



# Digital Forensics Tool Development

(with. Batch File Programming)

%\_date% = 2019-10-15

%\_Name% = C0W3LL



# Contents

1. Batch File ?
2. Batch File Command
3. Command & Tool Explanation
4. Code review - 1
5. Volatility ?
6. Code review - 2
7. 예제



# batch file ?

여러 명령어를 한번에 실행시키기 위한 언어

DOS부분과 Windows에서만 동작

바보적인 작업이나 선택적인 작업에 대한 자동화 실행을 만들 수 있다.

다만 UI를 지원하지 않으며 단순히 CMD창이나 DOS화면을 통해서만 동작하는 단점이 있음



# Batch File Command

echo : 메시지를 보여주거나 명령어 반향을 켜거나 끄 (ex. echo off : 명령을 표시하지 않음)

pause : 실행중인 배치 파일 일시 중지

set : set 명령은 MS-DOS 환경 변수를 화면에 보여주거나 지정하고 제거하는데 이용되는 명령



# cmd Command & Tool Explanation

1. promqry.exe : MS 배포 툴, 네트워크가 프리미어스큐어스 모드인지 아닌지 여부를 확인
2. arp : IP주소에 대한 MAC주소 확인
3. net user : 계정 정보 출력
4. pslist.exe : 프로세스 목록 및 정보 표시
5. handle.exe : 핸들 프로세스 확인
6. Autorunsc.exe : 자동 실행되는 프로세스 확인
7. FDpro.exe : 메모리 덤프 툴
8. tcp dump.exe : 패킷 캡처 툴

# Code review - 1

```
9  :: intro
10  set _ver=0.0.1
11  title Incident Response Collect Script for Live Response %_ver%
12
13  ::Color config
14  cls
15  color 03
16
17  :: Banner
18  @echo *****
19  @echo ***** Incident Response Script %_ver% *****
20  @echo *****
21
22  goto :main
23
24  :main|
25  ::System Default imformation
26  set _date=%date%
27  set _HOSTNAME=%COMPUTERNAME%
28  set _SYSDRIVE=%SYSTEMDRIVE%\
29  set _ToolPath=toolkit
30  if "%PROCESSOR_ARCHITECTURE%" == "x86" set arch=32
31  if "%PROCESSOR_ARCHITECTURE%" == "AMD64" set arch=64
32
33  :: Artifact Output Path
34  set /p outputPath=[+] Output Path :
35
```

관리자: Incident Response Collect Script for Live Response 0.0.1

```
*****
**** Incident Response Script 0.0.1 ****
*****
+] Output Path :
```

# Code review - 1

```
24 :main|
25 ::System Default imformation
26 set _date=%date%
27 set _HOSTNAME=%COMPUTERNAME%
28 set _SYSDRIVE=%SYSTEMDRIVE%
29 set _ToolPath=toolkit
30 if "%PROCESSOR_ARCHITECTURE%" == "x86" set arch=32
31 if "%PROCESSOR_ARCHITECTURE%" == "AMD64" set arch=64
32
33 :: Artifact Output Path
34 set /p outputPath=[+] Output Path :
35
36 ::Memory Dump Yes or No
37 set /p isMemoryDump=[+] Memory Dump Execute? (y/n) :
38 if "%isMemoryDump%" == "y" (
39     set isMemoryDump=yes
40 ) else (
41     set isMemoryDump=no
42 )
43
44 :: Network Packet Dump Yes or No
45 set /p isPacketDump=[+] Packet Dump Execute? (y/n) :
46 if "%isPacketDump%" == "y" (
47     set isPacketDump=yes
48 ) else (
49     set isPacketDump=no
50 )
51
52 :: Examiner Name & Output Path
53 set /p examiner=[+] Examiner:
54 set _path=%outputPath%\%_HOSTNAME%_%date%_%Examiner%
55 mkdir %_path%
```

관리자: Incident Response Collect Script for Live Response 0.0.1

```
*****
***** Incident Response Script 0.0.1 *****
*****
```

```
[+] Output Path : anaylze
[+] Memory Dump Execute? (y/n) :n
[+] Packet Dump Execute? (y/n) :n
[+] Examiner:kimwoojong
```

Wingan\OneDrive\바탕 화면\포렌식 도구개발 툴

보기

PC > 바탕 화면 > 포렌식 도구개발 툴

이름	수정한 날짜
anaylze	2019-10-05 오후 5:23
test	2019-10-05 오후 4:46
toolkit	Wingan\OneDrive\바탕 화면\포렌식 도구개발 툴
tool-list	보기
vol	
whay	PC > 바탕 화면 > 포렌식 도구개발 툴 > anay
autoanalysis.bat	이름
fcopy.exe	
test.bat	COW8311_2019-10-05_kimwoojong

# Code review - 1

```
56
57 echo Computer information
58 echo Packet Dump Execute : %isPacketDump%
59 echo Memory Dump Execute : %isMemoryDump%
60 echo Path : %outputPath%
61 echo DATE : %_date%
62 echo Host Name : %_HOSTNAME%
63 echo Tool Path : %_Toolpath%
64 echo Arcitecture : %arch%
65
66 pause
```

관리자: Incident Response Collect Script for Live Response 0.0.1

```
*****
***** Incident Response Script 0.0.1 *****
*****
[+] Output Path : anaylze
[+] Memory Dump Execute? (y/n) :n
[+] Packet Dump Execute? (y/n) :n
[+] Examiner: kimwoojong
Computer information
Packet Dump Execute : no
Memory Dump Execute : no
Path : anaylze
DATE : 2019-10-05
Host Name : COW8311
Tool Path : toolkit
Arcitecture : 64
계속하려면 아무 키나 누르십시오 . . .
```



# Code review - 1

```
73 ::Network
74 if not exist %_path%\network (
75     mkdir %_path%\network
76     set network_PATH=%_path%\network
77 )
78
79 :: promqry.exe, Promiscuous Mode Detector
80 echo 1_ Checking for Promiscuous Mode
81 %_ToolPath%\SIS\promqry.exe > %network_PATH%\promiscuous_detect_promqry.txt
82
83 echo 2_ Checking for ARP Cache
84 arp -a > %network_PATH%\arp_a.txt
85
86 echo 3_ Checking for Net User
87 net user > %network_PATH%\net_user.txt
88
89
90
91 ::Process
92 if not exist %_path%\process (
93     mkdir %_path%\process
94     set process_PATH=%_path%\process
95 )
96
97 :: pslist.exe, process list
98 echo 2_1_ Checking for Process
99 %_ToolPath%\SIS\pslist.exe /accepteula > %process_PATH%\pslist.txt
100 %_ToolPath%\SIS\handle.exe /accepteula > %process_PATH%\handle.txt
101 %_ToolPath%\SIS\autorunsc.exe /accepteula > %process_PATH%\autorunsc.txt
102
```

이름	수정한 날짜	유형
network	2019-10-05 오후 5:51	파일 폴더
process	2019-10-05 오후 5:51	파일 폴더

이름	수정한 날짜	유형
arp_a.txt	2019-10-05 오후 5:51	텍스트 문
net_user.txt	2019-10-05 오후 5:51	텍스트 문
promiscuous_detect_promqry.txt	2019-10-05 오후 5:51	텍스트 문

arp\_a.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움(H)

인터페이스: 192.168.25.40 --- 0x5

인터넷 주소	물리적 주소	유형
192.168.25.1	00-08-52-14-e5-c7	동적
192.168.25.7	70-2c-1f-0f-ef-cd	동적
192.168.25.22	00-16-6c-c7-dc-10	동적
192.168.25.54	f0-03-8c-7d-89-59	동적
192.168.25.63	ff-ff-ff-ff-ff-ff	정적
224.0.0.22	01-00-5e-00-00-16	정적
224.0.0.251	01-00-5e-00-00-fb	정적
224.0.0.252	01-00-5e-00-00-fc	정적
239.255.255.250	01-00-5e-7f-ff-fa	정적
255.255.255.255	ff-ff-ff-ff-ff-ff	정적

인터페이스: 192.168.153.1 --- 0x15

인터넷 주소	물리적 주소	유형
192.168.153.254	00-50-56-fc-9d-4d	동적
192.168.153.255	ff-ff-ff-ff-ff-ff	정적
224.0.0.22	01-00-5e-00-00-16	정적
224.0.0.251	01-00-5e-00-00-fb	정적
224.0.0.252	01-00-5e-00-00-fc	정적
239.255.255.250	01-00-5e-7f-ff-fa	정적
255.255.255.255	ff-ff-ff-ff-ff-ff	정적


# Code review - 1

```
104  ::Memory
105  if not exist %_path%\memory (
106      mkdir %_path%\memory
107      set memory_PATH=%_path%\memory
108  )
109
110  echo 3_1_Dumping Memory
111  if "%isMemoryDump%" == "yes" (
112      goto :acquire_memoryDump
113  ) else (
114      goto :packet dump
115  )
116  echo %isMemoryDump%
117
118  :acquire_memoryDump
119  %_ToolPath%\memory\FDPro.exe %memory_PATH%\memdump.dd
```

ingan\OneDrive\바탕 화면\포렌식 도구개발 툴\analyze\COW8311\_2019-10-05\_kimwoojong\

보기

C > 바탕 화면 > 포렌식 도구개발 툴 > analyze > COW8311\_2019-10-05\_kimwoojong > |

이름	수정된 날짜	유형
 memdump.dd	2019-10-05 오후 6:13	DD 파일

# Code review - 1

```
137 :SELECT_NIC
138 set /p _NIC=what's the NIC number you want to acquire (1.2.3...)? || goto :SELECT_NIC
139
124 mkdir %_path%\Packet_Dump
125 set packetdump_PATH=%_path%\Packet_Dump
126
126 what's the NIC number you want to acquire (1.2.3...)? 8
127 2019-10-05 18:23:20.56 - created "Packet_Dump" directory in W
128 2019-10-05 18:23:20.56 - Packet Dump Start
129
130 *****
131 **
131 ** Tcpdump v4.5.1 (Nov 20, 2013) for Windows **
132 ** Win98/ME/NT4/2000/XP/2003/Vista/2008/Win7/Win8/Win2012 **
133 **
133 ** built with Microolap Packet Sniffer SDK v6.1 and **
134 ** Microolap WinPCap to Packet Sniffer SDK migration module. **
135 **
136 ** (c) Microolap Technologies, **
137 ** Khalturin A.P. & Naumov D.A. **
138 ** http://www.microolap.com **
139 **
139 ** Trial license. **
140 **
141 *****
142
143 *****
144 **
144 ** Tcpdump v4.5.1 (Nov 20, 2013) for Windows **
145 ** Win98/ME/NT4/2000/XP/2003/Vista/2008/Win7/Win8/Win2012 **
146 **
146 ** built with Microolap Packet Sniffer SDK v6.1 and **
147 ** Microolap WinPCap to Packet Sniffer SDK migration module. **
148 **
149 ** (c) Microolap Technologies, **
```



# Volatility

볼라틸리티란 ?

Python 기반 Windows Memory Forensic Tool

Windows, Linux, Mac OS에서 실행할 수 있음

Open source 이며, Plugin 형태로 다양한 기능들을 제공

플러그인을 자신이 직접 만들어 사용가능

메모리 덤프 파일(img, raw, dmp), 하이버네이션 파일(hiber), 가상 머신 메모리(vmem) 분석가능



# Volatility

## 볼라틸리티로 획득할 수 있는 정보

트리 형태의 프로세스 리스트, 프로세스가 로드한 DLL과 핸들, 네트워크 정보, 시스템에서 로드했던 드라이버 목록

실행 중이거나 종료 또는 루트킷으로 은닉된 프로세스의 오프셋, SID(보안 식별자), PID, 스레드 수, 핸들 수, 시작 및 종료시간



# Volatility

## Volatility Command

psscan, pslist, pstree : 프로세스 정보 확인

memmap : 프로세스 메모리 분석

memdump : 메모리 덤프

connscan : 네트워크 정보 확인

Handels : 핸들 정보 확인

# Code review - 2

```
9  :: intro
10 title Autovol%_ver%
11
12 set _ver=0.1
13
14 :: color config
15 cls
16 color 03
17
18 :: main
19 goto :main
20
21 :main
22 :: System Default Information
23 set _date=%date%
24 set _HOSTNAME=%COMPUTERNAME%
25 set _SYSDRIVE=%SYSTEMDRIVE%
26 set _Tool=vol.py
27 set _SamplePath=sample
28 if "%PROCESSOR_ARCHITECTURE%" == "x86" set arch=32
29 if "%PROCESSOR_ARCHITECTURE%" == "AMD64" set arch=64
30
31 :: Banner
32 @echo *****
33 @echo                               Autovol
34 @echo *****
35
36 echo *****Computer Information*****
37 echo      date : %date%
38 echo      Computer Name : %COMPUTERNAME%
39 echo      System Drive : %SYSTEMDRIVE%
40 echo      Process Architecture : %PROCESSOR_ARCHITECTURE%
41 echo      Tool Name : %_Tool%
42 echo      Path : %outputPath%
43 echo *****
```

관리자: Autovol0.1

```
*****
                               Autovol
*****
*****Computer Information*****
      date : 2019-10-06
      Computer Name : COW8311
      System Drive : C:
      Process Architecture : AMD64
      Tool Name : vol.py
      Path : test
*****
```

Tool name – vol.py

이 부분을 제외하면  
Code review – 1 부분이랑 같습니다!!

## Code review - 2

```
46 set /p outputPath=[+] Output Path:
47 mkdir "%outputPath%"
48
49 set /p file=Input analysis file :
50 :: imageinfo know
51 set /p imageinfo=Do you know the profile of this file? (Yes/No) :
52 if "%imageinfo%" == "Yes" (
53     set /p profile=Input profile :
54     goto :menu
55 ) else (
56     goto :imageinfo
57 )
```

```
1_ imageinfo
Volatility Foundation Volatility Framework 2.6
INFO      : volatility.debug      : Determining profile based on KDBG search...
           Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86, Win7SP1x86
           AS Layer1            : IA32PagedMemoryPae (Kernel AS)
           AS Layer2            : FileAddressSpace (C:\Users\Wingan\OneDrive\H
olatility-master\sample\SCP.vmem)
           PAE type             : PAE
           DTB                  : 0x185000L
           KDBG                 : 0x82b7ac28L
           Number of Processors : 2
           Image Type (Service Pack) : 1
           KPCR for CPU 0       : 0x82b7bc00L
           KPCR for CPU 1       : 0x807ca000L
           KUSER_SHARED_DATA     : 0xffdf0000L
           Image date and time   : 2019-10-06 08:16:09 UTC+0000
           Image local date and time : 2019-10-06 01:16:09 -0700
           Input profile         : Win7SP0x86
```



# Code review - 2

```
58 ::menu
59 :menu
60 echo What do you want?
61 set /p num=0.profile 1.imageinfo 2.psscan 3.pslist 4.pstree 5.filescan 6.memmap 7.memdump 8.connsnscan 9.handles :
62 if "%num%"=="0" (
63     goto :profile
64 )
65 if "%num%"=="1" (
66     goto :imageinfo
67 )
68 if "%num%"=="2" (
69     goto :psscan
70 )
71 if "%num%"=="3" (
72     goto :pslist
73 )
74 if "%num%"=="4" (
75     goto :pstree
76 )
77 if "%num%"=="5" (
78     goto :filesnscan
79 )
80 if "%num%"=="6" (
81     goto :memmap
82 )
83 if "%num%"=="7" (
84     goto :memdump
85 )
86 if "%num%"=="8" (
87     goto :connsnscan
88 )
89 if "%num%"=="9" (
90     goto :handles
91 ) else (
92     goto :quit
93 )
```

What do you want?

0.profile 1.imageinfo 2.psscan 3.pslist 4.pstree 5.filescan 6.memmap 7.memdump 8.connsnscan 9.handles



코드가 더러워  
죄송합니다...;;

# Code review - 2

```
95  :: profile
96  :profile
97  echo 0_profile
98  set /p profile = Input profile :
99  goto :menu
100
101  :: imageinfo
102  :imageinfo
103  echo 1_imageinfo
104  python %_Tool% -f %_SamplePath%\%file% imageinfo
105  set /p profile=Input profile :
106  goto :menu
107
108  :: psscan
109  :psscan
110  echo 2_psscan
111  python %_Tool% -f %_SamplePath%\%file% --profile="%profile%" psscan
112  goto :menu
113
114  :: pslist
115  :pslist
116  echo 3_pslist
117  python %_Tool% -f %_SamplePath%\%file% --profile="%profile%" pslist
118  goto :menu
119
```

```
1_ imageinfo
Volatility Foundation Volatility Framework 2.6
INFO : volatility.debug : Determining profile based on KDBG search...
Suggested Profile(s) : Win7SP1x86_23418, Win7SP0x86, Win7SP1x86
AS Layer1 : IA32PagedMemoryPae (Kernel AS)
```

olatility-master

Number  
Image Type

KLUS

Image

Image local

Input profile :

Offset(V)	Name	PID	PPID	Thds	Hnds	Sess	Wow64	Start	Exit
0x858d1020	System	4	0	91	511	-----	0	2019-10-06 08:14:34 UTC+0000	
0x864a08b8	smss.exe	252	4	2	30	-----	0	2019-10-06 08:14:34 UTC+0000	
0x86bc5600	csrss.exe	348	332	9	466	0	0	2019-10-06 08:14:35 UTC+0000	
0x86dd8130	wininit.exe	428	332	4	82	0	0	2019-10-06 08:14:35 UTC+0000	
0x86dd8c78	csrss.exe	436	420	10	189	1	0	2019-10-06 08:14:35 UTC+0000	
0x875fc5c0	services.exe	484	428	11	223	0	0	2019-10-06 08:14:35 UTC+0000	
0x87610d40	winlogon.exe	516	420	5	114	1	0	2019-10-06 08:14:35 UTC+0000	
0x876105a8	lsass.exe	528	428	9	553	0	0	2019-10-06 08:14:35 UTC+0000	
0x86dd7d40	lsass.exe	536	428	10	140	0	0	2019-10-06 08:14:35 UTC+0000	
0x87646b50	svchost.exe	664	484	12	361	0	0	2019-10-06 08:14:36 UTC+0000	
0x87685030	vmacthlp.exe	724	484	4	54	0	0	2019-10-06 08:14:36 UTC+0000	
0x876a9458	svchost.exe	772	484	7	276	0	0	2019-10-06 08:14:36 UTC+0000	
0x876e2030	svchost.exe	864	484	18	379	0	0	2019-10-06 08:14:36 UTC+0000	
0x876ee5c8	svchost.exe	896	484	20	386	0	0	2019-10-06 08:14:36 UTC+0000	
0x876f55f8	svchost.exe	928	484	40	776	0	0	2019-10-06 08:14:36 UTC+0000	
0x877135d8	audiodg.exe	1016	864	5	124	0	0	2019-10-06 08:14:36 UTC+0000	
0x87726438	svchost.exe	1084	484	14	552	0	0	2019-10-06 08:14:36 UTC+0000	
0x87743598	svchost.exe	1168	484	19	357	0	0	2019-10-06 08:14:36 UTC+0000	
0x87781670	spoolsv.exe	1288	484	14	277	0	0	2019-10-06 08:14:36 UTC+0000	
0x8779d290	svchost.exe	1332	484	19	322	0	0	2019-10-06 08:14:36 UTC+0000	
0x877e39a8	svchost.exe	1432	484	9	172	0	0	2019-10-06 08:14:37 UTC+0000	
0x87808158	VGAAuthService.	1480	484	3	86	0	0	2019-10-06 08:14:37 UTC+0000	
0x8783c030	vmtoolsd.exe	1556	484	10	272	0	0	2019-10-06 08:14:37 UTC+0000	
0x878e3600	taskhost.exe	1920	484	10	159	1	0	2019-10-06 08:14:37 UTC+0000	
0x878ed3d0	svchost.exe	2012	484	6	94	0	0	2019-10-06 08:14:37 UTC+0000	
0x87940338	spssvc.exe	356	484	5	151	0	0	2019-10-06 08:14:38 UTC+0000	
0x8782ca58	dllhost.exe	744	484	21	203	0	0	2019-10-06 08:14:38 UTC+0000	
0x87833d40	dllhost.exe	1396	484	17	206	0	0	2019-10-06 08:14:39 UTC+0000	
0x8781c030	msdtc.exe	692	484	16	156	0	0	2019-10-06 08:14:39 UTC+0000	
0x879ad860	WmiPrivSE.exe	2132	664	11	192	0	0	2019-10-06 08:14:40 UTC+0000	



# Code review -

```
138 :: memdump
139 :memdump
140 echo 7_ memdump
141 set /p _pid= Input Pid :
142 python %_Tool% -f %_SampleP
143 strings %outputPath%\%_pid%
144 goto :menu
145
```

```
0.profile 1.imageinfo 2.psscan 3.pslist
7_ memdump
Input Pid : 2456
```

psscan.txt - Windows 메모장

파일(F) 편집(E) 서식(O) 보기(V) 도움말

Offset(P)	Name	PID	PPID	PDB
0x00000000052d1020	System	4	0	
0x00000000bd4190b0	StikyNot.exe	2456	23	
0x00000000bd41fd40	vmtoolsd.exe	2448	23	
0x00000000bd472a48	SearchIndexer.	2604	48	
0x00000000bd4b2d40	SearchProtocol	2708	20	
0x00000000bd4c0730	SearchFilterHo	2728	26	
0x00000000bd507d40	notepad.exe	2848	23	
0x00000000bd5285c0	WmiPrvSE.exe	2928	6	
0x00000000bd549d40	WmiApSrv.exe	2980	13	
0x00000000bde08158	VGAAuthService.	1480	4	

```
11822 HCAMP{THIS_IS_NOT_FLAG}
11823 0Sp
11824 0Sp
11825 0Sp1
11826 0Sp
11827 0Sp
11828 1Spn
11829 1Sp
11830 1Sp`
11831 1Sp
11832 Gal
11833 1SpS
11834 oe Print
11835 1Sp
11836 Segoe Print Bold
11837 1Sp
11838 1Sp
11839 Bold
11840 dlic
11841 c2on conl
11842 ,2,-1,
11843 1)"/>
```

## Find Results

Address	Value
Found 13 occurrences of 'HCAMP'.	
Line 11822	HCAMP{THIS IS NOT FLAG}
Line 14827	HCAMP{
Line 24067	{W**qgenerator Msftedit 5.41.2...0Wtx11520Whighlight0Wf0Wfs22 HCAMPW{THIS IS NOT...Wlanq9Wf1Wpar
Line 24069	HCAMP{THIS IS NOT FLAG}
Line 24076	x5040Wtx5400Wtx5760Wtx6120Wtx6480...11160Wtx11520Whighlight0Wf0Wfs22 HCAMPW{Wlanq9Wf1Wpar
Line 24077	HCAMP{
Line 24087	{W**qgenerator Msftedit 5.41.2...0Wtx11520Whighlight0Wf0Wfs22 HCAMPW{THIS IS NOT...Wlanq9Wf1Wpar
Line 24099	{W**qgenerator Msftedit 5.41.21.2510;...0Wtx11160Wtx11520Whighlight0Wf0Wfs22 HCAMPW{Wlanq9Wf1Wpar
Line 26528	00Wtx11160Wtx11520Whighlight0Wf0Wfs22 HCAMPW{THIS IS NOT FLAGW{Wlanq9Wf1Wpar
Line 26579	HCAMP{
Line 26584	HCAMP{THIS IS NOT FLAG}
Line 759916	x5040Wtx5400Wtx5760Wtx6120Wtx6480...11160Wtx11520Whighlight0Wf0Wfs22 HCAMPW{Wlanq9Wf1Wpar

660KB

,774KB

## Code review - 2

```
158 :: quit
159 :quit
160 set /p stop=Do you want stop Volatility? (Yes/No) :
161 if "%stop%" == "Yes" (
162     echo *****Computer Information*****
163     echo         date : %date%
164     echo         Computer Name : %COMPUTERNAME%
165     echo         System Drive : %SYSTEMDRIVE%
166     echo         Process Architecture : %PROCESSOR_ARCHITECTURE%
167     echo         Tool Name : %_Tool%
168     echo         Path : %outputPath%
169     echo *****
170     pause
171 ) else (
172     goto :menu
173 )
```

What do you want?  
0.profile 1.imageinfo 2.psscan 3.pslist 4.pstree 5.filescan 6.memmap 7.memdump 8.connsnscan 9.handles : exit  
Do you want stop Volatility? (Yes/No) :Yes  
\*\*\*\*\*Computer Information\*\*\*\*\*  
date : 2019-10-06  
Computer Name : COW8311  
System Drive : C:  
Process Architecture : AMD64  
Tool Name : vol.py  
Path : SCP  
\*\*\*\*\*  
계속하려면 아무 키나 누르십시오 . . .  
C:\Users\Wingan\OneDrive\바탕 화면\Tool\2019-09-21\vol\volatility-2.6\volatility-master>



## 예제

누군가 메모장에 비밀 메시지를 기록해놓았다.

비밀 메시지에 내용을 찾아라

플래그 포맷 : C0WBB3LLCTF{CONTENT}

힌트 : 메모장은 프로세스에 올라갈 때 notepad.exe로 올라간다.



# 예제

관리자: Autovol0.1

\*\*\*\*\*

\*\*\*\*\*Computer Information\*\*\*\*\*

date : 2019-10-06  
Computer Name : COW8311  
System Drive : C:  
Process Architecture : AMD64  
Tool Name : vol.py  
Path : SCP

\*\*\*\*\*

[+] Output Path:SCP

Input analysis file : SCP.vmem

Do you know the profile of this file? (Yes/No) :No

1\_ imageinfo

Volatility Foundation Volatility Framework 2.6

INFO : volatility.debug : Determining profile based on KDBG search...

Suggested Profile(s) : Win7SP1x86\_23418, Win7SP0x86, Win7SP1x86

AS Layer1 : IA32PagedMemoryPae (Kernel AS)

AS Layer2 : FileAddressSpace (C:\Users\wingan\OneDrive\바탕 화면\Tool\2019-09-21\vol\volatility-2.6\volatility-master\sample\SCP.vmem)

PAE type : PAE

DTB : 0x185000L

KDBG : 0x82b7ac28L

Number of Processors : 2

Image Type (Service Pack) : 1

KPCR for CPU 0 : 0x82b7bc00L

KPCR for CPU 1 : 0x807ca000L

KUSER\_SHARED\_DATA : 0xffdf0000L

Image date and time : 2019-10-06 08:16:09 UTC+0000

Image local date and time : 2019-10-06 01:16:09 -0700

Input profile : Win7SP1x86\_23418\_

# 예제

```
What do you want?  
0.profile 1.imageinfo 2.psscan 3.pslist 4.pstree 5.filescan 6.memmap 7.memdump 8.connsnscan 9.handles : 3  
3_ pslist
```





0x88307d40	notepad.exe	2848	2352	4	79	1	0	2019-10-06 08:14:49 UTC+0000
0x883285c0	WmiPrvSE.exe	2928	664	12	290	0	0	2019-10-06 08:14:58 UTC+0000
0x88349d40	WmiApSrv.exe	2980	484	7	119	0	0	2019-10-06 08:14:58 UTC+0000



# 예제

```
7_ memdump
Input Pid : 2848
Volatility Foundation Volatility Framework 2.6
*****
Writing notepad.exe [ 2848] to 2848.dmp

Strings v2.53 - Search for ANSI and Unicode strings in binary images.
Copyright (C) 1999-2016 Mark Russinovich
Sysinternals - www.sysinternals.com
```

 2848.dmp	2019-10-07 오전 2:52	DMP 파일	422,388KB
 2848.txt	2019-10-07 오전 2:52	텍스트 문서	26,243KB

# 예제

```

1 4qQ
2 C:\Windows\system32\SHELL32.dll
3 C:\Windows\system32\SHELL32.dll.124.Config
4 C:\Windows\system32\
5 en-US
6 3;0
7 0123456789
8 3;0
9 h:mm:ss tt
10 h:mm tt
11 M/d/yyyy
12 MMMM, yyyy
13 dddd, MMMM dd, yyyy
14 RESCDIR
15 I~2
16 C:\Windows\rescache
17 2fT3
18 ^K[
19 vn
20 ^QJ
21 9g:
22 ~^÷
23 Eh`
24 't<5
25 q-k,I
26 ;%`
27 0V!
28 fEc
29 YZV
30 9\`t
31 f%□
32 G
33 Ye2
34 m&S
35 M4=
36 cWs
37 z=ü
38
39 2"Q
40 E'~

```

Find	
Text: ^	COWB3LL
Find Results	
Address	Value
Found 2 occurrences of 'COWB3LL'.	
Line 7672	COWB3LLCTF{y0u_c4n_U53_V0latili7y}
Line 1220589	COWB3LLCTF{y0u_c4n_U53_V0latili7y}

```

7667 }H=
7668 @!9
7669 p*"
7670 }e0
7671 _D6
7672 COWB3LLCTF{y0u_c4n_U53_V0latili7y}
7673 |z5
7674 9p8
7675 +00
7676 /C:\
7677 9lidadeFilenameChars
7678 }b(
7679 8Xp

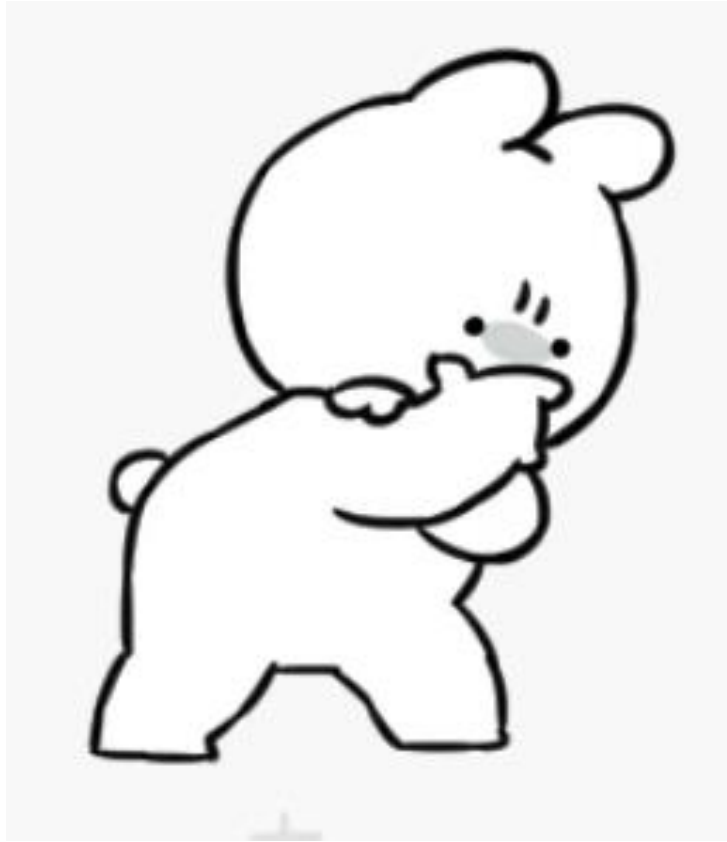
```

NEXT

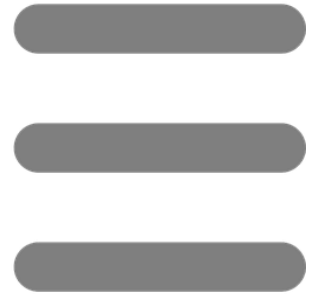


# Q&A

---



**Q.**



**A.**

