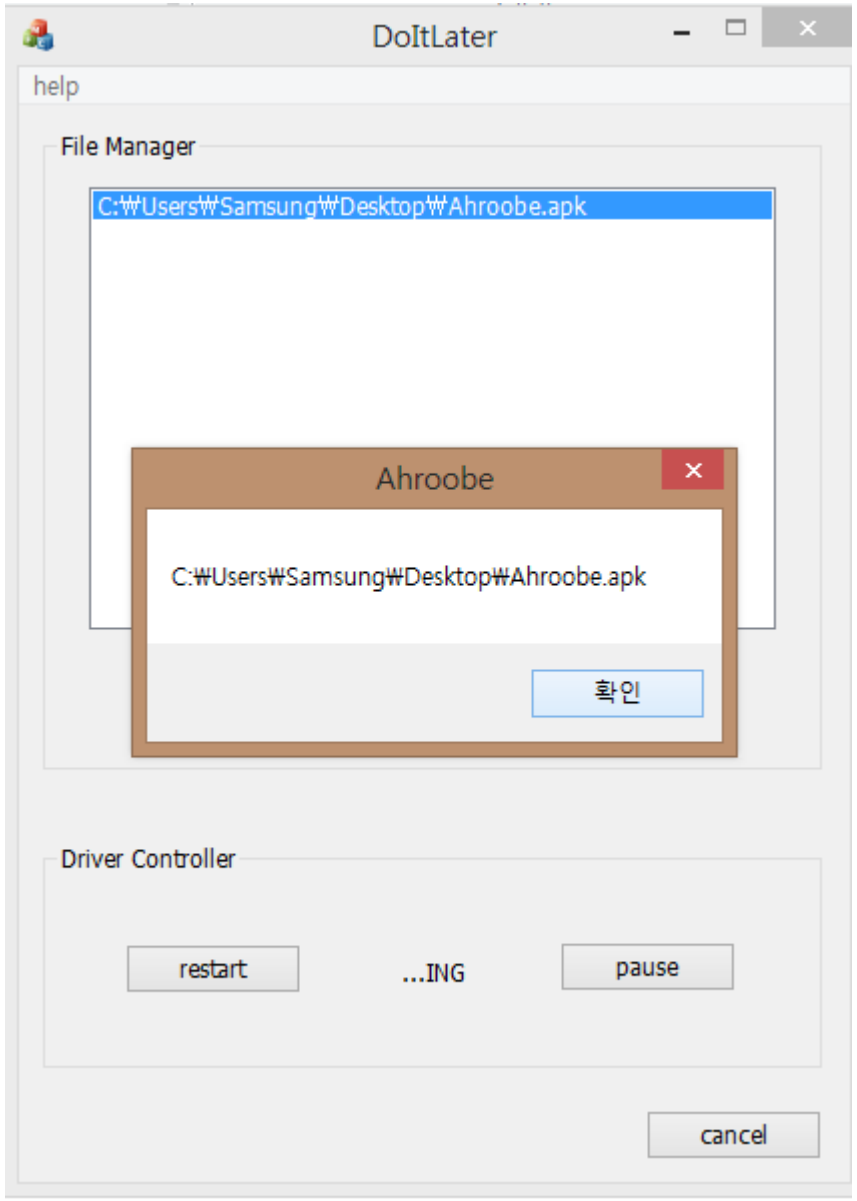
Added event controller on “Add” button.

Get file address by dlg.GetPathName()
and add on list (by file_list.SetCurSel(string))

Demonstration



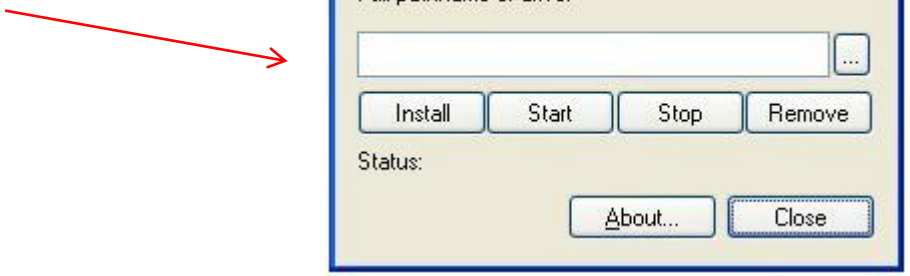
Demonstration



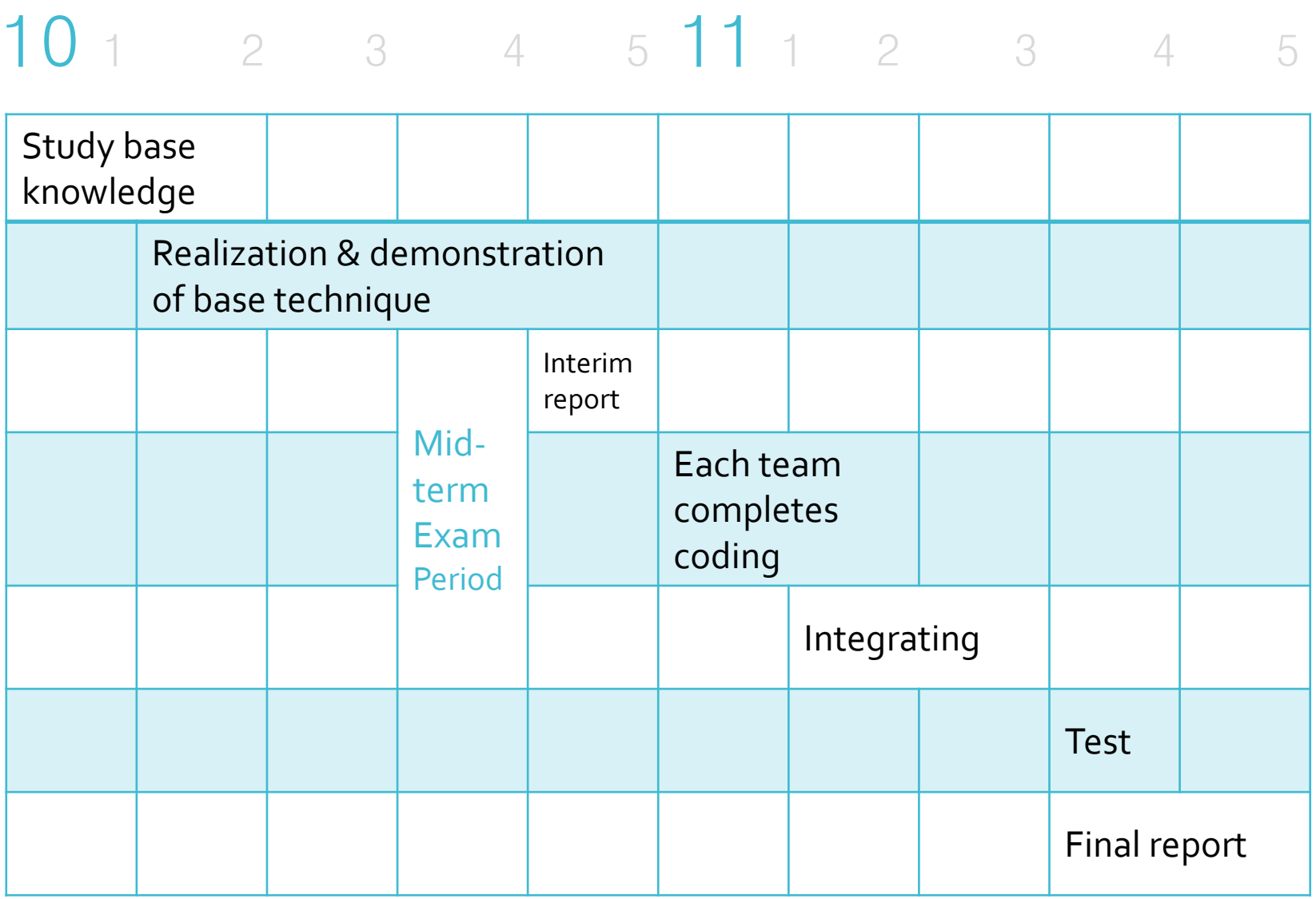
Driver Loading

- We found source code of “InstDriver” and use it under understanding.

```
class Load_Driver {
private:
    SC_HANDLE   scm;         // SCManager 핸들
    SC_HANDLE   svc;          // Service 핸들
    HANDLE  hFile;              // sys파일 핸들 (device 핸들과 다른 것임)
    wchar_t fullSysName[500];   // .sys파일 절대 경로
    wchar_t sysName[100];       // .sys파일 이름 (.sys 앞 부분, first.sys면 first)
    bool Load_Driver::OpenSCM();
    bool Load_Driver::GetDirectory(wchar_t *str);
    bool Load_Driver::GetDirectory(wchar_t *fullPath, wchar_t *sysName);
    bool Load_Driver::GetHandle(wchar_t *fullSysName);
public:
    Load_Driver(wchar_t *sysName)                   { GetDirectory(sysName); }
    Load_Driver(wchar_t *fullPath, wchar_t *sysName) { GetDirectory(fullPath, sysName); }
    ~Load_Driver() {
        CloseServiceHandle(svc);
        CloseServiceHandle(scm);
    }
    bool Load_Driver::InstallDriver();
    bool Load_Driver::StartDriver();
    bool Load_Driver::StopDriver();
    bool Load_Driver::RemoveDriver();
};
```



Timeline


10
1
2
3
4
5
11
1
2
3
4
5
Study base knowledge
Realization & demonstration of base technique
Interim report
Mid-term Exam Period
Each team completes coding
Integrating
Test
Final report

End of interim report

- Thank you for your attention
- You are welcomed to question

Anti-Copying Copy Protector ; Final

Information Security

DoItLater Leader : WonJin Yoon, SeoungYoon Kim, ChangMin Choi, SunJae Kim, BuRu Jang, TaeSeong Kim, WonTae Jeong, SuYeon Lee

Table of Contents

Explanation of project

Technical explanation – Driver part

Technical explanation – Program

Demo Video

CONTENTS

- Explanation of project
- Technical explanation – Driver part
- Technical explanation – Program
- Demo Video
- Q & A

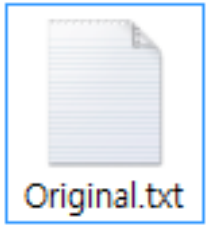
Explanation of project

- We wants to prevent important files from being copied with out permission.

- Let's think of a situation.
  Malicious user(internal betrayer) tries to copy DB.
  He accesses server and mounts USB(or maybe HDD).

Explanation of project

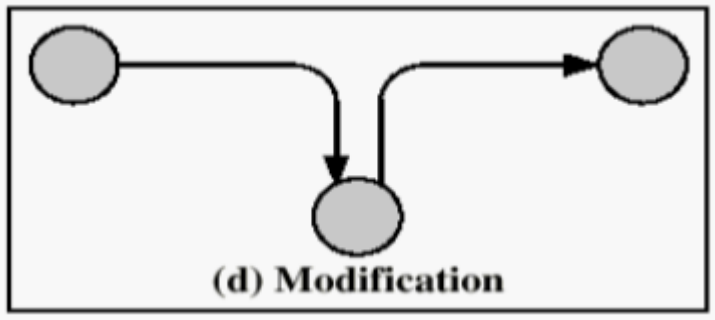
What attacker sees

Original File	Copied File
	
C:\DB\Original.txt	E:\Original.txt

- He copies DB into his USB. It seems to be flawless.
- Instead of copying, dummy file is written in USB.

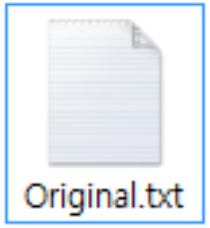
- Anyway, he checks copied file by opening it.
- But under this system(server which originally contains Original file), copied file seems valid in his eyes.

Explanation of project

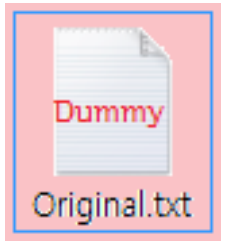


When Copying

Original File



C:\DB\Original.txt



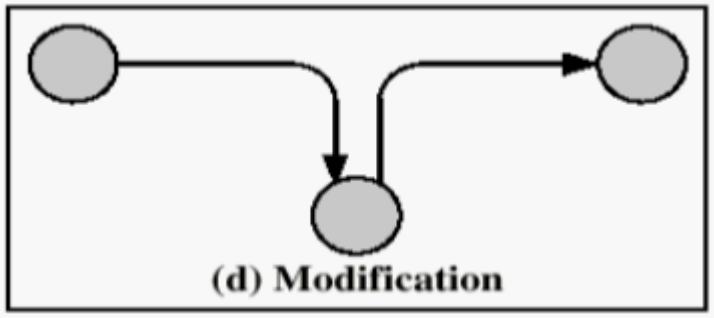
Copied File

E:\Original.txt

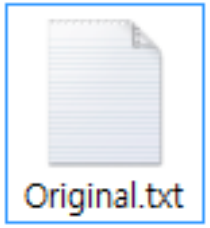
- However, We hooked SSDT Table.
  * SSDT is a table that contains addresses of internal functions.
- So we could manipulate reading and writing process.
- The result is… Deceiving attacker.
  So that he can't even think of neutralize our program.
- What he see is original file. But what he get is dummy.

When Opening

Explanation of project

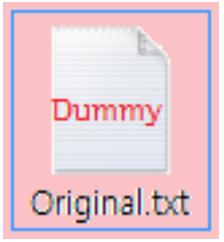


Original File



C:\DB\Original.txt

Copied File



E:\Original.txt

- However, We hooked SSDT Table.
  * SSDT is a table that contains addresses of internal functions.
- So we could manipulate reading and writing process.

- The result is… Deceiving attacker.
  So that he can't even think of neutralize our program.
- What he see is original file. But what he get is dummy.

CONTENTS

- Explanation of project
- Technical explanation – Driver part
- Technical explanation – Program
- Demo Video
- Q & A

Technical explanation

Driver part

- In driver part, we have two sub part.
- One is Read&Write function(in SSDT) hooking. – Our main idea.
  To manipulate copying procedure.
- The other is Terminate Process function(in SSDT) hooking.
  To prevent our process from forced termination.

Technical explanation

Driver part

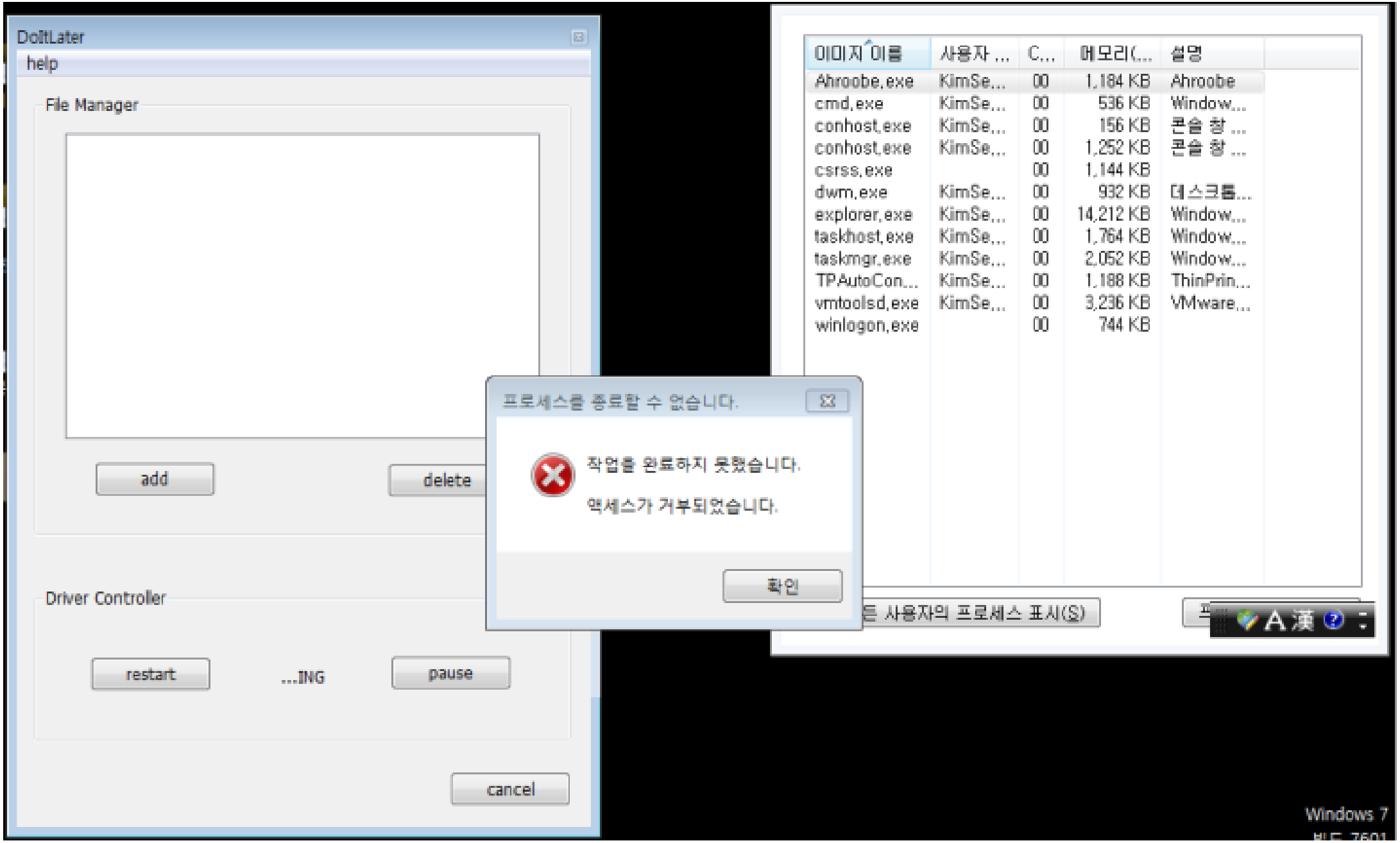
- **Read&Write function(in SSDT) hooking** is base of our main idea.
- We can manipulate reading and writing process. So that we can achieve our concept.
- We have limited time to present and have explained in previous slides, so we will skip detailed information of this sub part.

  (And also we had presented detailed technique in interim report and proposal)
- We will write detailed information on "report".

Technical explanation

Driver part

- **ZwTerminateProcess(in SSDT) hooking** is to prevent our process from forced termination.
- ZwTerminateProcess function gets PID of target process. And we get "struct" of target process by PID.
- And GetCurrentProcess function gets "struct" of killer process.
- Compare those two "struct" and if abnormal terminating signal is detected, terminating will be denied.


DoItLater
help
File Manager
add
delete
Driver Controller
restart
...ING
pause
cancel
이미지 이름
사용자 ...
C...
메모리(...
설명
Ahroobe.exe KimSe... 00 1,184 KB Ahroobe
cmd.exe KimSe... 00 536 KB Window...
conhost.exe KimSe... 00 156 KB 콘솔 창 ...
conhost.exe KimSe... 00 1,252 KB 콘솔 창 ...
csrss.exe 00 1,144 KB
dwm.exe KimSe... 00 932 KB 데스크톱...
explorer.exe KimSe... 00 14,212 KB Window...
taskhost.exe KimSe... 00 1,764 KB Window...
taskmgr.exe KimSe... 00 2,052 KB Window...
TPAutoCon... KimSe... 00 1,188 KB ThinPrin...
vmtoolsd.exe KimSe... 00 3,236 KB VMware...
winlogon.exe 00 744 KB
든 사용자의 프로세스 표시(S)
프로세스를 종료할 수 없습니다.
작업을 완료하지 못했습니다.
액세스가 거부되었습니다.
확인
Windows 7

CONTENTS

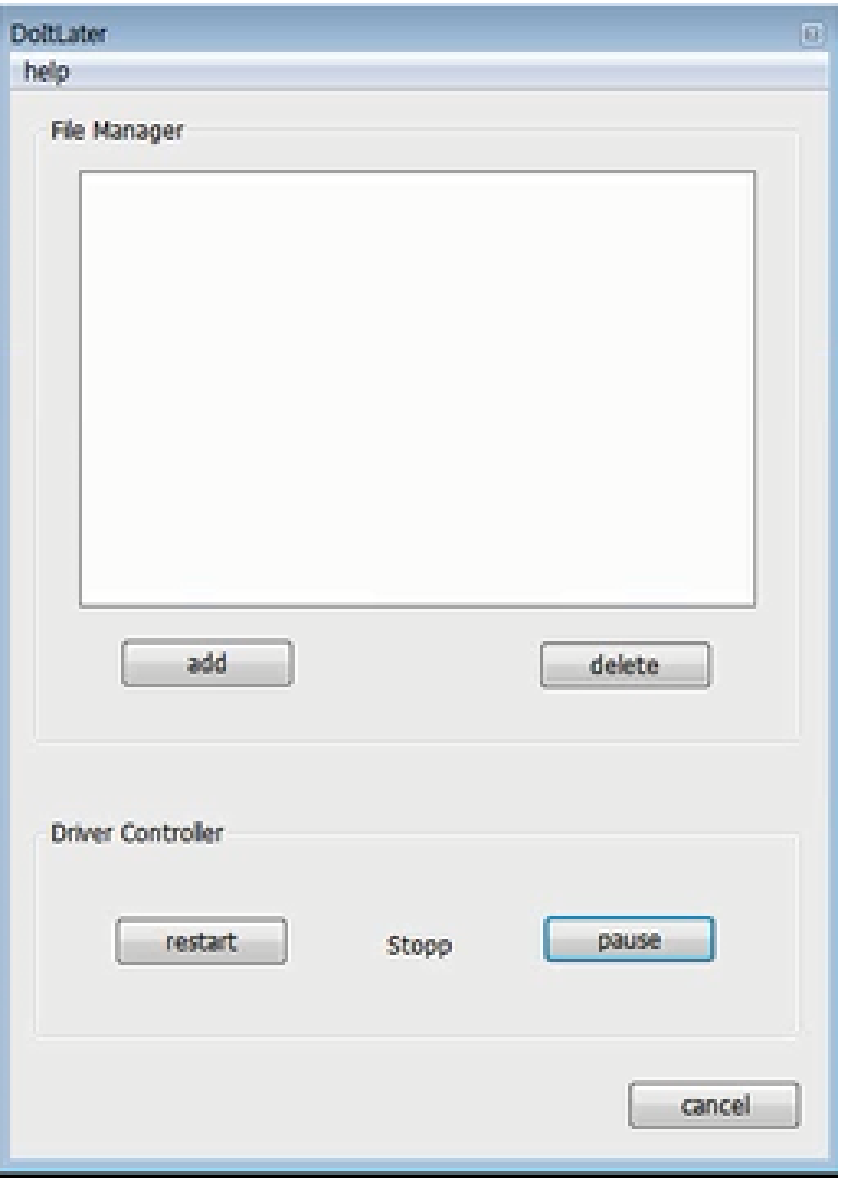
Explanation of project

Technical explanation – Driver part

Technical explanation – Program

Demo Video

Q & A

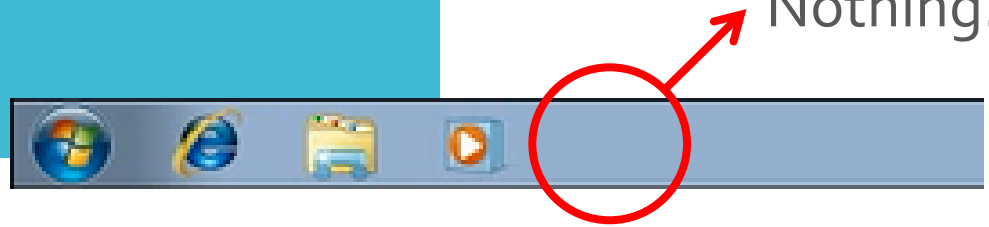


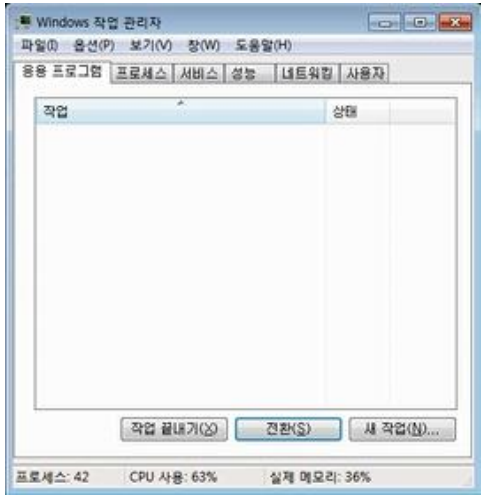
- When the driver is loaded and started, exit, add and delete buttons are disabled.
- After driver is paused all the buttons are enabled.

Technical explanation

Program

- Esc, Alt+F4, Enter key and Close button are prohibited in our program.
- Program is not shown in task bar.



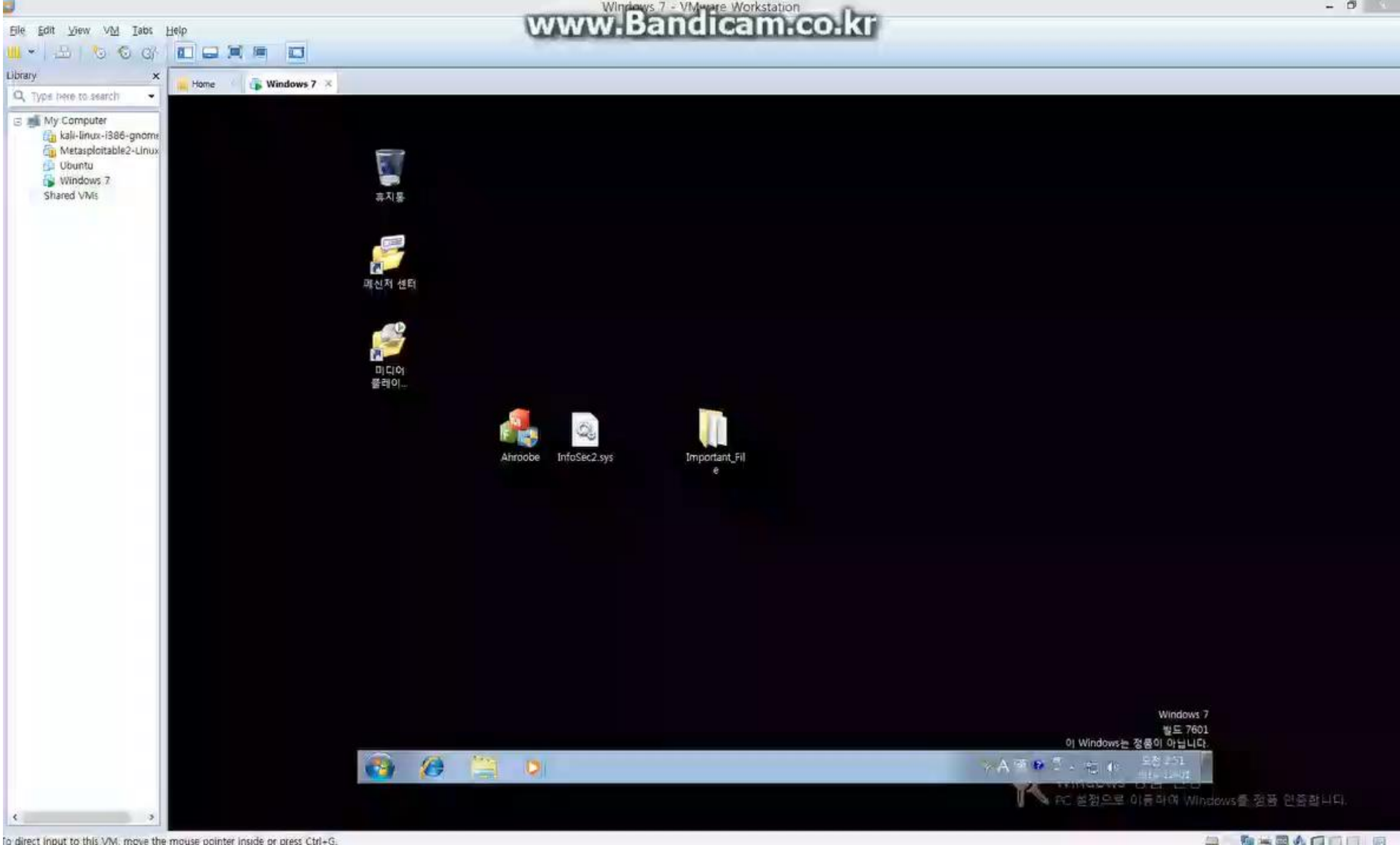


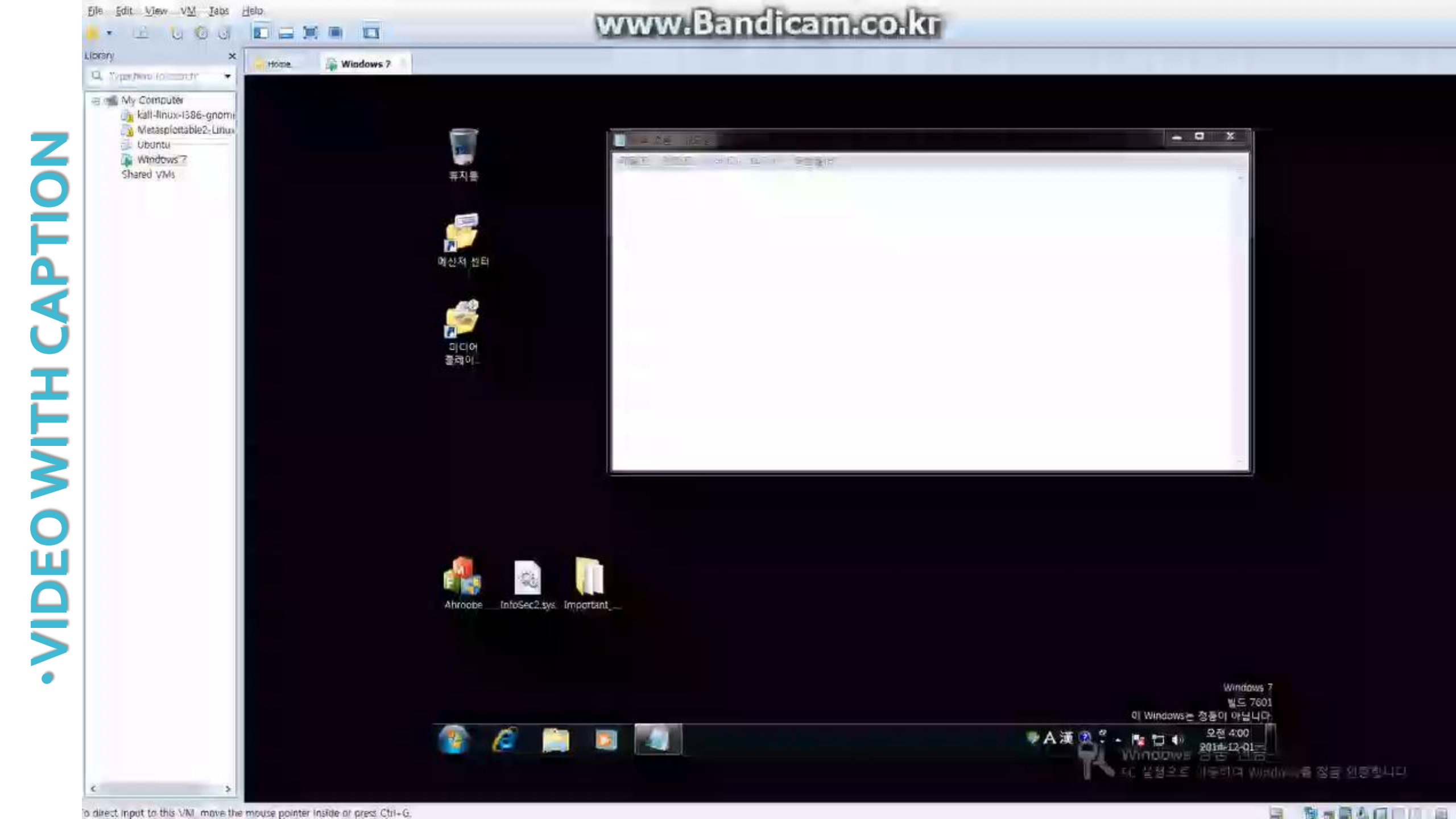
CONTENTS

- Explanation of project
- Technical explanation – Driver part
- Technical explanation – Program
- Demo Video
- Q & A

Demo Video

- GOTO VIDEO WITHOUT CAPTION(1:29)
- GOTO VIDEO WITH CAPTION(3:14)





•VIDEO WITH CAPTION

CONTENTS

- Explanation of project
- Technical explanation – Driver part
- Technical explanation – Program
- Demo Video
- Q & A

End of Final report

- Thank you for your attention.