

201716905 김강민

# Windows 악성코드 지속성 유지 기법

IT 정보공학과 BCG LAP 201716905 김강민

00. Windows Registry

01. Run Registry

02. Windows 스케줄러

03. 시작 폴더

04. Winlogon Registry

05. Image File Execution Option(IFE0)

06. Applnit\_DLLs

07. DLL Search Order hijacking

08. COM hijacking

09. Service

10. 지속성 탐지 방법

## · Windows Registry

- Windows OS의 설정과 선택 항목을 담고 있는 데이터베이스
- 모든 HW, 대부분의 SW, 사용자 PC 선호도 등에 대한 정보와 설정 저장
- 레지스트리 편집기를 통해 수정 가능

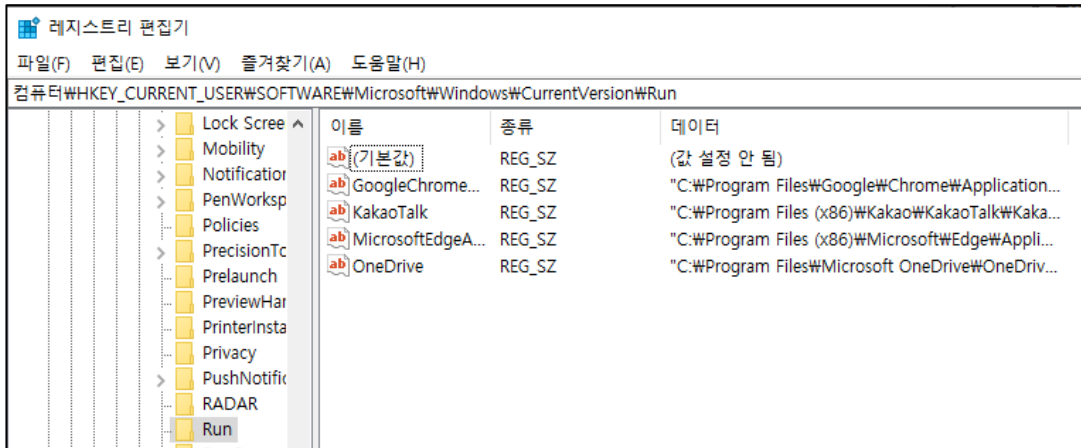
## • Windows Registry

- **HKEY\_CLASSES\_ROOT(HKCR)** : 응용 프로그램의 정보
- **HKEY\_CURRENT\_USER(HKCU)** : 현재 로그인한 사용자 환경 설정
- **HKEY\_LOCAL\_MACHINE(HKLM)** : 시스템 전체에 대한 환경 설정
- **HKEY\_USERS** : Desktop 설정 및 Network 연결 정보
- **HKEY\_CURRENT\_CONFIG** : 디스플레이와 프린터에 관한 정보



## • Run Registry

- System이 부팅될 때 실행되는 프로세스를 등록하는 Registry
- Run, RunOnce, RunOnceEx Registry 존재
- HKEY\_CURRENT\_USER와 HKEY\_Local\_Machine에 존재



HKCU\Software\Microsoft\Windows\CurrentVersion\Run
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnce
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
HKCU\Software\Microsoft\Windows\CurrentVersion\RunOnceEx
HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce
HKCU\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run
HKLM\Software\Microsoft\Windows\CurrentVersion\Policies\Explorer\Run

## • Run Registry

	HKCU	HKLM
쓰기 권한	사용자 수정 가능	관리자 권한 필요
적용 범위	해당 사용자 부팅시 실행	전체 시스템

	Run	RunOnce	RunOnceEx
기능	시스템 부팅마다 실행	한 번만 실행, 프로그램 시작 후 레지스트리 삭제	한 번만 실행, 프로그램 종료 후 레지스트리 삭제

## · Windows 스케줄러

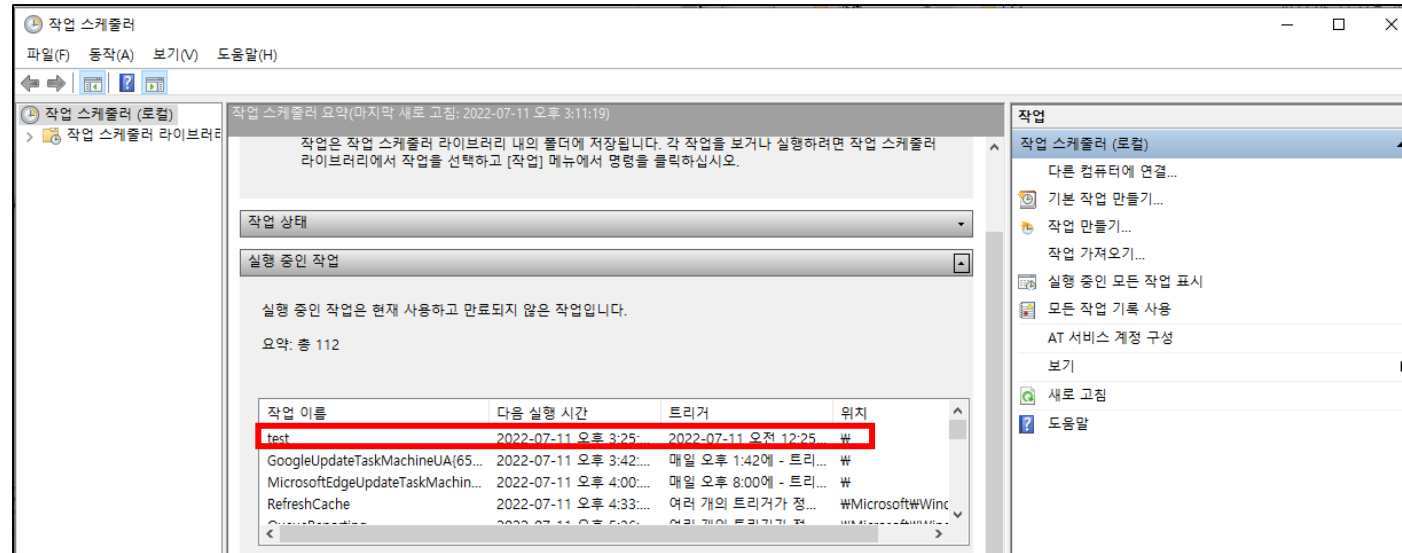
- 지정된 시간이나 시스템이 시작하는 동안 악성코드가 실행할 수 있도록 예약
- 실제 Command Line 프로그램은 schtasks.exe와 at.exe
- schtasks.exe와 at.exe로 cmd를 통해 공격자가 원하는 시간에 프로그램 또는 스크립트를 스케줄하는 데 사용
- linux의 crontab과 유사

```
cmd.exe /c schtasks /Create /SC once /TN drogon /RU SYSTEM /TR %WinDir%\system32\shutdown.exe /r /t 0 /f /ST:시간
```

## • Windows 스케줄러

- Atutoruns 또는 작업 스케줄러를 통해 현재 스케줄 작업 목록 탐지

```
C:\Users\KIM_GANG_MIN>schtasks /create /tn "test" /SC hourly /st 00:25 /tr C:\Users\KIM_GANG_MIN\Desktop\Secure\NordSec2020.pdf  
성공: 예약된 작업 "test"을(를) 만들었습니다.
```



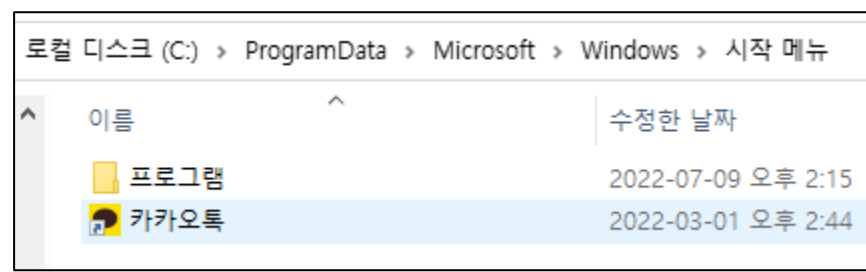
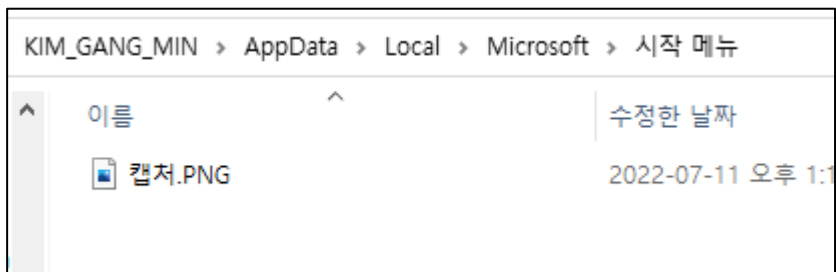


## · 시작 폴더

- 시스템 시작 시, 시작 폴더를 살펴보고 폴더 안의 파일 실행
- App Data : 유저 시작 폴더, ProgramData : 시스템 시작 폴더

C:\%AppData%\Microsoft\Windows\Start Menu\Programs\Startup

C:\ProgramData\Microsoft\Windows\Start Menu\Programs\Startup



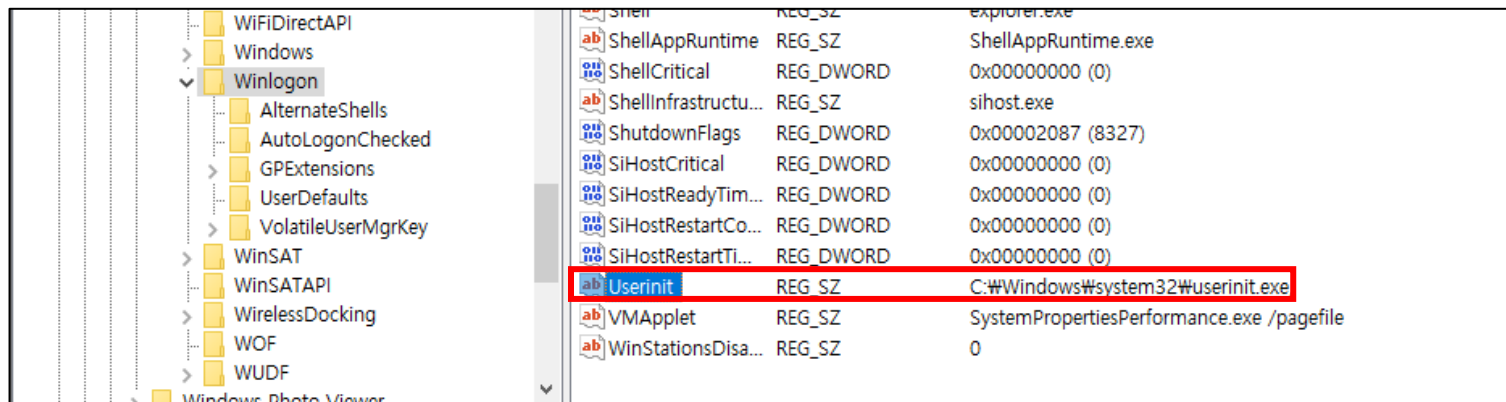
## • Winlog.exe

- Winlog.exe는 logon과 logoff를 처리
- 로그인 시 사용자 프로필을 로드
- 화면 보호기 실행 중일 시 컴퓨터를 잠금

## • Winlogon Registry

- 로그인 유저 인증 시 userinit.exe를 통해 초기화 진행
- userinit.exe를 악성코드로 대체하거나 추가

MKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\Userinit

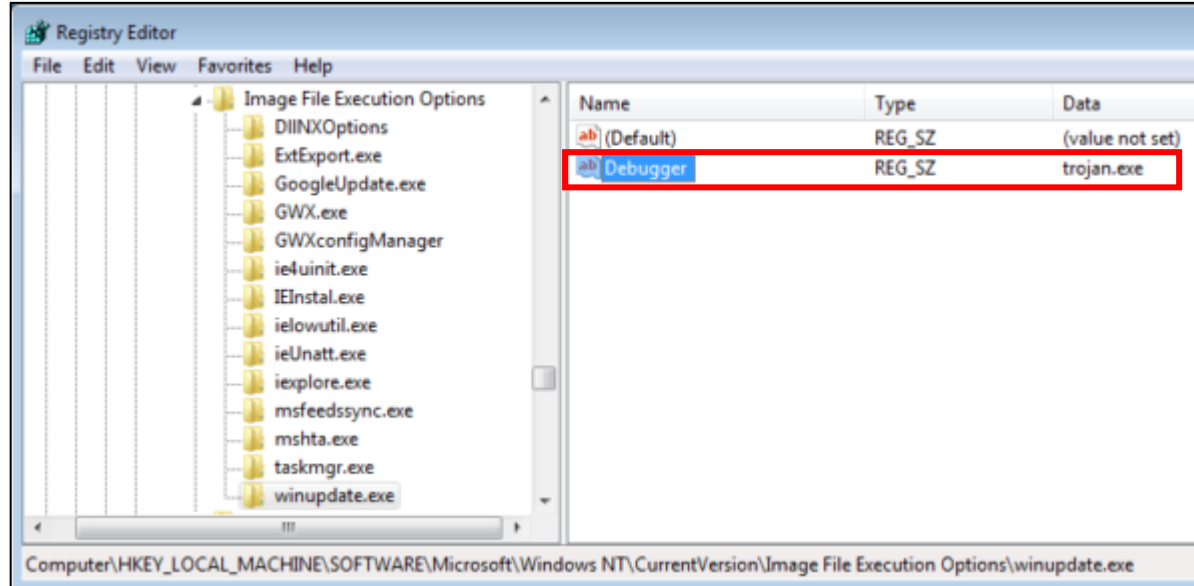


WIFIDirectAPI	Shell	REG_SZ	explorer.exe
Windows	ShellAppRuntime	REG_SZ	ShellAppRuntime.exe
Winlogon	ShellCritical	REG_DWORD	0x00000000 (0)
AlternateShells	ShellInfrastructure	REG_SZ	sihost.exe
AutoLogonChecked	ShutdownFlags	REG_DWORD	0x00002087 (8327)
GPEExtensions	SiHostCritical	REG_DWORD	0x00000000 (0)
UserDefaults	SiHostReadyTim...	REG_DWORD	0x00000000 (0)
VolatileUserMgrKey	SiHostRestartCo...	REG_DWORD	0x00000000 (0)
WinSAT	SiHostRestartTi...	REG_DWORD	0x00000000 (0)
WinSATAPI	Userinit	REG_SZ	C:\Windows\system32\userinit.exe
WirelessDocking	VMApplet	REG_SZ	SystemPropertiesPerformance.exe /pagefile
WOF	WinStationsDisa...	REG_SZ	0
WUDF			

## • Image File Execution Option(IFEO)

- IFEO는 디버거에서 실행 파일을 직접 실행할 수 있게 만드는 역할
- 변경 시 디버거가 Attach 시킬 때 정상 프로그램이 아닌 악성 프로그램 실행

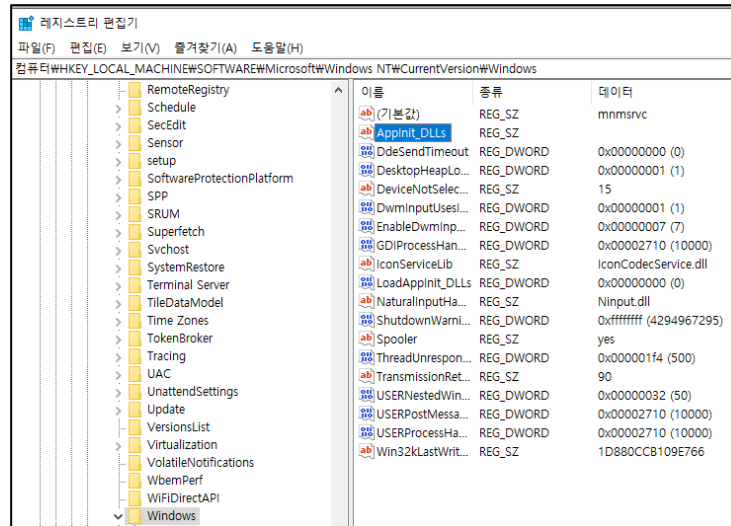
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\<File name>



## • AppInit\_DLLs

- 시스템의 프로세스에 DLL을 로드할 수 있는 메커니즘
- User32.dll 로드하는 모든 프로세스에 로드 (거의 모든 프로세스)
- DLL이 프로세스의 주소 공간에 로드 되면 프로세스 내에서 실행될 수 있고, API 가로채어 대체 기능 수행 가능

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\AppInit\_DLLs



## • Applnit\_DLLs

- Microsoft는 기본적으로 Applnit을 통해 DLL을 로드하지 않도록 설정
- LoadApplnit\_DLLs 값을 1로 변경

HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Windows\Applnit\_DLLs

이름	종류	데이터
(기본값)	REG_SZ	mnmsrvc
Applnit_DLLs	REG_SZ	
DdeSendTimeout	REG_DWORD	0x00000000 (0)
DesktopHeapLo...	REG_DWORD	0x00000001 (1)
DeviceNotSelec...	REG_SZ	15
DwmInputUsesl...	REG_DWORD	0x00000001 (1)
EnableDwmInp...	REG_DWORD	0x00000007 (7)
GDIProcessHan...	REG_DWORD	0x00002710 (10000)
IconServiceLib	REG_SZ	IconCodecService.dll
LoadApplnit_DLLs	REG_DWORD	0x00000000 (0)
NaturalInputHa...	REG_SZ	Ninput.dll
ShutdownWarni...	REG_DWORD	0xffffffff (4294967295)
Spooler	REG_SZ	yes
ThreadUnrespon...	REG_DWORD	0x000001f4 (500)
TransmissionRet...	REG_SZ	90
USERNestedWin...	REG_DWORD	0x00000032 (50)
USERPostMessa...	REG_DWORD	0x00002710 (10000)
USERProcessHa...	REG_DWORD	0x00002710 (10000)
Win32kLastWrit...	REG_SZ	1D880CCB109E766

## • KnownDLLs

- DLL 로드 시에 메모리를 확인하고, 있을 시 메모리에 있는 DLL을 사용, 로드되지 않을 시 KnownDLLs 레지스트리 키에 정의된 DLL인지 확인
- 해당 KnownDLLs 목록 안에 있으면 System32에서 로드, 없을 경우 Search Order에 따라 DLL을 찾음
- 프로그램이 사용하는 사용자 정의 DLL은 KnownDLLs에 존재하지 않음

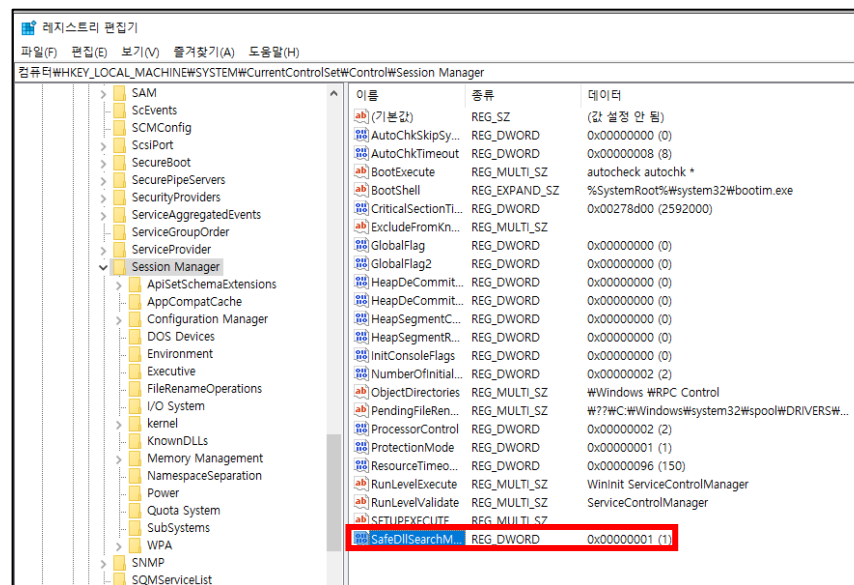
HKLM\SYSTEM\CurrentControlSet\Control\Session Manager\KnownDLLs



## • SafeDllSearchMode

- SafeDllSearchMode 활성화 여부에 따라 Search Order가 달라짐
- Default 값은 enable (일반 사용자는 설정을 건드리지 않았다고 봐도 무방)

HKLM\System\CurrentControlSet\Control\Session Manager\SafeDllSearchMode





## • System Order

### - SafeDllSearchMode : enable

1. 응용 프로그램이 로드된 디렉터리
2. 시스템 디렉터리 (GetSystemDirectory 함수를 사용하여 이 디렉터리의 경로를 획득 - C:\Windows\System32)
3. 16비트 시스템 디렉터리 (이 디렉터리의 경로를 가져오는 함수는 없지만 검색 됨 - C:\Windows\System)
4. 윈도우즈 디렉터리 (GetWindowsDirectory 함수를 사용하여 이 디렉터리의 경로를 획득 - C:\Windows)
5. 현재 디렉터리
6. PATH 환경 변수에 나열된 디렉터리

### - SafeDllSearchMode : disable

1. 응용 프로그램이 로드된 디렉터리
2. 현재 디렉터리
3. 시스템 디렉터리 (GetSystemDirectory 함수를 사용하여 이 디렉터리의 경로를 획득 - C:\Windows\System32)
4. 16비트 시스템 디렉터리 (이 디렉터리의 경로를 가져오는 함수는 없지만 검색 됨 - C:\Windows\System)
5. 윈도우즈 디렉터리 (GetWindowsDirectory 함수를 사용하여 이 디렉터리의 경로를 획득 - C:\Windows)
6. PATH 환경 변수에 나열된 디렉터리

## • DLL System Order

1. 응용 프로그램이 로드된 디렉터리
2. 시스템 디렉터리 (GetSystemDirectory 함수를 사용하여 이 디렉터리의 경로를 획득 - C:\Windows\System32)
3. 16비트 시스템 디렉터리 (이 디렉터리의 경로를 가져오는 함수는 없지만 검색 됨 - C:\Windows\System)
4. 윈도우즈 디렉터리 (GetWindowsDirectory 함수를 사용하여 이 디렉터리의 경로를 획득 - C:\Windows)
5. 현재 디렉터리
6. PATH 환경 변수에 나열된 디렉터리

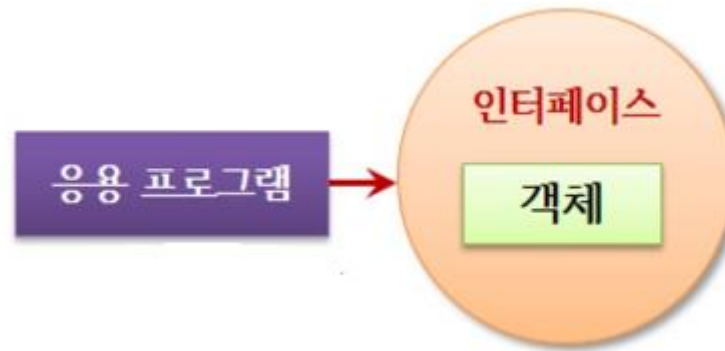
### <Example>

\* SelfDllSearchMode : enable

1. KnownDLL에 없는 정상적인 dll : example.dll
2. example.dll의 위치 : C:\Windows (우선순위 4)
3. 공격자가 악성 dll을 시스템 디렉터리(C:\Windows\System32)에 저장 (우선순위 2)
4. 정상적인 프로세스가 example.dll 요청
5. 시스템은 KnownDLLs 확인 → 발견 x
6. Search Order에 따라 탐색
7. 악성 dll을 정상 dll보다 빨리 발견 및 실행
8. 악성 dll 로드

## · Component Object Model(COM) 객체

- DirectX의 프로그래밍 언어 독립성과 하위 호환성을 가능하게 하는 기술
- COM 객체를 가리키는 포인터를 특별한 함수를 이용해서, 또는 다른 COM 인터페이스의 메서드를 이용
- 소프트웨어 컴포넌트가 서로의 코드에 대해 알지 못하더라도 서로 상호작용하고 통신할 수 있는 시스템

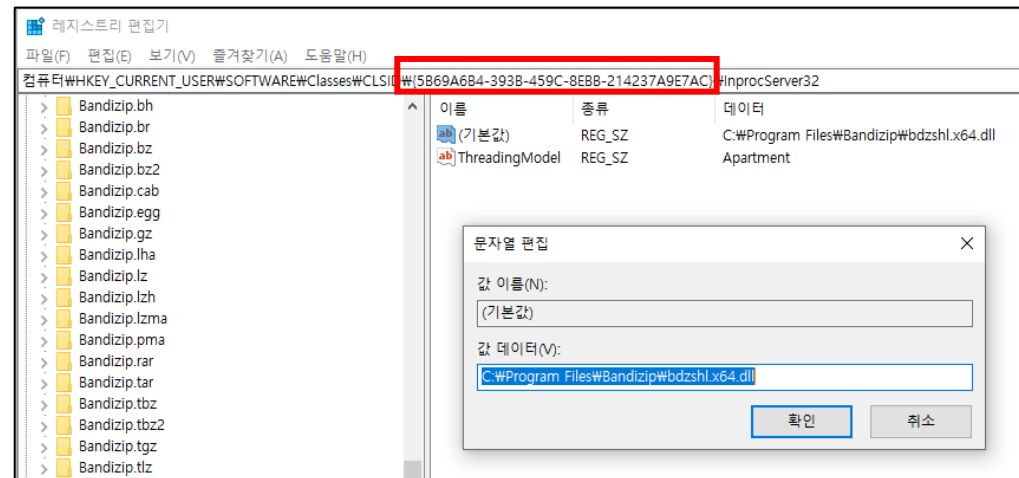


## • CLSIDs

- COM 객체를 식별하는 고유 번호 (PID와 유사)
- $HKEY\_Classes\_ROOTWCLSID = HKEY\_Local\_Machine\SOFTWARE\Classes\WCLSIDW + HKEY\_Current\_User\SOFTWARE\Classes\WCLSID$
- HKEY\_Classes\_ROOT이나 HKEY\_Local\_Machine은 관리자 권한이 필요하지만, HKEY\_Current\_User는 일반 user도 조작 가능

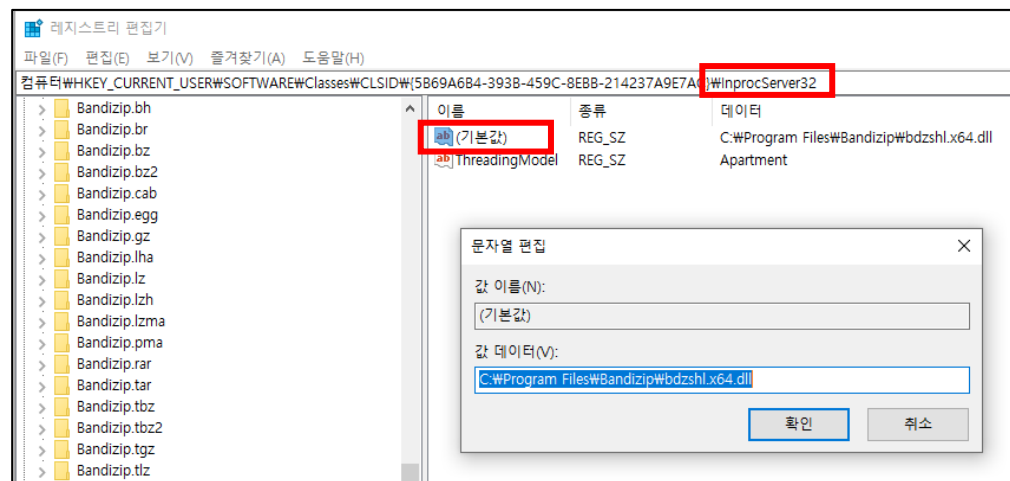
HKCU\CLSID\<CLSID>\InprocServer32\<기본값>

컴퓨터\HKEY_CLASSES_ROOT\WCLSID\{5B69A6B4-393B-459C-8EBB-214237A9E7AC}			
	이름	종류	데이터
> {5A57485B-151F-4868-945B-FBB95B5740}	(기본값)	REG_SZ	AABdzCtx Class
> {5A580C11-E5EB-11d1-A86E-0000F8084F}			
> {5a6efd3c-a6b5-44ee-873f-9b25bdbe045b}			
> {5A823D1E-8DDC-44DD-8A84-F9D32C7F}			
> {5A8A3AAA-D614-46B9-B814-18D168A4B}			
> {5A8A3AAB-D614-46B9-B814-18D168A4B}			
> {5A8A3AAC-D614-46B9-B814-18D168A4B}			
> {5A90DD8E-2A0C-45D1-873A-82B61604C}			
> {5A94A793-7529-4933-8240-4146CC3D4}			
> {5A984BF5-D07C-4C99-9E5C-37156F21EE}			



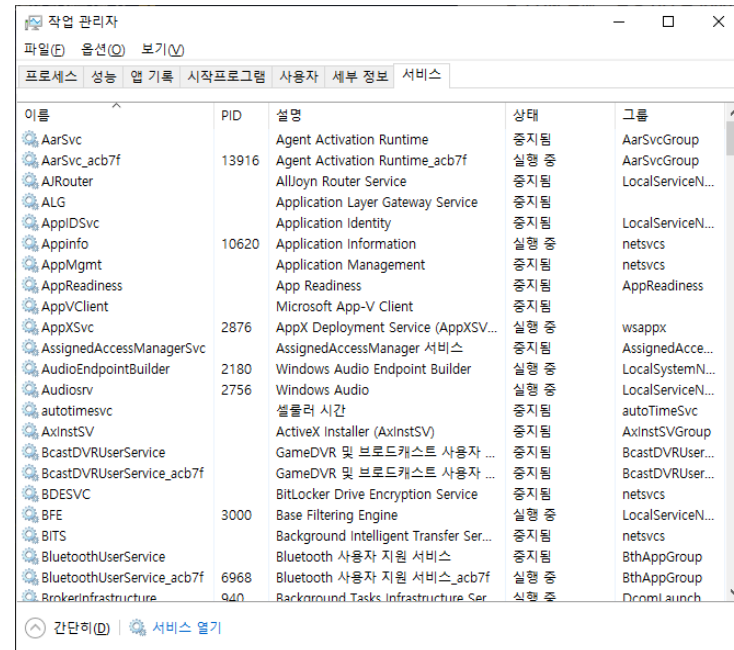
## • COM hijacking

1. CLSID의 InprocServer32의 Key 값 수정
2. System은 HKEY\_CURRENT\_USER를 우선적으로 경로 load
3. COM 객체가 HKEY\_CURRENT\_USER에 적힌 경로를 참고
4. COM 객체는 악성 DLL을 load



## • Windows Service

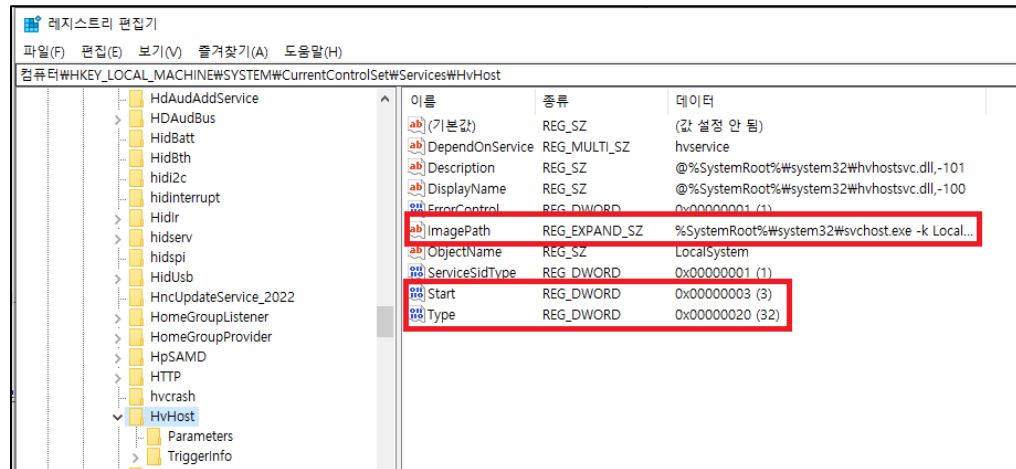
- 윈도우가 부팅될 때 자동으로 실행되며 지속적으로 존재
- 유저 인터페이스 없이 백그라운드에서 실행되는 프로그램이며 이벤트 로깅, 프린트, 에러 레포팅 등과 같은 핵심 운영 시스템 기능을 제공



## • Window32OProcess

- 실행파일(.exe)로 구현, 개별 프로세스 실행
- Imagepath : Service를 위한 실행파일 경로
- Start : Service 언제 시작되는지 정하는 값
- Type : Service type

HKLM\SYSTEM\CurrentControlSet\Services



## • Window32OProcess

- Start : Seservice 언제 시작되는지 정하는 값

Value	Meaning
SERVICE_AUTO_START (0x00000002)	부팅 중에 자동으로 서비스를 시작
SERVICE_BOOT_START (0x00000000)	시스템 로더에 의해 시작된 장치 드라이버
SERVICE_DEMAND_START (0x00000003)	'StartService 함수를 호출할 때 시작하는 서비스
SERVICE_DISABLED (0x00000004)	서비스를 시작 불가
SERVICE_SYSTEM_START (0x00000001)	장치 드라이버가 IoInitSystem 함수를 실행

- Type : Service type

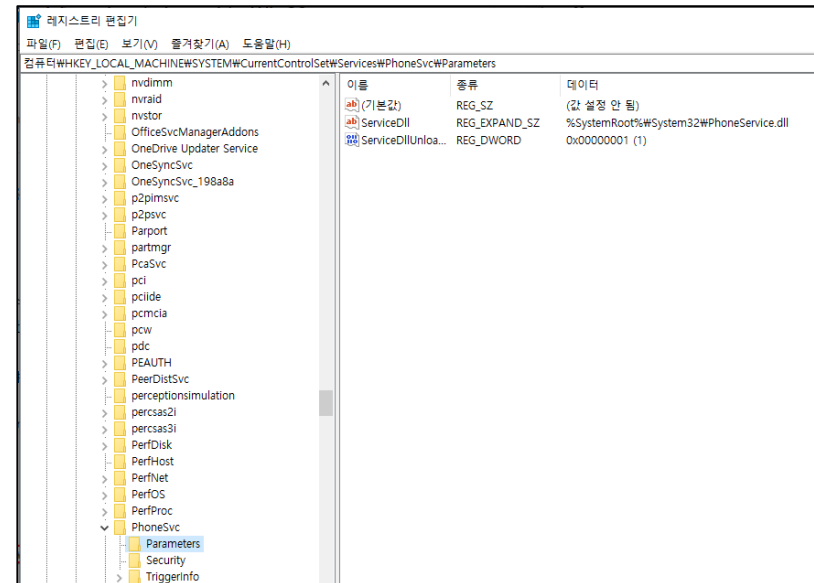
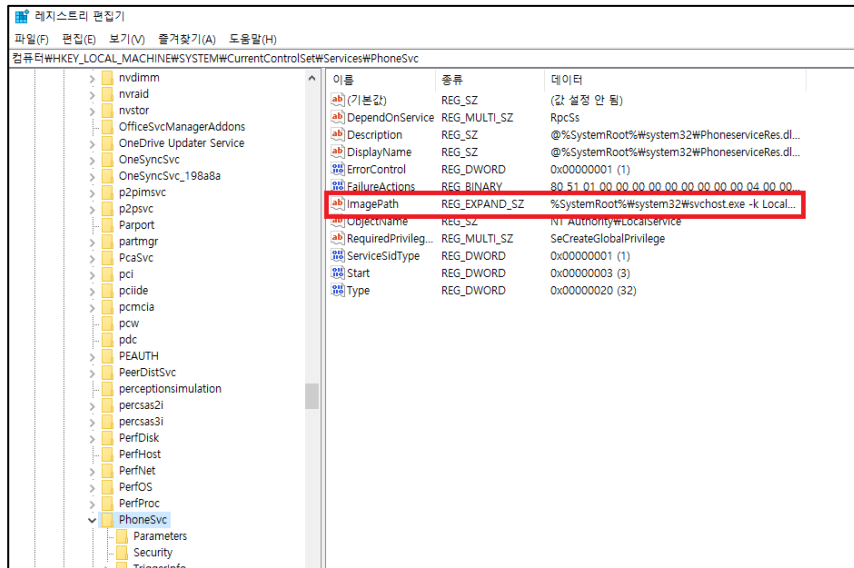
Value	Meaning
SERVICE_ADAPTER (0x00000004)	예약된 Service
SERVICE_FILE_SYSTEM_DRIVER (0x00000002)	File system 드라이버 Service
SERVICE_KERNEL_DRIVER (0x00000001)	드라이버 Service
SERVICE_RECOGNIZER_DRIVER (0x00000008)	예약된 Service
SERVICE_WIN32_OWN_PROCESS (0x00000010)	자체 실행되는 Service
SERVICE_WIN32_SHARE_PROCESS (0x00000020)	하나 이상의 프로세스와 공유하는 Service



## • Window32ShareProcess

- 실행파일(.dll)로 구현, 공유 호스트 프로세스(svchost.exe) 실행
- Service가 dll이면 svchost를 통해 실행

HKLM\SYSTEM\CurrentControlSet\Services\Parameters



## · Service 생성 방법

### 1. sc 유틸리티 사용

- cmd를 통해 sc create와 sc start 사용하여 서비스를 생성 및 실행

### 2. 배치 스크립트(.bat) 사용

- 배치 스크립트로 sc를 사용한 명령어를 실행해 서비스 생성 및 실행

### 3. 윈도우 API 사용

- CreateService()와 StartService()와 같은 윈도우 API를 사용해 서비스를 생성 및 시작
- OpenScManager()를 통해 서비스 제어 관리자 핸들 업고, CreateService() 호출해 WinShareProcess(dll) 유형의 서비스 생성 후 악성 dll로 연결

## · Kernel Driver Service

- 새로운 Service 생성하지 않고 기존 서비스를 수정
- 사용하지 않거나 비활성화된 Service를 hijacking (탐지 어렵)
- 정상 드라이버를 악의적인 드라이버로 대체하고, 서비스와 관련된 레지스트리 수정 후 자동시작 설정

## · 지속성 탐지

- 정상 프로그램과 관련되지 않은 서비스 레지스트리 항목의 변화 모니터링
- 서비스와 관련된 바이너리 수정과 서비스 시작 유형의 변화 확인
- 서비스와 상호작용하는데 사용할 수 있는 sc, 파워셸, WMI와 같은 도구의 사용을 모니터링하고 로깅
- AutoRuns 유틸리티를 통해 서비스 사용을 검사