CrowdStrike 보안 업데이트 사칭
와이퍼 악성코드 분석

요약

1. CrowdStrike의 소프트웨어 장애 사건을 이용하여
   보안 소프트웨어 업데이트 파일로 위장한 와이퍼 악성코드가 유포됨.
   피해자가 직접 클릭하여 와이퍼 악성코드를 실행하도록 유도
2. 와이퍼 악성코드는 1차 Dropper인 보안 패치 위장 악성코드를 통해
   2차 인젝터인 Champion.pif를 거쳐 RegAsm.exe 내 삽입되어
   데이터 삭제, 시스템 파괴, 복구 방해를 수행함.
3. 목표 지향적인 와이퍼 악성코드는 이메일 첨부 파일, 악성 웹사이트,
   제로데이 취약점을 통해 배포되어 데이터나 시스템에
   영구적인 손상을 안겨 큰 피해를 발생시킬 수 있음.

대응 방안

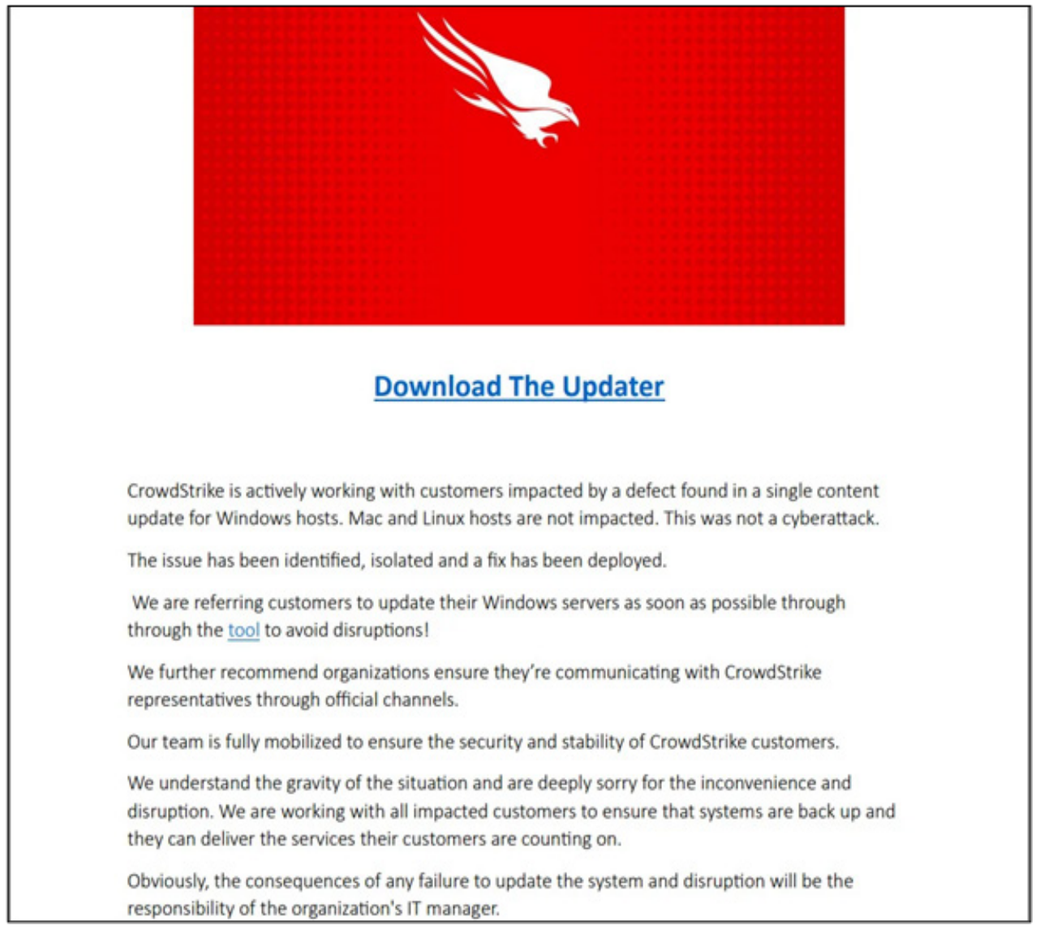
1. Privacy-i EDR과 같은 EDR 제품을 통해 취약점 실행을 행위 기반으로 차단
2. 주요 데이터는 주기적인 백업을 통해 시스템 파괴 시에도
   복구가 가능하도록 대비
3. 논리적 망분리를 적용하여 악성코드 PC 유입을 원천 차단
4. AV(패턴기반탐지) + EDR(행위기반탐지) 솔루션
5. PC 취약점을 주기적으로 점검, 보완
6. 신뢰할 수 없는 메일의 첨부파일 실행 금지
7. 비업무 사이트 및 신뢰할 수 없는 웹사이트의 연결 차단
8. OS나 어플리케이션은 최신 형상 유지

목차

1. 개요
1.1 배경
2. 정보
2.1 파일 정보
2.2 해외 유명 보안업체 C社 사칭 공격 흐름
3. 분석
3.1 C社 업데이트 위장 Dropper 분석
3.1.1 DLL 로드 확인
3.1.2 Temp 폴더 존재 확인
3.1.3 Temp 폴더 경로 확인
3.1.4 Temp 폴더 내 파일 생성 가능 여부 확인
3.1.5 Dropper 내 악성 리소스 획득
3.1.6 사용자 사용 언어 확인
3.1.7 Temp 폴더 작업 디렉토리 설정
3.1.8 페이로드 및 주요 악성 리소스 분할 드롭
3.1.9 윈도우 유틸리티를 통한 페이로드 재조립 및 악성 행위 수행
3.1.10 타임아웃 및 Dropper 종료
3.2 Champion.pif 인젝터 분석
3.2.1 권한 확인 및 관리자 권한으로 실행
3.2.2 안티 디버깅
3.2.3 와이퍼 코드 로드
3.2.4 윈도우 유틸리티 생성
3.2.5 프로세스 권한 획득
3.2.6 RegAsm 실행 및 와이퍼 코드 삽입
3.3 RegAsm.exe 내 와이퍼 코드 분석
3.3.1 RegAsm.exe의 경로 획득
3.3.2 RegAsm.exe의 파일 타입 확인
3.3.3 RegAsm.exe의 파일 파괴
3.3.4 시스템 장애 / 파괴
4. Privacy-i EDR 탐지 정보
4.1 이상 파일 I/O 행위 탐지를 통한 와이퍼 탐지 및 차단
5. 대응

1. 개요

1.1 배경



Download The Updater

CrowdStrike is actively working with customers impacted by a defect found in a single content update for Windows hosts. Mac and Linux hosts are not impacted. This was not a cyberattack.

The issue has been identified, isolated and a fix has been deployed.

We are referring customers to update their Windows servers as soon as possible through the tool to avoid disruptions!

We further recommend organizations ensure they're communicating with CrowdStrike representatives through official channels.

Our team is fully mobilized to ensure the security and stability of CrowdStrike customers.

We understand the gravity of the situation and are deeply sorry for the inconvenience and disruption. We are working with all impacted customers to ensure that systems are back up and they can deliver the services their customers are counting on.

Obviously, the consequences of any failure to update the system and disruption will be the responsibility of the organization's IT manager.

[그림 1] 해외 유명 보안업체 크라우드 스트라이크 사칭 이메일

소만사 악성코드 분석 센터는 최근 CrowdStrike (크라우드 스트라이크)의
보안 소프트웨어 업데이트 파일로 위장하여 전 세계 시스템에 장애를 발생시켜
큰 피해를 유발한 와이퍼 악성코드가 유포되고 있는 것을 확인하고,
신속히 악성코드 샘플을 확보해 분석하였다.
또한 본 악성코드 분석 보고서 발간을 통해 와이퍼 악성코드의 특징과 대응 방안에 대해 서술하였다.

공격자는 피해자에게 크라우드 스트라이크를 사칭한 소프트웨어 업데이트 이메일을 전송하여
사용자가 이메일 내 첨부 파일을 다운로드하도록 유도하였다.
해당 이메일 내부에는 실제 크라우드 스트라이크 아이콘이 등록된 악성코드가 존재하며,
이를 통해 사용자는 큰 의심 없이 악성코드를 실행하도록 위장하였다.

공격자는 피해자를 속여 윈도우 호스트의 장애 및 보안 업데이트를 수행할 수 있도록
관련 내용 항목을 첨부하였고,
피해자가 메일 내 링크를 클릭할 시 Update라는 압축 파일이 다운로드 되며,
압축 파일 내에는 보안 패치로 위장된 와이퍼 악성코드가 존재했다.
악성코드가 실행되면 피해자의 시스템을 파괴하고, 공격자에게 실행 결과를 전송한다.

2. 정보

2.1 파일 정보

Name	update.zip
Type	Compress ZIP File
Behavior	Malicious Compress ZIP File
SHA-256	96dec6e07229201a02f538310815c695cf6147c548ff1c6a0def2fe38f3dcbc8

[표 1] 해외 유명 보안 업체 크라우드 스트라이크의 업데이트 파일로 위장한 압축 파일

Name	CrowdStrike.exe
Type	PE File
Behavior	Malware Dropper
SHA-256	4491901eff338ab52c85a77a3fbd3ce80fda738046ee3b7da7be468da5b331a3

[표 2] 악성코드를 드롭하는 업데이트 위장 실행 파일

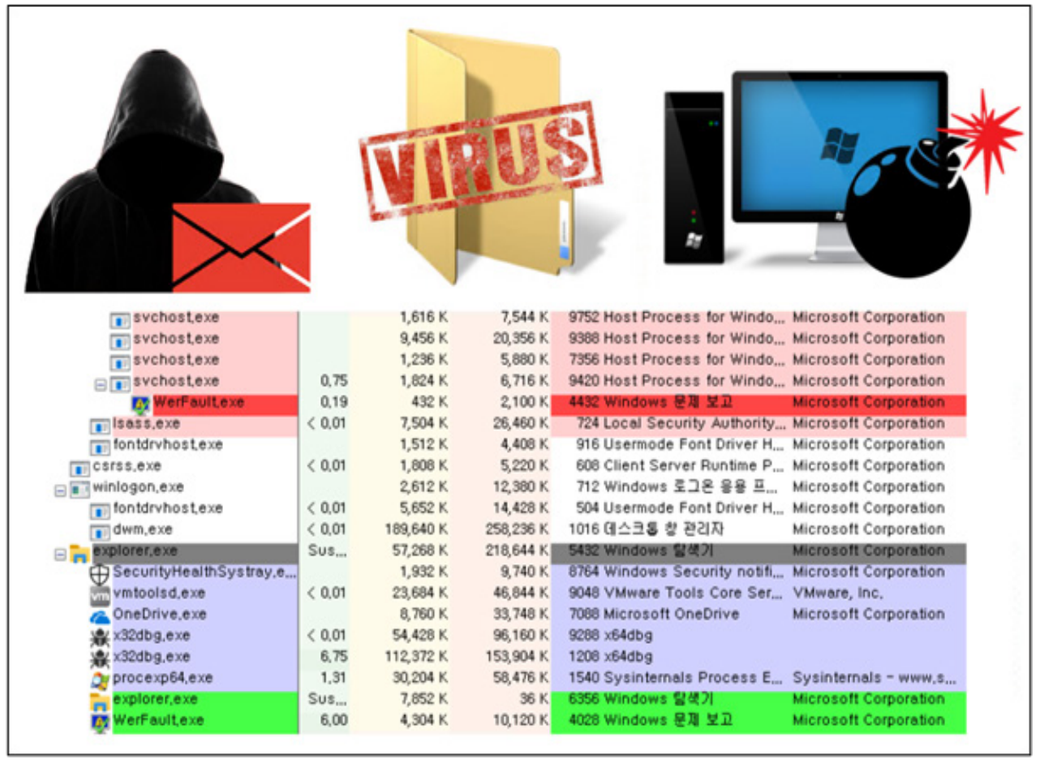
Name	Champion.pif
Type	Autoit V3 Script
Behavior	Malware Injector
SHA-256	865347471135bb5459ad0e647e75a14ad91424b6f13a5c05d9ecd9183a8a1cf4

[표 3] 드롭되어 최종 페이로드를 인젝션하는 인젝터

Name	RegAsm.exe
Type	Microsoft Utility
Behavior	Infected Wiper
SHA-256	c066aee7aa3aa83f763ebc5541daa266ed6c648fbffcde0d836a13b221bb2adc

[표 4] 와이퍼 악성코드가 인젝션되는 정상 윈도우 유틸리티

2.2 크라우드 스트라이크 사칭 공격 흐름



[그림 2] 국내 유명 엔터테인먼트 社 사칭 공격 흐름

[그림 3] 크라우드 스트라이크의 업데이트 파일로 위장한 와이퍼 악성코드

① 크라우드 스트라이크 사칭 악성 이메일 제작

공격자는 피해자에게 보낼 와이퍼가 첨부된 악의적인 이메일을 제작한다.

② 피해자에게 악성 이메일 송신

해커는 악의적인 이메일 내 보안 패치로 위장한 압축 파일을 삽입하고 피해자에게 송신한다.

③ 압축 파일 내 보안 패치로 위장한 와이퍼 악성코드

압축을 해제하고 보안 패치를 실행한 사용자의 PC에서 와이퍼 악성코드가 실행된다.

④ 와이퍼 악성코드에 의한 피해자 시스템 파괴

와이퍼 악성코드가 실행되어 사용자의 시스템 내 모든 정보가 파괴된다.

⑤ 해커에서 작업 결과 전송

파괴 작업 수행 결과를 해커에게 텔레그램을 통해 전송한다.

3. 분석

3.1 와이퍼 악성코드 Dropper 분석

3.1.1 DLL 로드 확인

```
00403883 <cs      81EC D4020000      sub esp,2D4                                  EntryPoint
00403889          53                 push ebx
0040388A          55                 push ebp
0040388B          56                 push esi
0040388C          57                 push edi                                     edi:EntryPoint
0040388D          6A 20              push 20
0040388F          33ED               xor ebp,ebp
00403891          5E                 pop esi
00403892          896C24 18          mov dword ptr ss:[esp+18],ebp
00403896          C74424 10 68924000 mov dword ptr ss:[esp+10],cs.409268           [esp+10]:L"Erro
0040389E          896C24 14          mov dword ptr ss:[esp+14],ebp
004038A2          FF15 30804000      call dword ptr ds:[<&InitCommonControls>]
```

[그림 4] DLL 로드 확인

COMCTL32.DLL이 정상적으로 로드되어 있는지 확인한다.
이는 공통 컨트롤을 사용하기 위함이며,
본 악성코드에서 추후 사용자의 악성코드 실행 유도를 하기 위해 툴팁을 생성하고
Temp 폴더가 없을 시 이를 생성하도록 유도하기 위한 툴팁에도 사용된다.

3.1.2. Temp 폴더 존재 확인

```
004062FC    56          push esi
004062FD    8B7424 08   mov esi,dword ptr ss:[esp+8]      [esp+8]:EntryPoint
00406301    57          push edi                         edi:"SHFOLDER"
00406302    8B3CF5 60B04000 mov edi,dword ptr ds:[esi*8+40B060]  edi:"SHFOLDER", [esi*8+40B060]:"SHF
00406309    57          push edi                         edi:"SHFOLDER"
0040630A    FF15 F4804000 call dword ptr ds:[<&GetModuleHandleA>]
00406310    85C0        test eax,eax
00406312    75 0B       jne cs.40631F
00406314    57          push edi                         edi:"SHFOLDER"
00406315    FF15 F0804000 call dword ptr ds:[<&LoadLibraryA>]
0040631B    85C0        test eax,eax
0040631D    74 0E       je cs.40632D
0040631F    FF34F5 64B04000 push dword ptr ds:[esi*8+40B064]  [esi*8+40B064]:"SHGetFolderPathW"
00406326    50          push eax
00406327    FF15 EC804000 call dword ptr ds:[<&GetProcAddress>]
0040632D    5F          pop edi                          edi:"SHFOLDER"
0040632E    5E          pop esi
0040632F    C2 0400     ret 4

004038D6    55          push ebp
004038D7    68 64924000 push cs.409264
004038DC    FF15 84814000 call dword ptr ds:[<&SHGetFileInfoW>]
004038E2    68 4C924000 push cs.40924C                   40924C:L"NSIS
004038E7    68 C0AD4600 push cs.46ADC0
004038EC    E8 18270000 call cs.406009
004038F1    FF15 B0804000 call dword ptr ds:[<&GetCommandLineW>]
004038F7    50          push eax                         eax:L"퓡-"
004038F8    BF A0304C00 mov edi,cs.4C30A0                edi:EntryPoint
004038FD    57          push edi                         edi:EntryPoint
004038FE    E8 06270000 call cs.406009
```

[그림 5] Temp 폴더 존재 확인

향후 Dropper는 Temp 폴더에 악성코드를 드롭하고 이후 해당 악성코드를 실행시키는 흐름을 갖는다.
본 행위를 위해 동적으로 API를 로드하여 Temp 폴더의 존재를 확인하는 작업을 수행한다.

3.1.3 Temp 폴더 경로 확인

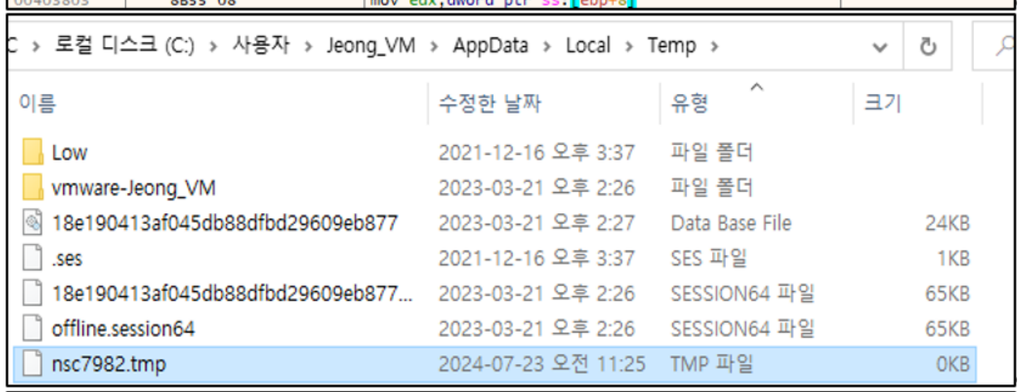
```
004039F5    BB C8704D00 mov ebx,cs.4D70C8
004039FA    53          push ebx
004039FB    68 04200000 push 2004
00403A00    FF15 AC804000 call dword ptr ds:[<&GetTempPathW>]
00403A06    E8 C1FDFFFF call cs.4037CC
00403A0B    85C0        test eax,eax
00403A0D    75 24       jne cs.403A33
00403A0F    68 FF1F0000 push 1FFF
00403A14    53          push ebx
```

[그림 6] Temp 폴더 경로 확인

Temp 폴더의 존재를 확인한 후 GetTempPathW API를 사용하여 Temp 폴더의 경로를 확인한다.

3.1.4 Temp 폴더 내 파일 생성 가능 여부 확인

```
004037EC    56          push esi                         esi:L"C:\\Users\\J
004037ED    FF15 84804000 call dword ptr ds:[<&CreateDirectoryW>]
004037F3    56          push esi                         esi:L"C:\\Users\\J
004037F4    68 C0304D00 push cs.4D30C0
004037F9    E8 81260000 call cs.405E7F
004037FE    5E          pop esi                          esi:L"C:\\Users\\J
004037FF    C3          ret
00403800    55          push ebp
00403801    8BEC        mov ebp,esp
00403803    8B55 08     mov edx,dword ptr ss:[ebp+8]
```




```
00403A21    E8 FF250000 call <JMP.&lstrcatW>
00403A26    E8 A1FDFFFF call cs.4037CC
00403A2B    85C0        test eax,eax                     eax:L"C:\\Users\\
00403A2D    0F84 99000000 je cs.403ACC
00403A33    68 C0304D00 push cs.4D30C0                   4D30C0:L"C:\\User
00403A38    FF15 70814000 call dword ptr ds:[<&DeleteFileW>]
```

[그림 7] Temp 폴더 내 파일 생성 가능 여부 확인

Temp 폴더가 존재하는지 최종적으로 확인하기 위해 CreateDirectoryW API를 호출하여
Temp 폴더가 이미 존재하여 생성하지 않아도 되는지 확인한다.
이와 함께 폴더 내 파일 생성 가능 여부를 확인하기 위해
임의의 임시 파일을 만든 후 성공 시 이를 제거한다.

3.1.5 Dropper 내 악성 리소스 획득

```
00403337    8BEC          mov ebp,esp
00403339    56            push esi
0040333A    8B75 0C       mov esi,dword ptr ss:[ebp+C]
0040333D    6A 00         push 0
0040333F    8D45 0C       lea eax,dword ptr ss:[ebp+C]
00403342    50            push eax
00403343    56            push esi
00403344    FF75 08       push dword ptr ss:[ebp+8]
00403347    FF35 10B04000 push dword ptr ds:[40B010]
0040334D    FF15 58814000 call dword ptr ds:[<&ReadFile>]
00403353    85C0          test eax,eax
00403355    74 0A         je cs.403361
00403357    3975 0C       cmp dword ptr ss:[ebp+C],esi
0040335A    75 05         jne cs.403361
0040335C    33C0          xor eax,eax
0040335E    40            inc eax
0040335F    EB 02         jmp cs.403363
00403361    33C0          xor eax,eax
00403363    5E            pop esi
00403364    5D            pop ebp
00403365    C2 0800       ret 8
00403368    6A 00         push 0
0040336A    6A 00         push 0
0040336C    FF7424 0C     push dword ptr ss:[esp+C]
00403370    FF35 10B04000 push dword ptr ds:[40B010]
00403376    FF15 60814000 call dword ptr ds:[<&SetFilePointer>]
0040337C    C2 0400       ret 4
```

[그림 8] Dropper 내 악성 리소스 획득

자기 자신의 바이너리를 읽어 내부에 존재하는 악성 리소스를 획득한다.
악성 리소스는 이후, 이전에 구한 Temp 폴더 내 생성되어 추가적인 악성 행위를 수행하게 된다.

3.1.6 사용자 사용 언어 확인

```
00405942    3BC3          cmp eax,ebx
00405944    74 12         je cs.405958
00405946    FFD0          call eax                    GetUserDefaultUILanguage
00405948    0FB7C0        movzx eax,ax
0040594B    50            push eax
0040594C    68 C0304D00   push cs.4D30C0               4D30C0:L"C:\\Users\\Jeong_
00405951    E8 FB050000   call cs.405F51
00405956    EB 5B         jmp cs.4059B3
00405958    6A 30         push 30
```

[그림 9] 사용자 사용 언어 확인

GetUserDefaultUILanguage API를 호출하여 사용자가 사용하는 언어를 확인한다.
이는 추후 사용자의 악성코드 실행을 유도하기 위한 툴팁에
안내 메시지로 사용될 언어를 확인하는 작업이다.

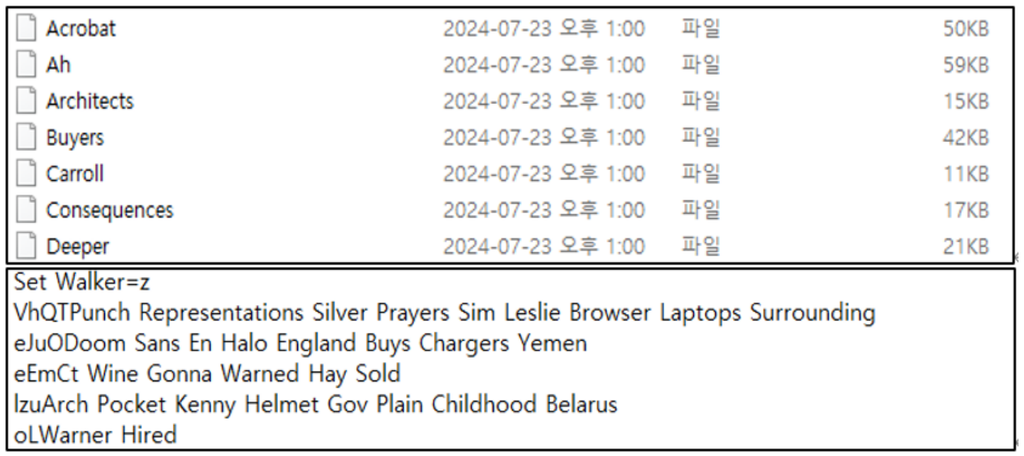
3.1.7 Temp 폴더 작업 디렉토리 설정

```
00401875    FF75 08       push dword ptr ss:[ebp+8]   [ebp+8]:L"C:\\Us
00401878    68 B0B04C00   push cs.4CB0B0               4CB0B0:L"C:\\Us
0040187D    E8 87470000   call cs.406009
00401882    FF75 08       push dword ptr ss:[ebp+8]   [ebp+8]:L"C:\\Us
00401885    FF15 78804000 call dword ptr ds:[<&SetCurrentDirectoryW>]
0040188B    E9 53180000   jmp cs.4030E3
00401890    6A F5         push FFFFFFF5
00401892    E9 F2FDFFFF   jmp cs.401689
00401897    53            push ebx
00401898    E8 BFFBFFFF   call cs.40145C
```

[그림 10] Temp 폴더의 작업 디렉토리 설정

Temp 폴더 내 악성 파일 드롭 및 악의적인 행위에 필요한 데이터를 생성하기 위해
현 작업 디렉토리를 Temp 폴더로 변경한다.

3.1.8 페이로드 및 주요 악성 리소스 분할 드롭



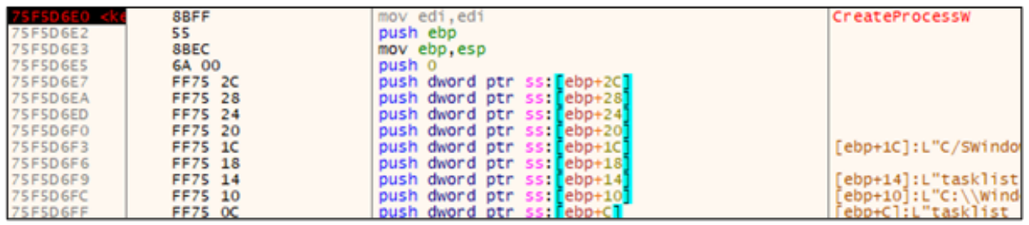
[그림 11] Temp 폴더 내 와이퍼 및 주요 악성 리소스 분할 드롭

Temp 폴더 내 와이퍼의 PE 파일 조각과 악성 리소스들을 분할하여 드롭한다.
이 분할된 파일들은 이후 다시 재조합 및 파싱되어 페이로드와 악성 리소스로 동작하게 된다.

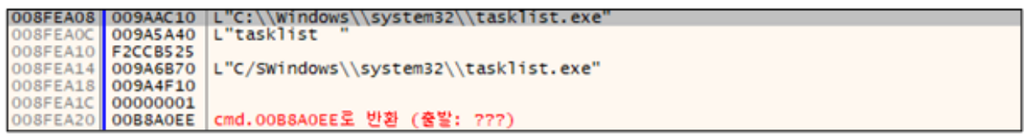
Acrobat, Ah, Architects, Buyers, Carroll, Consequences, Deeper, Democracy, Develops, Ferry, Fu, Gov, Guest, Halo, Handle, Honda, Hub, Job, Jul, Lasting, Moreover, Number, offline, Often, Reci-pes, Relative, Ripe, Sept, Str, Treating, Ukraine, Viagra, Vision, Wave

[표 5] 드롭되는 파일 이름

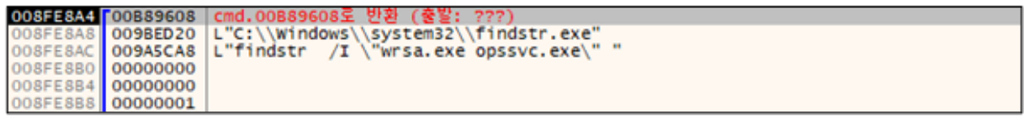
3.1.9 윈도우 유틸리티를 통한 페이로드 재조립 및 악성 행위 수행



CreateProcessW API를 반복적으로 호출하여 윈도우 프로세스를 실행하며 악성 행위를 수행하는데,
이는 아래와 같이 수행된다.



tasklist.exe tasklist	현재 실행중인 프로세스 목록 획득



findstr.exe findstr	설치된 엔드포인트 보안 프로그램 확인

```
008FE8A4 00B89608 cmd.00B89608로 반환 (출발: ???)
008FE8A8 009BED40 L"C:\\Windows\\system32\\findstr.exe"
008FE8AC 009A5A98 L"findstr  /I \"avastui.exe avgui.exe bdservicehost.exe nswscsvc.exe sophoshealth.exe\" "
008FE8B0 00000000
008FE8B4 00000000
008FE8B8 00000001
008FE8BC 00080000
```

findstr.exe findstr /I avastui.exe avgui.exe bdser-vicehost.exe nswcsvc.exe sophoshealth.exe	설치된 주요 안티바이러스 제품 목록 확인

```
008FE4DC 00B89608 cmd.00B89608로 반환 (출발: ???)
008FE4E0 009A5D40 L"C:\\Windows\\system32\\findstr.exe"
008FE4E4 009A4350 L"findstr  /V \"locatedflatrendsoperating\" Ukraine "
008FE4E8 00000000
008FE4EC 00000000
008FE4F0 00000001
008FE4F4 00080000
```

findstr.exe findstr /I locatedflatrendsoperating Ukraine	분할 생성한 Ukraine 파일 내 특정 문자열 검색

```
CBAE6810 L"C:\\Windows\\System32\\cmd.exe"
CBAE4D8A0 L"cmd  /c md 564784"
00000000
```

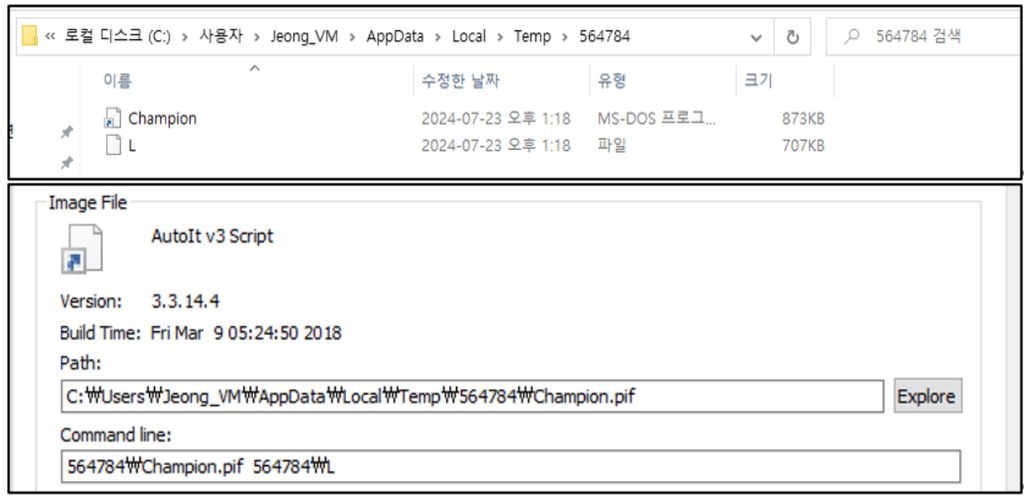
cmd.exe cmd /c md 564784	Temp 폴더 내 디렉토리 생성

```
008FE4DC 00B89608 cmd.00B89608로 반환 (출발: ???)
008FE4E0 009A5D40 L"C:\\Windows\\system32\\cmd.exe"
008FE4E4 009A4350 L"cmd  /c copy /b Treating + Viagra + Vision + Jul + Str 564784\\L "
008FE4E8 00000000
008FE4EC 00000000
008FE4F0 00000001
008FE4F4 00080000
```

cmd.exe cmd /c copy /b Treating + Viagra + Vision + Jul + Str 564784\L	폴더 내 이전에 분할 생성한 파일을 재조립하여 페이로드 생성


```
008FE2B8 008FE360
008FE2BC 00000000
008FE2C0 008FE89C
008FE2C4 00B95F10 cmd.00B95F10로 반환 (출발: ???)
008FE2C8 02E547B8 L"C:\\Users\\Jeong_VM\\AppData\\Local\\Temp\\564784\\Champion.pif"
008FE2CC 02E5C7E8 L"564784\\Champion.pif  564784\\L "
008FE2D0 00000000
008FE2D4 00000000
008FE2D8 00000001
008FE2DC 00080410
008FE2E0 0098F838 L"=::=::\\"
```

Champion.pif 564784\L	재조립한 페이로드와 악성 리소스 실행



[그림 12] 추가 페이로드 생성 및 실행

새로 생성된 Temp 폴더 내 564784 폴더에는
Champion.pif 라는 이름을 갖는 AutoIt v3 Script 파일이 생성된다.
이와 함께 L이라는 이름을 갖는 파일도 생성되는데,
이들은 각각 최종 와이퍼 코드를 대상 프로세스에 인젝션하는 인젝터와 와이퍼 소스코드이다.
이들은 생성과 동시에 실행된다.

3.1.10 타임아웃 및 Dropper 종료

```
75F5D6E0 <k  8BFF      mov edi,edi                 CreateProcessW
75F5D6E2     55        push ebp
75F5D6E3     8BEC      mov ebp,esp
75F5D6E5     6A 00     push 0
75F5D6E7     FF75 2C   push dword ptr ss:[ebp+2C]
75F5D6EA     FF75 28   push dword ptr ss:[ebp+28]
75F5D6ED     FF75 24   push dword ptr ss:[ebp+24]
75F5D6F0     FF75 20   push dword ptr ss:[ebp+20]
75F5D6F3     FF75 1C   push dword ptr ss:[ebp+1C]
75F5D6F6     FF75 18   push dword ptr ss:[ebp+18]
75F5D6F9     FF75 14   push dword ptr ss:[ebp+14]  {ebp+14]:L"timeout  15"
75F5D6FC     FF75 10   push dword ptr ss:[ebp+10]  {ebp+10]:L"C:\\Windows\\system32\\timeout.exe"
75F5D6FF     FF75 0C   push dword ptr ss:[ebp+C]   {ebp+C]:L"timeout  15"
```

[그림 13] 타임아웃

1차 Dropper인 보안 패치 위장 악성코드는 마지막으로 타임아웃을 15초로 설정하고 종료한다.
이를 통해 최종적으로 삽입될 와이퍼 악성코드와 인젝터, 와이퍼 코드가 피해자 PC에 생성된다.

3.2 Champion.pif 인젝터 분석

3.2.1 권한 확인 및 관리자 권한으로 실행

```
002B0BDD     50        push eax
002B0BDE     FF75 E4   push dword ptr ss:[ebp-1C]
002B0BE1     FF75 D4   push dword ptr ss:[ebp-2C]
002B0BE4     68 20A43100 push champion.31A420      31A420:L"runas"
002B0BE9     FF15 34072F00 call dword ptr ds:[<&GetForegroundWindow>]
002B0BEF     50        push eax
002B0BF0     FF15 C0042F00 call dword ptr ds:[<&ShellExecuteW>]
002B0BF6     8D4D E4   lea ecx,dword ptr ss:[ebp-1C]
002B0BF9     E8 B810FCFF call champion.271CB6
002B0BFE   ^ E9 5547FCFF jmp champion.275358
002B0C03     83C0 02   add eax,2
002B0C06     C646 24 01 mov byte ptr ds:[esi+24],1
002B0C0A   ^ E9 4448FCFF jmp champion.275453
```

[그림 14] 관리자 권한 확인 및 실행

2차 인젝터인 Champion.pif는 자신의 권한을 확인하고
관리자 권한이 아닐 경우 관리자 권한으로 자기 자신을 실행한다.

3.2.2 안티 디버깅

```
00EE526C     FF15 3003F600 call dword ptr ds:[<&GetCurrentDirectoryW>]
00EE5272     8D45 F9   lea eax,dword ptr ss:[ebp-7]
00EE5275     50        push eax
00EE5276     FF75 08   push dword ptr ss:[ebp+8]   [ebp+8]
00EE5279     E8 4AFCFFFF call champion.EE4EC8
00EE527E     FF15 2C03F600 call dword ptr ds:[<&IsDebuggerPresent>]
00EE5284     85C0      test eax,eax
00EE5286   v 0F85 95B80300 jne champion.F20B21
00EE528C     33FF      xor edi,edi
00EE528E     BE 9082F900 mov esi,champion.F98290
00EE5293     47        inc edi
00EE5294     393D E072F900 cmp dword ptr ds:[F972E0],edi
00EE529A   v 0F84 99B80300 je champion.F20B39
```

[그림 15] 안티 디버깅

IsDebuggerPresent API를 호출하여 현재 디버깅되고 있는지 확인한 후 안티 디버깅을 수행한다.

3.2.3 와이퍼 코드 로드

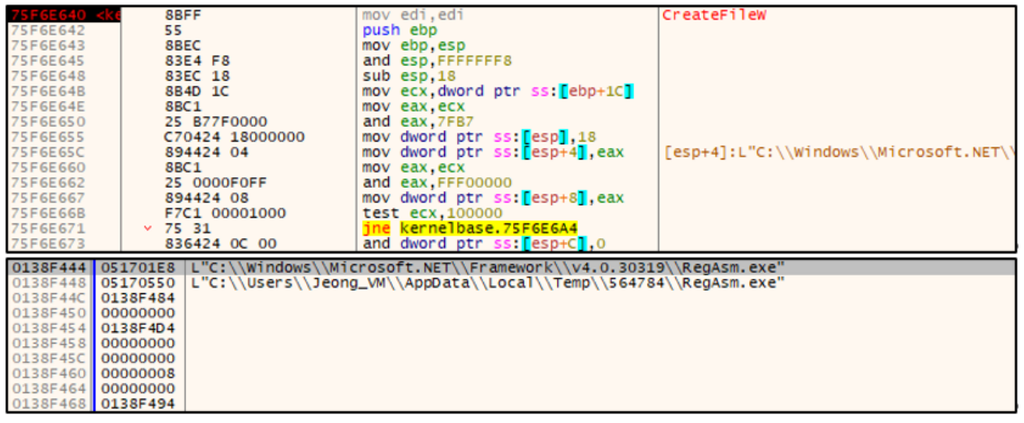
```
75F6E640 <k  8BFF      mov edi,edi                 CreateFileW
75F6E642     55        push ebp
75F6E643     8BEC      mov ebp,esp
75F6E645     83E4 F8   and esp,FFFFFFF8
75F6E648     83EC 18   sub esp,18
75F6E64B     8B4D 1C   mov ecx,dword ptr ss:[ebp+1C]
75F6E64E     8BC1      mov eax,ecx
75F6E650     25 B77F0000 and eax,7FB7
75F6E655     C70424 18000000 mov dword ptr ss:[esp],18
75F6E65C     894424 04 mov dword ptr ss:[esp+4],eax [esp+4]:L"C:\\Users\\Jeong_VM\\Ap
75F6E660     8BC1      mov eax,ecx
75F6E662     25 0000F0FF and eax,FFF00000
75F6E667     894424 08 mov dword ptr ss:[esp+8],eax
75F6E66B     F7C1 00001000 test ecx,100000
75F6E671   v 75 31     jne kernelbase.75F6E6A4

1: [esp+4] 0160C7D8 L"C:\\Users\\Jeong_VM\\AppData\\Local\\Temp\\564784\\L"
2: [esp+8] 80000000
3: [esp+C] 00000003
4: [esp+10] 0137F738
5: [esp+14] 00000003
```

[그림 16] 와이퍼 코드 로드

이전의 Dropper가 생성한 L이라는 이름의 와이퍼 코드 파일을 메모리로 읽어 이를 로드한다.
해당 코드 파일은 복잡하게 난독화 및 암호화되어 있어
이를 일련의 과정을 통해 복호화하고 재가공하여 추후 윈도우 유틸리티에 인젝션한다.

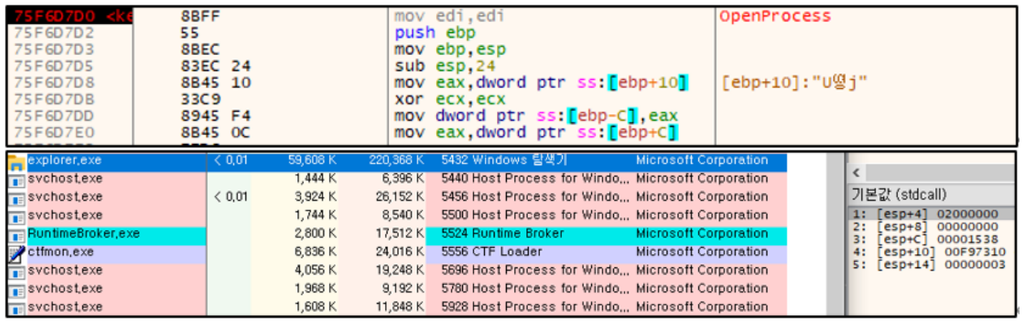
3.2.4 윈도우 유틸리티 생성



[그림 17] 윈도우 유틸리티 생성

인젝션을 수행해 와이퍼로 동작할 윈도우 유틸리티를 현재 폴더에 생성한다.
해당 윈도우 유틸리티는 RegAsm으로 어셈블리 등록 도구라고도 불리며,
악성 파일이 아닌 정상적인 윈도우 유틸리티이다.

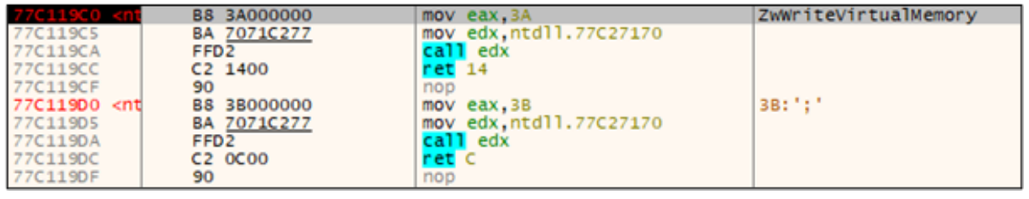
3.2.5 프로세스 권한 획득



[그림 18] 탐색기 핸들을 열어 권한 획득

RegAsm을 실행하고 인젝션을 수행하기에 앞서, 탐색기의 핸들을 열어 프로세스의 권한을 획득한다.
이를 통해 향후 와이퍼로서 동작할 RegAsm 프로세스에 필요한 권한을 주기 위함이다.

3.2.6 RegAsm 실행 및 와이퍼 코드 삽입

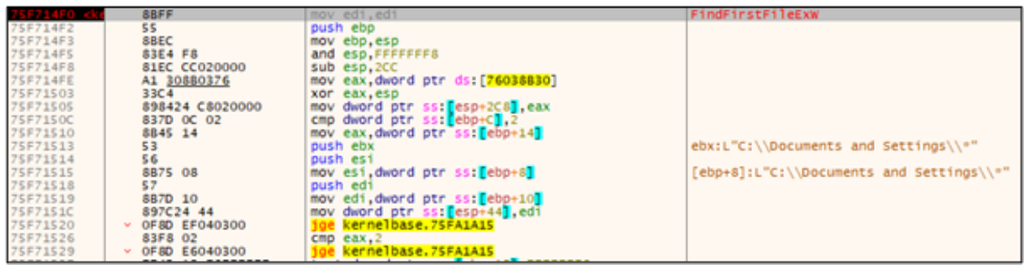


[그림 19] RegAsm 내 와이퍼 코드 삽입

RegAsm을 실행하고 이전의 L 파일에서 획득한 와이퍼 악성코드를 삽입한다.
위와 같이 Native API를 이용하여 수행되는데,
이는 저수준의 함수 호출로서 후킹 기반 탐지를 피할 확률을 높혀준다.

3.3 RegAsm.exe 내 와이퍼 코드 분석

3.3.1 RegAsm.exe의 경로 획득



[그림 20] RegAsm.exe의 경로 획득

RegAsm.exe는 인젝터로부터 내부에 삽입된 와이퍼 코드가 실행되며
시스템 내 모든 경로를 탐색하며 파일을 파괴하기 시작한다.

3.3.2 RegAsm.exe의 파일 타입 확인

```
75F6E640 <ke    8BFF        mov edi,edi                                  CreateFileW
75F6E642        55          push ebp
75F6E643        8BEC        mov ebp,esp
75F6E645        83E4 F8     and esp,FFFFFFF8
75F6E648        83EC 18     sub esp,18
75F6E64B        8B4D 1C     mov ecx,dword ptr ss:[ebp+1C]
75F6E64E        8BC1        mov eax,ecx                                  eax:&"ADVAPI32.DLL"
75F6E650        25 B77F0000 and eax,7FB7                                 eax:&"ADVAPI32.DLL"

75F82D20 <ke    8BFF        mov edi,edi                                  GetFileType
75F82D22        55          push ebp
75F82D23        8BEC        mov ebp,esp
75F82D25        8B45 08     mov eax,dword ptr ss:[ebp+8]
75F82D28        83EC 10     sub esp,10
75F82D2B        56          push esi
75F82D2C        33F6        xor esi,esi
75F82D2E        85C0        test eax,eax
75F82D30      ~ 0F84 85000000 je kernelbase.75F82DBB
75F82D36        83F8 F4     cmp eax,FFFFFFF4
75F82D39      ~ 0F84 9FCE0200 je kernelbase.75FAFBDE
75F82D3F        83F8 F5     cmp eax,FFFFFFF5
75F82D42      ~ 0F84 85CE0200 je kernelbase.75FAFBCD
```

[그림 21] RegAsm.exe의 파일 타입 획득

RegAsm.exe는 경로 내 파일이 존재하면 해당 파일의 타입을 확인하고
SYS 파일 외 등 시스템에 장애를 발생시키지 않는 확장자가 발견되면 이를 파괴한다.

3.3.3 RegAsm.exe의 파일 파괴

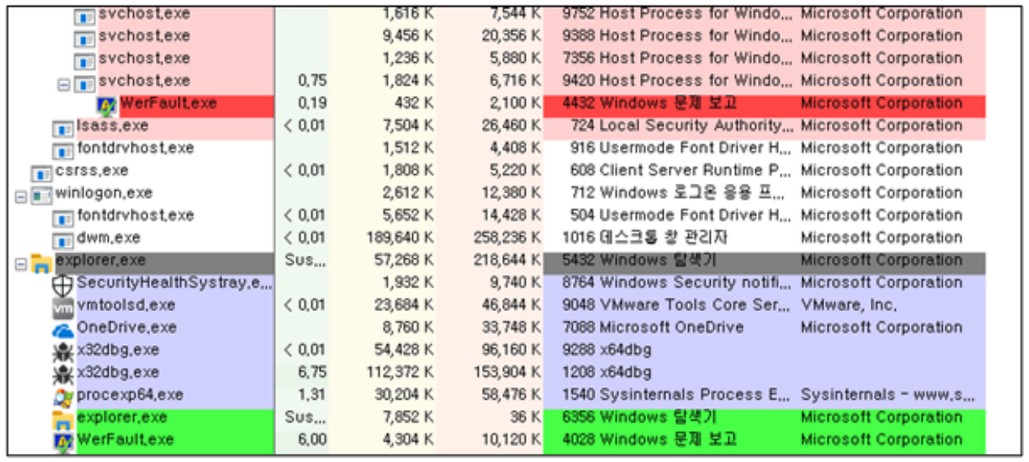
```
77C11680 <nt    B8 08001A00     mov eax,1A0008                           ZwWriteFile
77C11685        BA 7071C277     mov edx,ntdll.77C27170
77C1168A        FFD2            call edx
77C1168C        C2 2400         ret 24

0969EBFC  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0969EC0C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0969EC1C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0969EC2C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0969EC3C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0969EC4C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0969EC5C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0969EC6C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0969EC7C  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
```

[그림 22] RegAsm.exe의 파일 파괴

RegAsm.exe 내 와이퍼 악성코드는 대상 파일을 0으로 파일 크기만큼 덮어씌우며
모든 데이터를 파괴한다.
이후 대상 파일이 다 덮어씌워지면 해당 파일을 제거한다.

3.3.4 시스템 장애/파괴



[그림 23] 와이퍼로 유발된 시스템 장애/파괴

경로 내 존재하는 대부분의 파일을 파괴하여 시스템에는 심각한 장애가 유발된다.
위와 같이 피해 PC 내 존재하는 대부분의 파일은 파괴되어 정상적인 동작을 할 수가 없게 된다.

본 와이퍼 악성코드는 해외 유명 보안 업체인
크라우드 스트라이크의 소프트웨어 장애 사건을 이용하여 발생한 공격이다.
와이퍼 악성코드를 업데이트 파일로 위장하여 피해자에게 전달하였으며,
피해자가 직접 클릭하여 와이퍼 악성코드를 실행하도록 유도하였다.

그 결과로 피해 PC들은 시스템이 망가져 해당 PC 내 데이터는 복구할 수 없는 상황이 발생했다.

이와 같은 와이퍼 공격이 최근 빈번하게 발생함에 따라
소만사 악성코드 분석 센터는 와이퍼 악성코드의 특징을 간단히 설명한다.

[목표]

- 데이터 삭제: 와이퍼는 디스크 드라이브에 저장된 파일과 데이터를 완전히 삭제하여 복구가 불가능하게 만든다. 이는 데이터를 삭제하는 것이 아니라 파일 시스템과 파일 내용을 물리적으로 덮어쓰거나 삭제하여 복구가 불가능하도록 한다.
- 시스템 파괴: 일부 와이퍼는 운영 체제나 시스템 파일을 손상시키거나 삭제하여 컴퓨터가 부팅되지 않도록 만든다. 이 경우 시스템 복구가 어렵거나 불가능할 수 있다.
- 복구 방해: 와이퍼는 종종 복구를 어렵게 하거나 불가능하게 만드는 다양한 기술을 사용한다. 예를 들어, 복구 도구나 백업 파일을 손상시키거나 삭제할 수 있다.

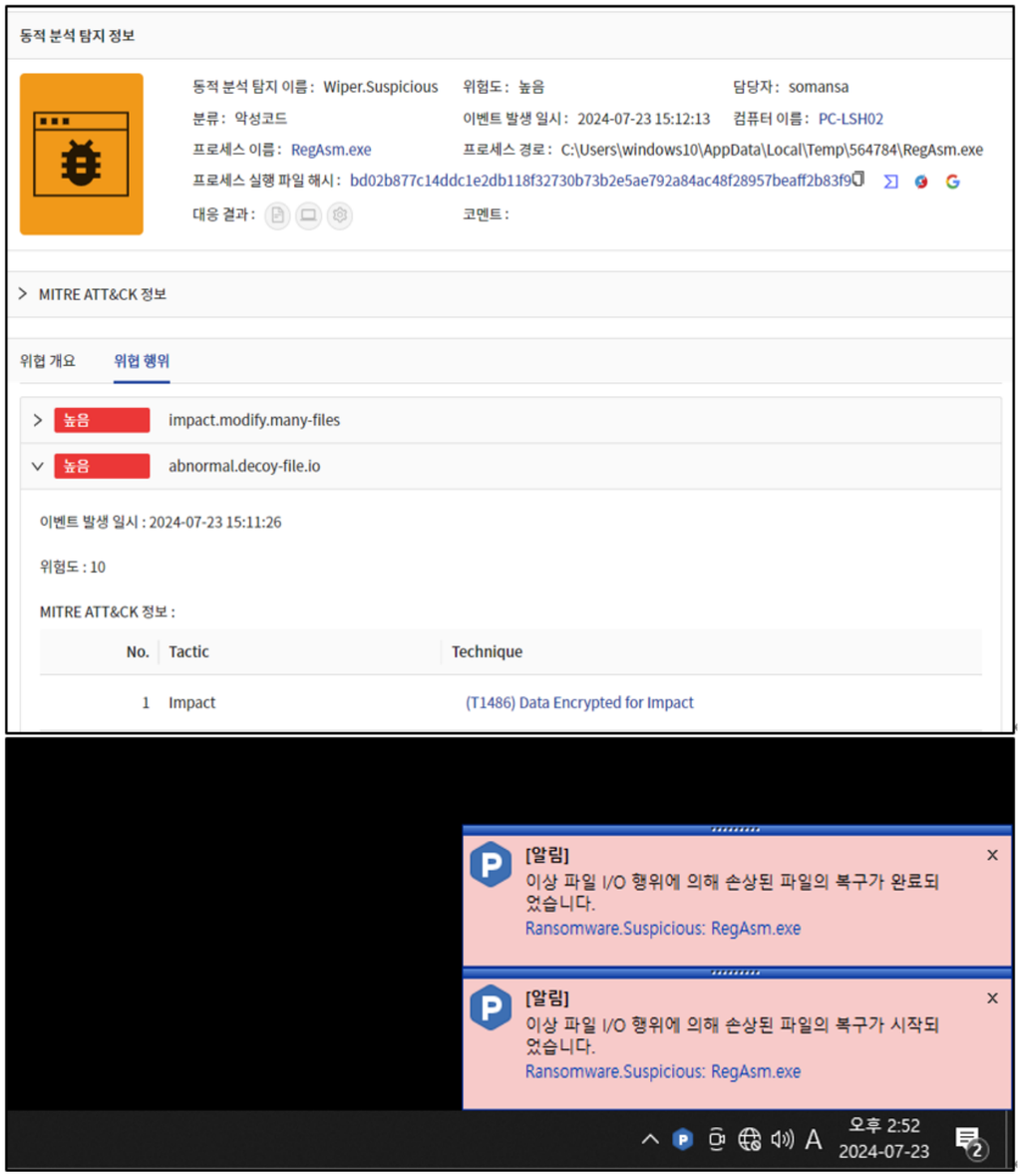
[특징]

- 목표 지향적: 와이퍼는 일반적으로 특정 조직, 기업, 국가 또는 개인을 목표로 삼는다. 종종 정치적, 경제적, 또는 사이버 전쟁의 일환으로 사용된다.
- 영구적인 손상: 데이터와 시스템을 복구 불가능하게 만들기 때문에, 피해를 본 사용자는 대개 데이터 손실을 겪게 된다.
- 배포 방식: 이메일 첨부 파일, 악성 웹사이트, 또는 제로데이 취약점을 통해 배포될 수 있다.

[표 6] 소만사 악성코드 분석 센터 분석 결과

4. Privacy-i EDR 탐지 정보

4.1 이상 파일 I/O 행위 탐지를 통한 와이퍼 탐지 및 차단



[그림 24] 이상 파일 I/O 기반 와이퍼 탐지 및 차단

와이퍼 악성코드는 시스템 내부의 데이터를 파괴하고 복구가 불가한 상태로 만드는
이상 파일 I/O 행위를 수행한다.
본 소만사 제품인 Privacy-i EDR은
이러한 시스템 파괴 행위를 이상 파일 I/O 행위로 탐지 및 차단하고 시스템의 복구까지 완료하였다.

5. 대응

1. Privacy-i EDR과 같은 EDR 제품을 통해 취약점 실행을 행위 기반으로 차단
2. 주요 데이터는 주기적인 백업을 통해 시스템 파괴 시에도 복구가 가능하도록 대비
3. 논리적 망분리를 적용하여 악성코드 PC 유입을 원천 차단
4. AV(패턴기반탐지) + EDR(행위기반탐지) 솔루션
5. PC 취약점을 주기적으로 점검, 보완
6. 신뢰할 수 없는 메일의 첨부파일 실행 금지
7. 비업무 사이트 및 신뢰할 수 없는 웹사이트의 연결 차단
8. OS나 어플리케이션은 최신 형상 유지

궁금하신 점이나 문의사항은 malware@somansa.com 으로 문의주십시오