

207 – Living

Team Information

Team Name : DogeCoin

Team Member : Dongbin Oh, Donghyun Kim, Donghyun Kim, Yeongwoong Kim

Email Address : dfc-dogecoin@naver.com

Instructions

Description Analyze the given artifacts and answer the questions.

Target	Hash (MD5)
living.zip	A9BEEE8574DF75AEF2E5D40AB16D4C9C

Questions

1. The first time the “Living Off The land attack technique” was executed on the PC is 2021-07-25 01:04:20 (UTC+9). Check the ADS file(s) created after the first run. (50 points)
 - Absolute path of the ADS file, ADS name, creation time required
2. The “calc.exe” hidden in the ADS file(s) was executed with various programs. Check the traces of the execution. (150 points)
 - Execution time, executed file (*.exe), related calc file inside ADS

Teams must:

- Develop and document the step-by-step approach used to solve this problem to allow another examiner to replicate team actions and results.
- Specify all tools used in deriving the conclusion(s).

Tools used:

Name:	WinPrefetchView	Publisher:	Nirsoft
Version:	1.36		
URL:	https://www.nirsoft.net/utils/win_prefetch_view.html		

Name:	PECmd	Publisher:	Eric Zimmerman
Version:	1.4.0.0		
URL:	https://ericzimmerman.github.io/#!index.md		

Name:	MFTECmd	Publisher:	Eric Zimmerman
Version:	0.5.0.1		
URL:	https://ericzimmerman.github.io/#!index.md		

Name:	CSVFileView	Publisher:	Nirsoft
Version:	2.54		
URL:	https://www.nirsoft.net/utils/csv_file_view.html		

Name:	NTFS Log Tracker	Publisher:	Blue angel
Version:	1.6		
URL:	https://sites.google.com/site/forensicnote/ntfs-log-tracker		

Name:	REGA	Publisher:	DFRC
Version:	1.5.3.0		
URL:	http://forensic.korea.ac.kr/tools.html		

Name:	Registry Explorer	Publisher:	Eric Zimmerman
Version:	1.6.0.0		
URL:	https://ericzimmerman.github.io/#!index.md		

Name:	WindowsTimeline Parser	Publisher:	Kacos2000
Version:	2.0.81.0		
URL:	https://github.com/kacos2000/WindowsTimeline		

Name:	Message Analyzer	Publisher:	Microsoft
Version:	1.4		
URL:	https://github.com/riverar/messageanalyzer-archive		

Name:	SrumECmd	Publisher:	Eric Zimmerman
Version:	0.5.0.2		
URL:	https://github.com/riverar/messageanalyzer-archive		

Name:	BitsParser	Publisher:	FireEye
Version:	-		
URL:	https://github.com/fireeye/BitsParser		

Step-by-step methodology:

1. The first time the “Living Off The land attack technique” was executed on the PC is 2021-07-25 01:04:20 (UTC+9). Check the ADS file(s) created after the first run. (50 points)

1.1. 문제 분석

“Living Off The land attack” 기법은 Windows에 기본적으로 설치되어 있는 프로그램 또는 스크립트를 활용하여 악성코드의 실행, 은닉 등을 하는 것을 의미한다.

본 문제에서 주어진 이미지는 KAPE를 이용하여 선별 수집되었으며, 다음과 같은 아티팩트가 남아 있는 것을 확인하였다.

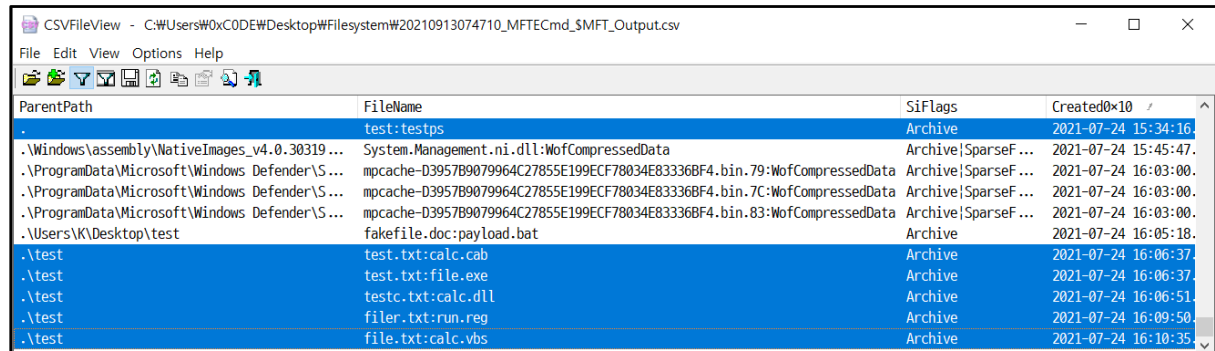
- **Filesystem Artifacts** [파일/폴더에 대한 이력 및 정보]
 - **\$MFT** [파일/폴더의 정보]
 - **\$LogFile, \$J** [파일/폴더에 대한 이력]
- **Prefetch** [프로그램 실행, 참조한 파일 목록]
- **Hive files** [레지스트리 정보]
 - **System**
 - **Software**
 - **NTUSER.DAT**
 - **AmCache.hve**
 - **UsrClass.dat**
- **Timeline** [프로그램 실행 및 파일 열람]
- **EventLog** [이벤트 목록]
- **Event Trace Log (ETL)** [성능 모니터링 이벤트 목록]
- **Windows Error Reporting (WER)** [프로그램 실행]
- **System Resource Usage Monitor (SRUM)** [프로그램 리소스 사용량]
- **BITS database** [파일 업로드/다운로드 기록]

상기 목록에 있는 아티팩트를 분석하여, ADS에 은닉된 파일 및 calc.exe 실행 흔적을 확인하였다. 이중, 문제와 관련된 흔적을 확인할 수 있는 아티팩트는 **붉은색**으로 표기하였다.

1.2. ADS에 은닉된 파일 식별

1.2.1. \$MFT 분석

MFTECMD로 \$MFT를 분석하여, ADS에 은닉된 파일 (Data Stream)을 확인하였다.



ParentPath	FileName	SiFlags	Created0x10
.	test: testps	Archive	2021-07-24 15:34:16
.\Windows\assembly\NativeImages_v4.0.30319...	System.Management.ni.dll:WofCompressedData	Archive SparseF...	2021-07-24 15:45:47
.\ProgramData\Microsoft\Windows Defender\S...	mpcache-D3957B9079964C27855E199ECF78034E83336BF4.bin.79:WofCompressedData	Archive SparseF...	2021-07-24 16:03:00
.\ProgramData\Microsoft\Windows Defender\S...	mpcache-D3957B9079964C27855E199ECF78034E83336BF4.bin.7C:WofCompressedData	Archive SparseF...	2021-07-24 16:03:00
.\ProgramData\Microsoft\Windows Defender\S...	mpcache-D3957B9079964C27855E199ECF78034E83336BF4.bin.83:WofCompressedData	Archive SparseF...	2021-07-24 16:03:00
.\Users\K\Desktop\test	fakefile.doc:payload.bat	Archive	2021-07-24 16:05:18
.\test	test.txt:calc.cab	Archive	2021-07-24 16:06:37
.\test	test.txt:file.exe	Archive	2021-07-24 16:06:37
.\test	testc.txt:calc.dll	Archive	2021-07-24 16:06:51
.\test	filer.txt:run.reg	Archive	2021-07-24 16:09:50
.\test	file.txt:calc.vbs	Archive	2021-07-24 16:10:35

[그림 1] \$MFT에 기록된 ADS 흔적

생성된 ADS 중에는 “WofCompressedData”라고해서 Sparse 속성으로 생성되는 Data Stream 이 있다. 해당 Data Stream 은 시스템 파일의 크기를 줄이는 목적으로 Windows OS 에서 생성하는 것이다 (Windows 10 이후 도입된 Compact OS 기능)¹. 따라서, 해당 Data Stream 은 “Living Off The land attack”으로 인하여 생성된 것으로 판단하지 않았다.

생성된 ADS 흔적을 정리하면 다음 표와 같으며, “Living Off The land attack”이 발생한 시각 이후로 발생한 것은 초록색으로 표현하였다. ADS 의 생성 시각은 \$MFT 의 \$STANDARD_INFORMATION 속성에 기록된 생성 시각으로부터 가져왔다. ADS 의 경로는 ADS 이름을 포함한 전체 경로로 표현하였다.

Creation Time (UTC+9)	ADS Name (Named Data Stream)
2021-07-25 00:34:16.2205756	C:\test:testps
2021-07-25 01:05:18.5599722	C:\Users\K\Desktop\test\fakefile.doc:payload.bat
2021-07-25 01:06:37.7629604	C:\test\test.txt:calc.cab
2021-07-25 01:06:37.7629604	C:\test\test.txt:file.exe
2021-07-25 01:06:51.7162617	C:\test\testc.txt:calc.dll
2021-07-25 01:09:50.5128058	C:\test\filer.txt:run.reg
2021-07-25 01:10:35.2789090	C:\test\file.txt:calc.vbs

[표 1] \$MFT로부터 확인한 ADS로 은닉된 파일

¹ What is WofCompressedData? Does WOF mean that Windows is a dog?, <https://devblogs.microsoft.com/oldnewthing/20190618-00/?p=102597>

1.2.2. Prefetch 분석

\$MFT 분석의 한계는 Data Stream이 삭제되거나, ADS를 가지고 있던 파일/폴더가 삭제되어 MFT Entry가 다른 정보로 덮어 씌면 ADS 흔적을 확인할 수 없다. 하지만, Prefetch는 이러한 \$MFT 분석의 한계를 보완할 수 있다.

(1) ADS로 은닉된 파일이 열람되거나 실행되었다면, Prefetch의 참조한 파일 목록에 남아있을 수 있다. 또한, (2) ADS에 은닉된 파일이 실행 가능한 (Executable) 파일이라면, Prefetch 생성 폴더 (\Windows\Prefetch)에 Prefetch 파일 자체가 ADS에 은닉되어 생성될 수 있다.² (2)의 경우는 주어진 이미지에서 확인할 수 없었다.

(1)의 경우를 통해서 ADS에 은닉된 파일을 확인하였으며, 다음의 표와 같이 정리하였다. 시간 표현은 년/월/일을 제외하였다.

Last Run Time (UTC+9)	Process Name	참조 파일 목록 (ADS Only)
2021-07-25 01:04:46	CERTUTIL.EXE	C:\TEST:TESTPS
2021-07-25 01:06:12	MORE.COM	C:\TEST:TESTPS
		C:\USERS\K\DESKTOP\TEST\FAKEFILE.DOC:PAYLOAD.BAT
2021-07-25 01:08:28	RUNDLL32.EXE	C:\TEST\TESTC.TXT:CALC.DLL
2021-07-25 01:08:51	MAKECAB.EXE	C:\TEST\TEST.TXT:CALC.CAB
2021-07-25 01:09:50	REG.EXE	C:\TEST\FILER.TXT:RUN.REG
2021-07-25 01:11:14	FINDSTR.EXE	C:\TEST\FILE.TXT:CALC.VBS
		C:\TEST\TESTC.TXT:CALC.DLL
2021-07-25 01:11:57	WSCSCRIPT.EXE	C:\TEST\FILE.TXT:CALC.VBS
2021-07-25 01:12:06	CSCRIPT.EXE	C:\TEST\FILE.TXT:CALC.VBS
2021-07-25 01:19:08	CMD.EXE	C:\USERS\K\DESKTOP\TEST\FAKEFILE.DOC:PAYLOAD.BAT

[표 2] Prefetch로부터 확인한 ADS로 은닉된 파일

² Windows Prefetch File (PF) format, [https://github.com/libyal/libscca/blob/main/documentation/Windows%20Prefetch%20File%20\(PF\)%20format.asciidoc](https://github.com/libyal/libscca/blob/main/documentation/Windows%20Prefetch%20File%20(PF)%20format.asciidoc), 1.2.1. NTFS alternate data streams

1.2.3. 종합 정리

\$MFT와 Prefetch 분석을 통해 “Living Off The land attack”으로 인해 ADS에 은닉된 파일 목록을 정리하면 다음 표와 같다. 생성된 ADS의 이름 **붉은색**으로 표기하였다.

ADS Name (Named Data Stream)	Creation Time (UTC+9)	\$MFT	Prefetch
C:\Users\K\Desktop\test\fakefile.doc:payload.bat	2021-07-25 01:05:18.5599722	O	O
C:\test\test.txt:calc.cab	2021-07-25 01:06:37.7629604	O	O
C:\test\test.txt:file.exe	2021-07-25 01:06:37.7629604	O	X
C:\test\testc.txt:calc.dll	2021-07-25 01:06:51.7162617	O	O
C:\test\filer.txt:run.reg	2021-07-25 01:09:50.5128058	O	O
C:\test\file.txt:calc.vbs	2021-07-25 01:10:35.2789090	O	O

[표 3] Living Off The land attack 이후 생성된 ADS 목록

2. The “calc.exe” hidden in the ADS file(s) was executed with various programs.
Check the traces of the execution. (150 points)

2.1. 분석 목표

Execution time, executed file (*.exe), related calc file inside ADS

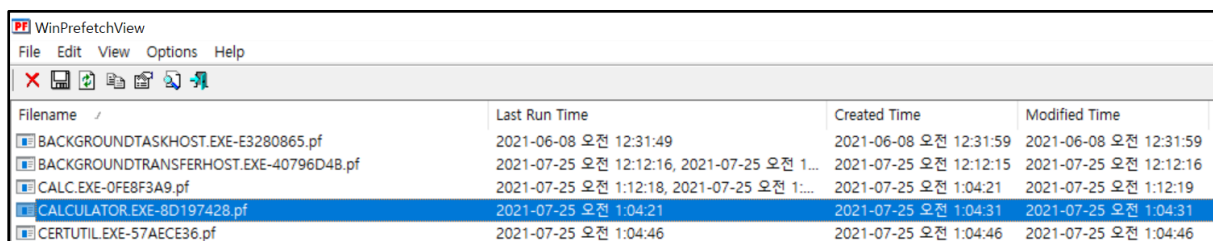
- 1) ADS에 은닉된 파일 (Stream)이 “어떤 프로그램”으로 “언제” 실행되었는지 확인
- 2) 어떤 ADS에 은닉된 파일 (Stream)이 calc.exe를 실행했는지 확인

2.2. 계산기 (calc.exe) 실행 흔적

Windows 10에서 계산기를 실행하면 calc.exe 뿐만 아니라 Windows App 버전 계산기 프로그램 (Microsoft.WindowsCalculator)이 실행될 수 있다. 또한, Calculator.exe 도 실행될 수 있는데, 해당 파일은 calc.exe가 최초로 실행될 때 Calculator.exe가 실행되는 것을 실험을 통해 확인하였다. 계산기를 실행하면 남을 수 있는 흔적의 형태는 다음과 같다.

- calc.exe
 - Prefetch
 - Hive: UserAssist, BAM
- Microsoft.WindowsCalculator
 - Timeline
 - Hive: UserAssist, BAM
- Calculator.exe
 - Prefetch

실제로, 주어진 이미지에서 Calculator.exe는 한 번만 실행되었다. 그러나, 실행 시각이 2021-07-25 01:04:21이어서, ADS에 은닉된 파일이 실행했다고 보기 어렵다. 왜냐하면, 최초로 ADS가 생성된 시각은 2021-07-25 01:05:18이므로 (Q1 답 참고), Calculator.exe 생성 시각이 더 과거이기 때문이다.



Filename	Last Run Time	Created Time	Modified Time
BACKGROUNDTASKHOST.EXE-E3280865.pf	2021-06-08 오전 12:31:49	2021-06-08 오전 12:31:59	2021-06-08 오전 12:31:59
BACKGROUNDTRANSFERHOST.EXE-40796D48.pf	2021-07-25 오전 12:12:16, 2021-07-25 오전 1...	2021-07-25 오전 12:12:15	2021-07-25 오전 12:12:16
CALC.EXE-0FE8F3A9.pf	2021-07-25 오전 1:12:18, 2021-07-25 오전 1...	2021-07-25 오전 1:04:21	2021-07-25 오전 1:12:19
CALCULATOR.EXE-8D197428.pf	2021-07-25 오전 1:04:21	2021-07-25 오전 1:04:31	2021-07-25 오전 1:04:31
CERTUTIL.EXE-57AECE36.pf	2021-07-25 오전 1:04:46	2021-07-25 오전 1:04:46	2021-07-25 오전 1:04:46

[그림 2] Calculator.exe 실행 횟수 및 실행 시각 확인

이러한 이유로, 본 문제를 해결하기 위해서는 calc.exe와 Microsoft.WindowsCalculator의 실행 흔적을 범위로 분석하였다.

2.3. ADS 흔적 식별

2.3.1. ADS 실행 및 생성 가능한 프로그램

Living Off The land attack에 활용되는 실행 파일과 스크립트는 LOLBAS³에 정리된 것을 기준으로, 공격자의 행위를 분석하였다. 본 절에서는 LOLBAS를 참고하여 ADS 실행 행위를 수행한 프로세스를 식별하고, 이를 바탕으로 calc.exe를 실행한 ADS를 특정한다.

ADS를 생성 또는 실행할 수 있는 실행 파일의 목록 중 본 시나리오에서 실행된 것은 다음과 같다. 실행 흔적은 Prefetch에서 확인하였다.

프로그램	생성 가능	실행 가능	대상 확장자 (in ADS)	비고
certutil.exe	O	X		
cmd.exe	O	O	.bat	
control.exe	X	O	.dll	
cscript.exe	X	O	.vbs	
esentutl.exe	O	X		
expand.exe	O	X		
extrac32.exe	O	O	.cab	CAB 압축 해제
findstr.exe	O	X		
forfiles.exe	X	O	.exe	
makecab.exe	O	X	.cab	CAB 압축
mavinject.exe	X	O	.dll	인젝션
MpCmdRun.exe	O	X		
mshta.exe	X	O	.hta, .js, .vbs	
reg.exe	O	X	.reg	
regedit.exe	O	O	.reg	
rundll32.exe	X	O	.dll	
sc.exe	X	O	.exe	
wmic.exe	X	O	.exe	
wscript.exe	O	O	.vbs	

[표 4] 본 시나리오에서 ADS 생성 또는 실행을 위해 사용할 수 있는 프로그램 목록

(1) ADS를 생성 또는 실행하기 위해서는 실행 파일을 실행할 수밖에 없으며, 이에 따라서 Prefetch에 관련

³ <https://lolbas-project.github.io/>

실행 파일의 흔적이 남는다.

“.reg” 확장자는 Registry에 정보를 기록하는 파일이다. (2) regedit.exe을 통해 .reg 파일을 실행한다면 Registry에 해당 흔적이 남으므로, 실행된 시점 이후로 영향 받은 Registry Key를 조사해야 한다.

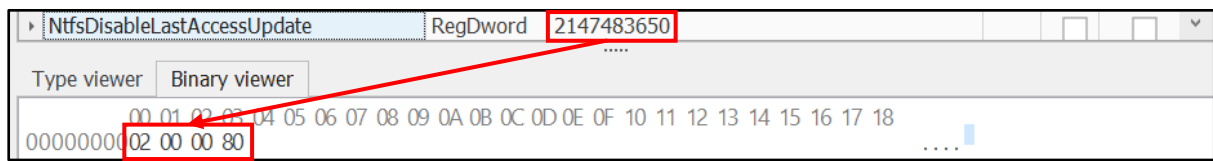
2.3.2. 생성된 ADS의 MAC 시각

추가된 Data Stream (ADS)도 시간 값 (MACE)을 가진다. \$STANDARD_INFORMATION과 \$FILENAME_INFORMATION 속성에 시간 값이 기록된다.

FileName	ParentPath	LastModified0x10	LastAccess0x10	Created0x10
fakefile.doc:payload.bat	.WUsers\WKW\Desktop\Wtest	2021-07-24 16:05:18.5599722	2021-07-24 16:06:29.4816653	2021-07-24 16:05:18.5599722
test.txt:file.exe	.Wtest	2021-07-24 16:08:51.2007680	2021-07-24 16:08:51.2007680	2021-07-24 16:06:37.7629604
test.txt:calc.cab	.Wtest	2021-07-24 16:08:51.2007680	2021-07-24 16:08:51.2007680	2021-07-24 16:06:37.7629604
testc.txt:calc.dll	.Wtest	2021-07-24 16:07:14.9817574	2021-07-24 16:08:28.8723148	2021-07-24 16:06:51.7162617

[그림 3] ADS에 기록된 시간 값 (\$STANDARD_INFORMATION)

NTFS에서 접근 시각의 갱신은 설정에 따라 갱신되지 않을 수 있으므로, Registry 값을 확인하였다⁴.



[그림 4] 접근 시각 갱신 관련 Registry 값 확인

NtfsDisableLastAccessUpdate 값은 다음과 같은 의미를 가진다. 본 시나리오에 제공된 Windows는 0x80000002 값을 가지며, 접근 시각이 갱신되는 것을 확인하였다.

- 0x80000000: 사용자 파일 및 폴더 대상. 접근 시각 갱신 활성화
- 0x80000001: 사용자 파일 및 폴더 대상. 접근 시각 갱신 비활성화
- 0x80000002: 시스템 (사용자 포함) 파일 및 폴더 대상. 접근 시각 갱신 활성화
- 0x80000003: 시스템 (사용자 포함) 파일 및 폴더 대상. 접근 시각 갱신 비활성화

따라서, 파일/폴더 조사에 MAC 시간 값을 활용하듯이 (3) ADS의 Stream에도 MAC 시간 값을 활용할 수 있다. 이를 통해서, ADS의 생성, 수정, 접근(실행) 행위를 파악한다.

⁴ How to Enable or Disable NTFS Last Access Time Stamp Updates in Windows 10 , <https://www.tenforums.com/tutorials/139015-enable-disable-ntfs-last-access-time-stamp-updates-windows-10-a.html>

2.3.3. \$J 분석 (저널 분석)

\$J (저널)을 통해서, 파일/폴더에 ADS 가 생성된 흔적을 확인할 수 있다. NTFS Log Tracker 로 \$J 를 분석하면, “Named_Data_Stream” 키워드를 포함하는 이벤트가 ADS 생성과 관련된 흔적이다.

TimeStamp	FileName	FullPath	Event
2021-07-25 01:06:37	test.txt	\test\test.txt	File_Created / Named_Stream_Changed
2021-07-25 01:06:37	test.txt	\test\test.txt	File_Created / Named_Data_Stream_Added / Named_Stream_Changed
2021-07-25 01:06:37	test.txt	\test\test.txt	File_Created / Named_Data_Stream_Added / Named_Data_Stream_Overwritten / Named_Stream_Changed
2021-07-25 01:06:37	test.txt	\test\test.txt	File_Created / Named_Data_Stream_Added / Named_Data_Stream_Overwritten / Named_Stream_Truncated / Named_Stream_Changed
2021-07-25 01:06:37	test.txt	\test\test.txt	File_Created / Named_Data_Stream_Added / Named_Data_Stream_Overwritten / Named_Stream_Truncated / Named_Stream_Changed / File_Closed

[그림 5] \$J를 통해 확인한 ADS 생성 흔적

(4) 이벤트가 발생한 시간과 ADS의 MAC 시간 값을 이용하여, ADS의 생성/변경 시점을 파악할 수 있다.

2.4. 타임라인 분석

2.4.1. 타임라인

공격 수행 시각(2021-07-25 01:04:20) 이후, 계산기의 실행 흔적 과 ADS의 생성 및 실행 흔적을 종합하여 타임라인을 작성하면 다음과 같다.

- ADS 생성 흔적 → 노란색
- ADS 실행 (단순 실행) → 연두색
- ADS 실행 (계산기 실행) → 붉은색
- ADS 실행 여부 조사 필요 → 회색

시간 (UTC+9)	이벤트	출처	경로	비고
01:04:20	실행	Prefetch	\WINDOWS\SYSTEM32\CSCRIPT.EXE	
01:04:20	실행	Prefetch	\WINDOWS\SYSTEM32\RUNDLL32.EXE	
01:04:21	실행	BAM	Microsoft.WindowsCalculator_8wekyb3d8bbwe	
01:04:21	실행	Timeline	Microsoft.WindowsCalculator_8wekyb3d8bbwe!App	
01:04:21	실행	Prefetch	\PROGRAMFILES\WINDOWSAPPS\MICROSOFT.WINDOWSCALCULATOR_10.2011.16.0_X64_8W_EKYB3D8BBWE\CALCULATOR.EXE	
01:04:30	실행	Prefetch	\WINDOWS\SYSTEM32\CSCRIPT.EXE	
01:04:30	실행	Prefetch	\WINDOWS\SYSTEM32\REG.EXE	
01:04:31	실행	Prefetch	\WINDOWS\SYSTEM32\CALC.EXE	
01:04:46	ADS 추가	\$J	\test	[ADS 수정] TESTPS (공격과 무관)
01:04:46	실행	Prefetch	\WINDOWS\SYSTEM32\CERTUTIL.EXE	
01:05:18	ADS 추가	\$J	\Users\K\Desktop\test\fakefile.doc	[ADS 생성] payload.bat
01:05:18	생성	\$MFT	\Users\K\Desktop\testfakefile.doc:payload.bat	
01:05:18	수정	\$MFT	\Users\K\Desktop\testfakefile.doc:payload.bat	

01:05:18	실행	Prefetch	\WINDOWS\SYSTEM32\CMD.EXE	
01:05:22	실행	Prefetch	\WINDOWS\SYSTEM32\RUNDLL32.EXE	
01:06:29	접근	\$MFT	\Users\K\Desktop\testfakefile.doc:payload.bat	[ADS 실행] payload.bat
01:06:29	실행	Prefetch	\WINDOWS\SYSTEM32\CMD.EXE	
01:06:29	실행	BAM	\Windows\System32\cmd.exe	
01:06:37	ADS 추가	\$J	\test\test.txt	[ADS 생성] file.exe, calc.cab
01:06:37	생성	\$MFT	\test\test.txt:file.exe	
01:06:37	생성	\$MFT	\test\test.txt:calc.cab	
01:06:37	실행	Prefetch	\WINDOWS\SYSTEM32\ESENTUTL.EXE	
01:06:44	실행	Prefetch	\WINDOWS\SYSTEM32\EXPAND.EXE	
01:06:47	실행	Prefetch	\WINDOWS\SYSTEM32\EXTRAC32.EXE	
01:06:47	실행	BAM	\Windows\System32\extrac32.exe	
01:06:51	ADS 추가	\$J	\test\testc.txt	[ADS 생성] calc.dll
01:06:51	생성	\$MFT	\test\testc.txt:calc.dll	
01:06:51	실행	Prefetch	\WINDOWS\SYSTEM32\FINDSTR.EXE	
01:06:58	ADS 추가	\$J	\test\testc.txt	[ADS 수정] calc.dll
01:06:58	실행	Prefetch	\WINDOWS\SYSTEM32\FINDSTR.EXE	
01:07:14	ADS 추가	\$J	\test\testc.txt	[ADS 수정] calc.dll
01:07:14	수정	\$MFT	\test\testc.txt:calc.dll	
01:07:14	실행	Prefetch	\WINDOWS\SYSTEM32\FINDSTR.EXE	
01:08:28	접근	\$MFT	\test\testc.txt:calc.dll	[ADS 실행] calc.dll #계산기 실행
01:08:28	실행	Prefetch	\WINDOWS\SYWOW64\RUNDLL32.EXE	
01:08:28	실행	Prefetch	\WINDOWS\SYSTEM32\CONTROL.EXE	
01:08:28	실행	Prefetch	\WINDOWS\SYSTEM32\RUNDLL32.EXE	
01:08:38	실행	Prefetch	\WINDOWS\SYSTEM32\CALC.EXE	
01:08:38	실행	Prefetch	\WINDOWS\SYSTEM32\FORFILES.EXE	
01:08:41	실행	Prefetch	\WINDOWS\SYSTEM32\MAKECAB.EXE	
01:08:51	ADS 추가	\$J	\test\test.txt	[ADS 수정] file.exe, calc.cab
01:08:51	수정	\$MFT	\test\test.txt:file.exe	
01:08:51	수정	\$MFT	\test\test.txt:calc.cab	
01:08:51	접근	\$MFT	\test\test.txt:file.exe	
01:08:51	접근	\$MFT	\test\test.txt:calc.cab	
01:08:51	실행	Prefetch	\WINDOWS\SYSTEM32\MAKECAB.EXE	
01:09:38	실행	Prefetch	\WINDOWS\SYSTEM32\MAVINJECT.EXE	
01:09:46	실행	Prefetch	\WINDOWS\SYSTEM32\MSHTA.EXE	
01:09:46	실행	Prefetch	\WINDOWS\SYSTEM32\CALC.EXE	

01:09:46	실행	BAM	\Windows\System32\mshta.exe	
01:09:46	실행	Timeline	\WINDOWS\SYSTEM32\mshta.exe	
01:09:50	ADS 추가	\$J	\test\filer.txt	[ADS 생성] run.reg
01:09:50	생성	\$MFT	\test\filer.txt:run.reg	
01:09:50	수정	\$MFT	\test\filer.txt:run.reg	
01:09:50	접근	\$MFT	\test\filer.txt:run.reg	
01:09:50	실행	Prefetch	\WINDOWS\SYSTEM32\REG.EXE	
01:09:55	실행	Prefetch	\WINDOWS\SYSTEM32\SC.EXE	[조사 필요] SC.EXE, WMIC.EXE
01:09:56	실행	Prefetch	\WINDOWS\SYSTEM32\SC.EXE	
01:10:21	실행	Prefetch	\WINDOWS\SYSTEM32\SC.EXE	
01:10:21	실행	Prefetch	\WINDOWS\SYSTEM32\SC.EXE	
01:10:30	실행	Prefetch	\WINDOWS\SYSTEM32\WBEM\WMIC.EXE	
01:10:30	실행	Prefetch	\WINDOWS\SYSTEM32\CALC.EXE	[ADS 생성] calc.vbs
01:10:35	ADS 추가	\$J	\test\file.txt	
01:10:35	생성	\$MFT	\test\file.txt:calc.vbs	
01:10:35	실행	Prefetch	\WINDOWS\SYSTEM32\FINDSTR.EXE	[ADS 수정] calc.vbs
01:10:41	ADS 추가	\$J	\test\file.txt	
01:10:41	실행	Prefetch	\WINDOWS\SYSTEM32\FINDSTR.EXE	[ADS 수정] valc.vbs
01:10:48	ADS 추가	\$J	\test\file.txt	
01:10:48	실행	Prefetch	\WINDOWS\SYSTEM32\FINDSTR.EXE	
01:11:01	실행	Timeline	\Users\K\Desktop\test\exe\calc.exe	
01:11:14	ADS 추가	\$J	\test\file.txt	[ADS 수정] calc.vbs
01:11:14	수정	\$MFT	\test\file.txt:calc.vbs	
01:11:14	실행	Prefetch	\WINDOWS\SYSTEM32\FINDSTR.EXE	
01:11:57	실행	Prefetch	\WINDOWS\SYSTEM32\WSCRIPT.EXE	[ADS 실행] calc.vbs #계산기 실행
01:11:57	실행	Prefetch	\WINDOWS\SYSTEM32\CALC.EXE	
01:11:58	실행	Timeline	Microsoft.WindowsCalculator_8wekyb3d8bbwe!App	
01:12:00	실행	Timeline	Microsoft.WindowsCalculator_8wekyb3d8bbwe!App	
01:12:06	접근	\$MFT	\test\file.txt:calc.vbs	[ADS 실행] calc.vbs #계산기 실행
01:12:06	실행	Prefetch	\WINDOWS\SYSTEM32\CALC.EXE	
01:12:06	실행	Prefetch	\WINDOWS\SYSTEM32\CSCRIPT.EXE	
01:12:06	실행	Timeline	Microsoft.WindowsCalculator_8wekyb3d8bbwe!App	
01:12:18	실행	Prefetch	\WINDOWS\SYSTEM32\CALC.EXE	
01:12:18	실행	Prefetch	\WINDOWS\SYSTEM32\CALC.EXE	
01:12:19	실행	Timeline	Microsoft.WindowsCalculator_8wekyb3d8bbwe!App	

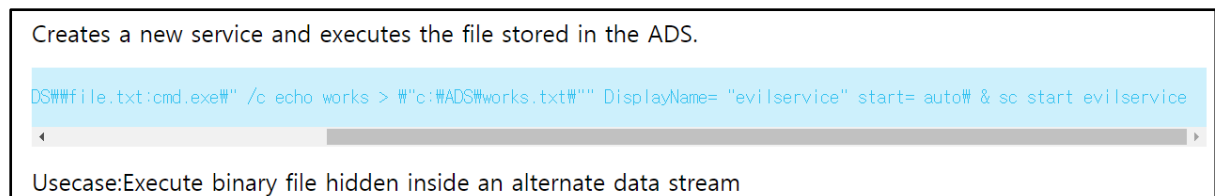
01:13:00	실행	Prefetch	\PROGRAMDATA\MICROSOFT\WINDOWS DEFENDER\PLATFORM\4.18.2106.6-0\MPCMDRUN.EXE	
01:13:00	실행	Prefetch	\PROGRAMDATA\MICROSOFT\WINDOWS DEFENDER\PLATFORM\4.18.2106.6-0\MPCMDRUN.EXE	
01:13:00	실행	Prefetch	\PROGRAMDATA\MICROSOFT\WINDOWS DEFENDER\PLATFORM\4.18.2106.6-0\MPCMDRUN.EXE	
01:13:00	실행	Prefetch	\PROGRAMDATA\MICROSOFT\WINDOWS DEFENDER\PLATFORM\4.18.2106.6-0\MPCMDRUN.EXE	
01:13:01	실행	Timeline	Microsoft.WindowsCalculator_8wekyb3d8bbwe!App	
01:14:01	실행	Timeline	Microsoft.WindowsCalculator_8wekyb3d8bbwe!App	
01:18:01	실행	BAM	\Users\K\Desktop\test\exe\calc.exe	
01:19:08	실행	Prefetch	\WINDOWS\SYSTEM32\CMD.EXE	

[표 5] ADS 생성, 수정, 실행 타임라인 (2021. 07. 25)

.reg 파일을 실행하는 regedit.exe는 타임라인 상 실행되지 않았다. 따라서, \test\filter.txt:run.reg는 생성은 되었지만 실행되지 않았으므로, Registry에서 계산기를 실행할 수 있는 흔적을 확인할 수 없었다.

2.4.2. SC.EXE, WMIC.EXE 추가 조사

Windows 서비스를 등록할 때 sc.exe가 실행된다. sc.exe는 ADS에 은닉된 실행 파일을 서비스로 등록하여 실행할 수 있다⁵.



[그림 6] sc.exe에 대한 LOLBAS 설명

sc.exe와 wmic.exe는 Prefetch의 참조 파일 목록에서 ADS와 관련된 파일을 확인할 수 없었다. 또한, Registry에서 등록된 서비스 목록을 살펴봐도 ADS와 관련된 흔적을 확인할 수 없었다.

하지만, \$J에서 작업 sc.exe 실행 이후 작업 스케줄러(schtask.exe)를 이용해 “Calc”라는 작업을 등록한 것을 확인하였다.

⁵ Sc.exe, <https://lolbas-project.github.io/lolbas/Binaries/Sc/>

TimeStamp	FileName	FullPath	
2021-07-25 01:00:00	필터	필터	file_created
2021-07-25 01:09:56	SC.EXE-443D0E78.pf	Windows\Prefetch\SC.EXE-443D0E78.pf	File_Created
2021-07-25 01:09:56	SC.EXE-443D0E78.pf	Windows\Prefetch\SC.EXE-443D0E78.pf	File_Created / Data_Added
2021-07-25 01:09:56	SC.EXE-443D0E78.pf	Windows\Prefetch\SC.EXE-443D0E78.pf	File_Created / Data_Added / File_Closed
2021-07-25 01:10:26	Calc	Windows\System32\Tasks\Calc	File_Created
2021-07-25 01:10:26	Calc	Windows\System32\Tasks\Calc	File_Created / Data_Added
2021-07-25 01:10:26	Calc	Windows\System32\Tasks\Calc	File_Created / Data_Added / File_Closed
2021-07-25 01:10:26	SCHTASKS.EXE-BA1E321E.pf	Windows\Prefetch\SCHTASKS.EXE-BA1E321E.pf	File_Created
2021-07-25 01:10:26	SCHTASKS.EXE-BA1E321E.pf	Windows\Prefetch\SCHTASKS.EXE-BA1E321E.pf	File_Created / Data_Added
2021-07-25 01:10:26	SCHTASKS.EXE-BA1E321E.pf	Windows\Prefetch\SCHTASKS.EXE-BA1E321E.pf	File_Created / Data_Added / File_Closed
2021-07-25 01:10:30	WMIC.EXE-887410DD.pf	Windows\Prefetch\WMIC.EXE-887410DD.pf	File_Created

[그림 7] sc.exe 실행 이후 생성된 파일

상기 타임라인의 Prefetch에 기록된 계산기는 시스템 폴더에 존재하는 계산기다. 작업 스케줄에 등록한 Calc는 시스템 폴더에 존재하는 계산기가 아니라, 사용자가 지정한 폴더에 저장된 계산기를 실행한다.

39	<Actions Context="Author">
40	<Exec>
41	<Command>c:\users\k\Desktop\test\exe\calc.exe</Command>
42	</Exec>

[그림 8] 작업 스케줄에 등록된 Calc 확인

따라서, sc.exe와 wmic.exe는 ADS를 실행했다고 보기 어렵고, 분석 목적에 부합하는 계산기도 실행했다고 보기 어렵다.

2.5. 결론

본 시나리오에서 공격 수행 이후 생성된 ADS 중에서, 실행된 ADS와 해당 ADS의 계산기 실행 여부를 정리하면 다음과 같다.

ADS 경로	실행	계산기 실행	실행한 프로세스	실행 시각 (UTC+9)
C:\test\filer.txt:run.reg	X	X	없음	-
C:\test\test.txt:calc.cab	X	X	없음	-
C:\test\test.txt:file.exe	X	X	없음	-
C:\Users\K\Desktop\test\fakefile.doc:payload.bat	O	X	cmd.exe	2021-07-25 01:05:08
C:\test\testc.txt:calc.dll	O	O	rundll32.exe	2021-07-25 01:08:28
C:\test\file.txt:calc.vbs	O	O	wscript.exe	2021-07-25 01:11:57
			cscript.exe	2021-07-25 01:12:06

[표 6] 실행된 ADS 목록 및 계산기 실행 여부