伊 아웃도어 브랜드 몽클레어 데이터 유출 및 암호화 주범
현재까지 약 60개 기업/기관 피해발생

Rust 언어로 개발 및 유포된 최초의 랜섬웨어 BlackCat (ALPHV)

목차

1. 개요

1.1 배경

BlackCat(ALPHV) 랜섬웨어는 2021년 11월 중순에 등장했으며 지금까지 다양한 변종이 발견되고 있다.
윈도우, 리눅스 등 운영체제를 가리지 않고 변종이 등장하는 것이 특징이다.
프로그래밍 언어 Rust를 사용해 개발되었기 때문이다.
Rust는 C와 C++ 처럼 빠르게 동작하면서, 한 가지 소스코드를 활용해
여러 운영체제에서 동작 가능한 실행 파일을 만들 수 있다.
그러나 Rust로 개발된 프로그램의 경우 메모리 관리 로직이 복잡해
악성코드 분석에 보다 많은 시간이 소요된다.

[그림 1] BlackCat 랜섬웨어 유출 정보 페이지

최근 FBI는 2022년 3월 기준 최소 60개 조직이 BlackCat 랜섬웨어 피해를 입었다고 발표했다.
실제 공격자들이 BlackCat 랜섬웨어에 감염된 기업의 내부 데이터를 일부 공개해
금전을 요구하는 협박 페이지를 제작한 것을 보면, 피해 기업이 지속적으로 발생하고 있음을 알 수 있다.

2. 파일정보

2.1 파일정보

Name	[Random].dll
Type	Portable Executable File
Behavior	Ransomware
SHA-256	f2b3f1ed693021b20f456a058b86b08abfc4876c7a3ae18aea6e95567fd55b2e

[파일 1] Rust 언어로 개발된 BlackCat 랜섬웨어

3. 공격 정보

[그림 2] BlackCat 랜섬웨어 행위 흐름

①	랜섬웨어의 동작에 필요한 설정값을 해석하고, 피해 단말기를 식별하기 위한 작업이 수행된다.
②	피해 단말기에서 윈도우 권한 상승 취약점을 악용해 높은 권한을 획득한다.
③	데이터 복구와 악성 행위 추적을 방해하기 위해 복구 데이터와 보안 이벤트 로그를 제거한다.
④	파일 암호화에 방해가 될 수 있는 프로세스와 서비스를 종료한다.
⑤	네트워크에 연결된 공유 서버에 접근을 시도하고 랜섬웨어를 전파한다.
⑥	피해 단말기에 존재하는 파일을 조회하고, 데이터를 이용할 수 없도록 암호화한다.

⑦	피해자에게 데이터 복구를 빌미로 금전을 요구하기 위한 정보인 랜섬 노트를 생성한다.
⑧	데이터 복구와 악성 행위 추적을 방해하기 위해 복구 데이터와 보안 이벤트 로그를 제거한다.
⑨	모든 악성 행위를 수행하고 종료된다.

3.1 랜섬웨어 설정 값 로드

[그림 3] 랜섬웨어 동작을 위해 설정 로드

BlackCat 랜섬웨어는 동작을 수행하기 위해 실행파일 내부에 설정값을 가지고 있다.
이 설정값에는 랜섬웨어 확장자, 공격자의 공개키, 암호화 제외 대상 폴더/파일 이름 등
랜섬웨어가 동작하기 위한 주요 정보가 포함되어 있다.

[표 1] 설정 값의 일부	
공격자의 공개키	MIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIBCgKCAQEAvcNJz4PSogSJ3Df7ho/Fdl3RTZYU1zY2sY6lojUACXaPYhaKpjJ/My5VZHotI7xo2Qml8OdWgB1pYABHYVgB5G9KCjXw7/kbSaObK+NBiokPzg9Tpc0BIYZt1z1VvsR/tnyx4hs/U3N1eMqqmOfEI2tX9GDEAZzC6XuuW50sQEF/noVO8RZi9InNIxKxMGhZ1TGRq3XkPypr8JYNoBisL3Yf3JhqfiH6sCQbXybrH9d05B8zP+jfyMbwDT3p+YMIuHEnnP86JIBZ7CuBAd0VTjC6IBYIUDorSSF3qGD2Me4ajVkOBdo4/dPBgfyqJJQLvC8fD5muwkFH2/0QPDv/KwIDAQAB
랜섬웨어 확장자	grp3smk
랜섬 노트 이름	RECOVER-${EXTENSION}-FILES.txt
랜섬 노트 템플릿	Hello, CTConventions\n\n» What happened?\n\nImportant files on your network was ENCRYPTED and now they have \"${EXTENSION}\" extension.\nIn order to recover your files you need to follow instructions below.\n\n» Sensitive Data\n\nSensitive data on your network was DOWNLOADED.\nIf you DON'T WANT your sensitive data to be PUBLISHED you have to act quickly.\n\nData includes:\n- MICROS DATABASE, Accounting, Drawings\n- Check Copies, Engineering, HR, Banking Information\n- Payroll Scan, Sales and Marketing, Financia\n- And more...\n\n» CAUTION\n\nDO NOT MODIFY ENCRYPTED FILES YOURSELF.\nDO NOT USE THIRD PARTY SOFTWARE TO RESTORE YOUR DATA.\nYOU MAY DAMAGE YOUR FILES, IT WILL RESULT IN PERMANENT DATA LOSS.\n\n» What should I do next?\n\n1) Download and install Tor Browser from: https://torproject.org/\n2) Navigate to: http://d75itpgjjfe2ys2qivqplbvmw3yyx7o5e4ppt2esit2lluhngulz4hqd.onion/?access-key=${ACCESS_KEY}
바탕화면 노트 템플릿	Important files on your network was DOWNLOADED and ENCRYPTED.\nSee \${NOTE_FILE_NAME}\" file to get further instructions."

3.2 피해자 단말기 고유 식별자 수집 및 토큰(ACCESS_KEY) 생성

```
004F8095    56              push esi
004F8096    68 19000200     push 20019
004F809B    6A 00           push 0
004F809D    57              push edi                    edi:L"SOFTWARE\\Microsoft\\Cryptography"
004F809E    68 02000080     push 80000002
004F80A3    E8 80460F00     call <JMP.&RegOpenKeyExW>

004F8194    50              push eax
004F8195    FF7424 10       push dword ptr ss:[esp+10]
004F8199    53              push ebx
004F819A    6A 00           push 0
004F819C    56              push esi                    esi:L"MachineGuid"
004F819D    57              push edi
004F819E    E8 8D450F00     call <JMP.&RegQueryValueExW>
```

[그림 4] 고유 식별자 수집을 위한 레지스트리 접근

```
004F8CA9    8D4C24 54       lea ecx,dword ptr ss:[esp+54]
004F8CAD    BA 64D06600     mov edx,target.66D064       edx:"wmic csproduct get UUID", 66D064:"wmic
004F8CB2    6A 17           push 17
004F8CB4    E8 175DFFFF     call target.4EE9D0
```

[그림 5] 고유 식별자 수집을 위한 WMIC 실행

피해자 단말기의 고유한 식별자를 수집하기 위해 “SOFTWARE\Microsoft\Cryptography” 하위 “MachineGuid”를 조회한다. 이 과정에서 오류가 발생하면 WMIC를 이용해 고유 식별자를 수집한다.

```
0048366A    8D8C24 B0000000 lea ecx,dword ptr ss:[esp+B0]
00483671    8D9424 BC010000 lea edx,dword ptr ss:[esp+1BC]  [esp+1BC]:"FEE74D56-F341-8241-35C4
00483678    53              push ebx
00483679    56              push esi
0048367A    E8 C19C0500     call target.4DD340
0048367F    83C4 08         add esp,8
00483682    8B8424 B0000000 mov eax,dword ptr ss:[esp+B0]  [esp+B0]:"cHO9KfRIHz4ldHj%2F2iceUc
```

[그림 6] 고유 식별자를 이용한 토큰 생성

피해자 단말기의 고유 식별자를 수집 후 임의의 연산을 통해 고유한 토큰을 생성한다.
이 값은 추후 피해자에게 금전을 요구하기 위한 접속 주소로 활용되며
해당 주소는 랜섬 노트에서 확인할 수 있다.

3.3 랜섬노트 데이터 수정

```
0048370C    83C4 0C         add esp,C
0048370F    8B4424 24       mov eax,dword ptr ss:[esp+24]
00483713    8D8C24 6C020000 lea ecx,dword ptr ss:[esp+26C]
0048371A    89FA            mov edx,edi                 edx:"RECOVER-${EXTENSION}-FILES.txt", edi:"RECOVER-
0048371C    FF70 2C         push dword ptr ds:[eax+2C]
0048371F    FF70 24         push dword ptr ds:[eax+24]  [eax+24]:"grp3smk"
00483722    6A 0C           push C
00483724    68 B87F6000     push target.607FB8          607FB8:"${EXTENSION}${ACCESS_KEY}${NOTE_FILE_NAME}"
00483729    56              push esi
0048372A    E8 F12F0000     call target.486720

00483794    83C4 0C         add esp,C
00483797    8D8C24 48020000 lea ecx,dword ptr ss:[esp+248]
0048379E    897424 34       mov dword ptr ss:[esp+34],esi
004837A2    89F2            mov edx,esi                 edx:"Hello, CTConventions\n\n>> What happened?\n\nImp
004837A4    FF3424          push dword ptr ss:[esp]
004837A7    FF7424 20       push dword ptr ss:[esp+20]
004837AB    6A 0D           push D
004837AD    68 C47F6000     push target.607FC4          607FC4:"${ACCESS_KEY}${NOTE_FILE_NAME}"
004837B2    895C24 24       mov dword ptr ss:[esp+24],ebx
004837B6    53              push ebx
004837B7    E8 642F0000     call target.486720

00483869    83C4 0C         add esp,C
0048386C    8D8C24 B0000000 lea ecx,dword ptr ss:[esp+B0]
00483873    89FA            mov edx,edi                 edx:"Important files on your network was DOWNLOAD
00483875    FF3424          push dword ptr ss:[esp]
00483878    FF7424 20       push dword ptr ss:[esp+20]
0048387C    6A 0D           push D
0048387E    68 C47F6000     push target.607FC4          607FC4:"${ACCESS_KEY}${NOTE_FILE_NAME}"
00483883    53              push ebx
00483884    E8 972E0000     call target.486720
```

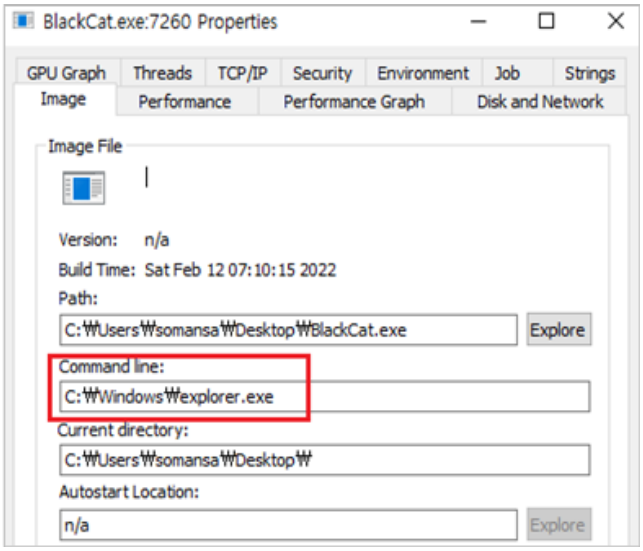
[그림 7] 템플릿 랜섬 노트 데이터 수정

설정값에 랜섬 노트 데이터가 존재한다. 하지만 이 데이터는 템플릿 형태이며
피해자에게 금전을 요구하기 위한 접속주소, 랜섬웨어 확장자 등의 데이터가 포함되어 있지 않다.
그래서 초기 동작 과정에서 템플릿 데이터를 수정해 랜섬 노트 데이터를 생성한다.

3.4 정상 프로세스로 위장하기 위한 프로세스 정보(PEB) 조작

```
dwProcessId = GetCurrentProcessId();
v8 = OpenProcess(0x438u, 0, dwProcessId);
if ( v8 != (HANDLE)-1 )
{
  v2(v8, 0, v20, 24, 0);
  if ( ReadProcessMemory(v8, BaseAddress, &Buffer, 4u, 0) )
  {
    if ( ReadProcessMemory(v8, &Buffer->Ldr, &v17, 4u, 0) )
    {
      GetWindowsDirectoryW(Destination, 0x104u);
      wcscat_s(Destination, 0x105u, L"\\explorer.exe");
      v15 = (wchar_t *)malloc(0x104u);
      wcscpy_s(v15, 0x104u, Destination);
      ((void (__stdcall *)(struct _RTL_CRITICAL_SECTION *, LPCSTR))v4)(Buffer->FastPebLock, lpProcName);
      ((void (__stdcall *)(int))v13)(*(_DWORD *)(v17 + 0x10) + 0x38);
      v13(&Buffer->ProcessParameters->CommandLine, v15);
```

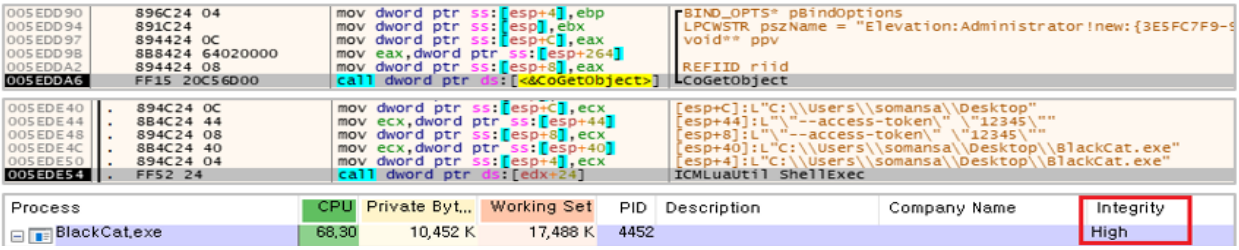
[그림 8] PEB 데이터를 이용한 프로세스 정보 조작



[그림 9] PEB 데이터를 이용한 프로세스 정보 조작

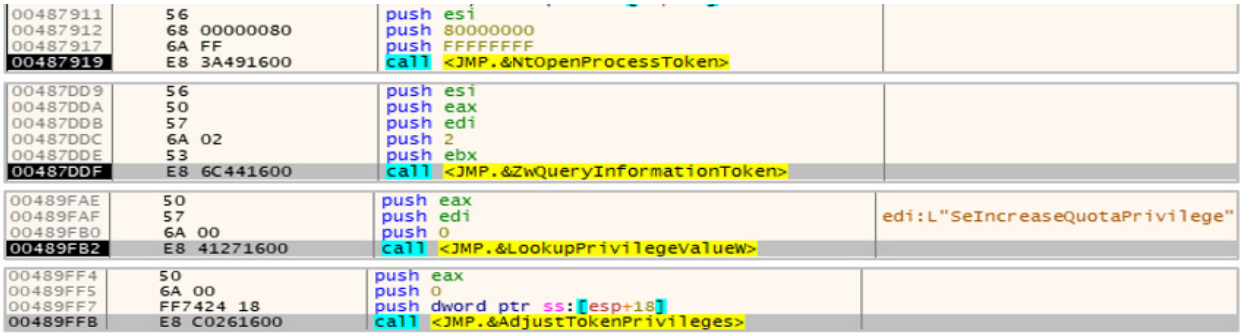
악성 행위를 수행하는 과정에서 보안 장비를 우회하고 사용자의 눈을 피하기 위해
프로세스 정보를 조작하는 과정이 수행된다.
이 과정에서 윈도우의 정상적인 프로세스인 “explorer.exe”로 프로세스 정보를 위장한다.

3.5 프로세스 권한 상승 (UAC 우회)



[그림 10] CMSTPLUA COM 객체를 이용한 권한 상승

랜섬웨어가 동작하며 시스템의 주요 데이터에 접근하기 위해 높은 권한이 필요하다.
BlackCat 랜섬웨어는 윈도우에서 관리자 권한으로 동작 중인 CMSTPLUA COM 객체의
문서화되지 않은 ShellExec 함수를 이용하는 취약점으로 높은 권한을 획득한다.
이러한 과정을 통해 권한을 상승하는 이유는 사용자 알림(UAC)이 발생하지 않아
피해자의 눈을 피해 높은 권한을 획득해 악성 행위를 수행할 수 있기 때문이다.



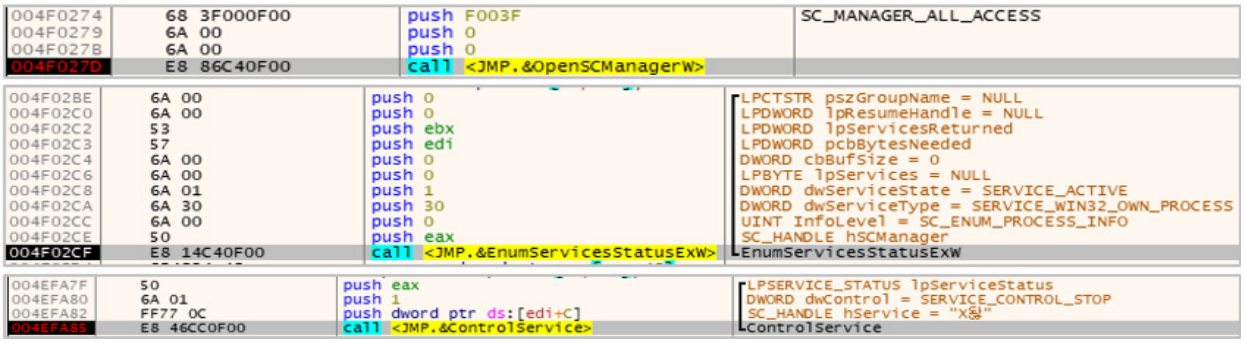
[그림 11] 프로세스 권한 획득

이후 랜섬웨어의 동작 안정성을 높이기 위해 프로세스가 획득 가능한 모든 권한을 요구하는데
이 과정에서 요청하는 권한은 아래 표와 같다.

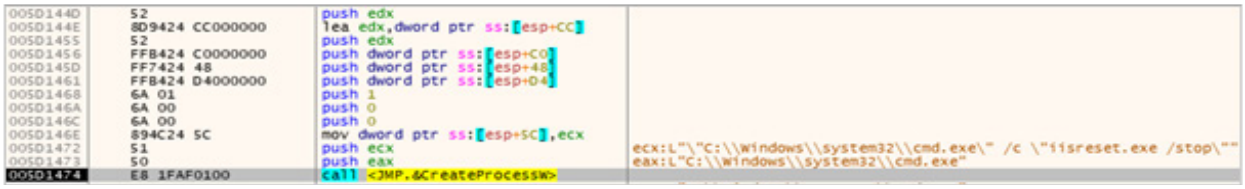
[표 2] 요청하는 프로세스 권한 목록

SeSecurityPrivilege, SeTakeOwnershipPrivilege, SeLoadDriverPrivilege, SeSystemProfilePrivilege, SeSystemtimePrivilege, SeProfileSingleProcessPrivilege, SeIncreaseBasePriorityPrivilege, SeCreatePagefilePrivilege, SeBackupPrivilege, SeRestorePrivilege, SeShutdownPrivilege, SeDebugPrivilege, SeSystemEnvironmentPrivilege, SeChangeNotifyPrivilege, SeRemoteShutdownPrivilege, SeUndockPrivilege, SeManageVolumePrivilege, SeImpersonatePrivilege, SeCreateGlobalPrivilege, SeIncreaseWorkingSetPrivilege, SeTimeZonePrivilege, SeCreateSymbolicLinkPrivilege, SeDelegateSessionUserImpersonatePrivilege, SeIncreaseQuotaPrivilege

3.6 서비스 종료



[그림 12] 윈도우에 등록된 서비스 종료

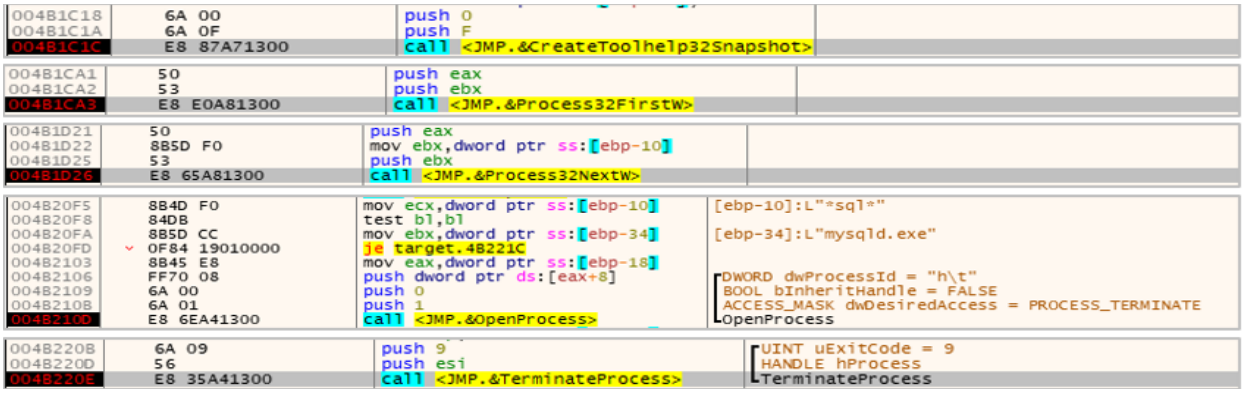


[그림 13] IIS 서비스 종료

랜섬웨어가 파일을 암호화하는 과정에서 데이터베이스 혹은 백업 관련 서비스가 동작중이면,
다른 서비스에서 이미 파일의 접근 권한을 가지고있어 암호화하지 못하는 경우가 발생할 수 있다.
이러한 현상을 방지하기위해 관련 서비스를 비활성화 한다. 종료 대상 서비스는 아래 표와 같다.

[표 3] 종료 대상 서비스
"mepocs", "memtas", "veeam", "svc$", "backup", "sql", "vss", "msexchange", "sql$", "mysql", "mysql$", "sophos", "MSExchange", "MSExchange$", "WSBExchange", "PDVFSService", "BackupExecVSSProvider", "BackupExecAgentAccelerator", "BackupExecAgentBrowser", "BackupExecDiveciMediaService", "BackupExecJobEngine", "BackupExecManagementService", "BackupExecRPCService", "GxBlr", "GxVss", "GxCIMgrS", "GxCVD", "GxCIMgr", "GXMMM", "GxVssHWProv", "GxFWD", "SAPService", "SAP", "SAP$", "SAPD$", "SAPHostControl", "SAPHostExec", "QBCFMonitorService", "QBDBMgrN", "QBIDPService", "AcronisAgent", "VeeamNFSSvc", "VeeamDeploymentService", "VeeamTransportSvc", "MVArmor", "MVarmor64", "VSNAPVSS", "AcrSch2Svc"

3.7 프로세스 종료

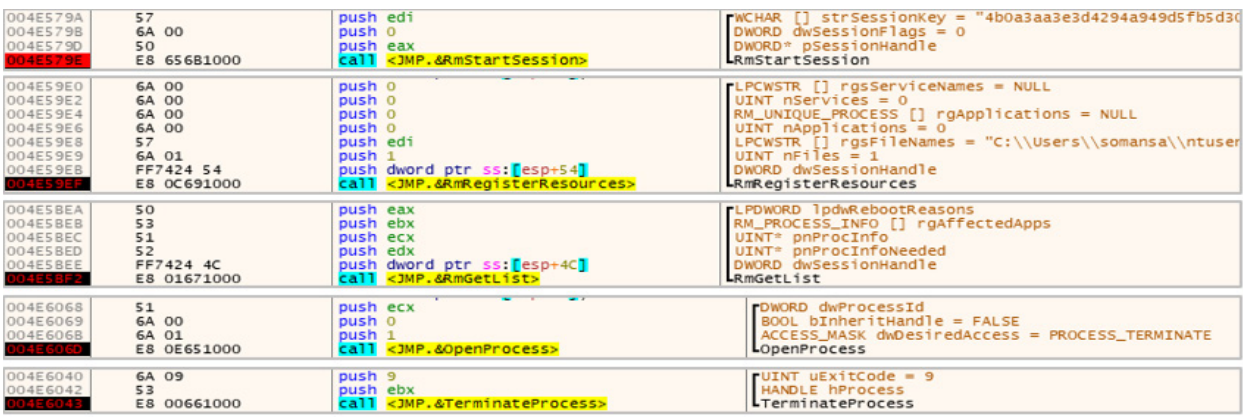


[그림 14] 타겟 프로세스 종료

랜섬웨어가 파일을 암호화하는 과정에서 다른 프로세스가 파일의 접근 권한을 점유하고 있으면,
암호화에 실패하는 경우가 발생할 수 있어 대상 프로세스는 종료된다.
주로 문서 편집, 데이터베이스, 보안 프로그램을 종료하며 종료 대상 프로세스는 아래 표와 같다.

[표 4] 종료 대상 프로세스
"agntsvc", "dbeng50", "dbsnmp", "encsvc", "excel", "firefox", "infopath", "isqlplussvc", "msaccess", "mspub", "mydesktopqos", "mydesktopservice", "notepad", "ocautoupds", "ocomm", "ocssd", "onenote", "oracle", "outlook", "powerpnt", "sqbcoreservice", "sql", "steam", "synctime", "tbirdconfig", "thebat", "thunderbird", "visio", "winword", "wordpad", "xfssvccon", "*sql*", "bedbh", "vxmon", "benetns", "bengien", "pvlsvr", "beserver", "raw_agent_svc", "vsnapvss", "CagService", "QBIDPService", "QBDBMgrN", "QBCFMonitorService", "SAP", "TeamViewer_Service", "TeamViewer", "tv_w32", "tv_x64", "CVMountd", "cvd", "cvfwd", "CVODS", "saphostexec", "saposcol", "sapstartsrv", "avagent", "avscc", "DellSystemDetect", "EnterpriseClient", "VeeamNFSSvc", "VeeamTransportSvc", "VeeamDeploymentSvc", "mbam.exe"

3.8 파일 접근 권한 점유 프로세스 종료



[그림 15] 피해자 단말기의 파일을 점유하고 있는 프로세스 종료

종료 대상 프로세스가 아니더라도 암호화 대상 파일의 접근 권한을 점유하고 있어
파일 암호화에 방해가 되는 프로세스를 대상으로 종료를 시도한다.

3.9 볼륨 섀도우 카피 데이터 제거

[그림 16] VSSADMIN을 이용한 볼륨 섀도우 카피 제거

[그림 17] WMI를 이용한 볼륨 섀도우 카피 제거

랜섬웨어에 감염되기 전 볼륨 섀도우 카피 데이터가 존재하면
피해자의 데이터를 복구할 수 있는 가능성이 존재한다.
공격자는 피해자의 데이터가 복구되는 가능성을 낮추기 위해 볼륨 섀도우 카피 데이터 삭제를 시도한다.

3.10 시스템 복구 비활성화

[그림 18] 시스템 복구 비활성화

랜섬웨어가 감염되기 전 시스템 복구 데이터를 이용해 피해자가 파일을 복구할 수 있는 가능성이 존재한다.
공격자는 데이터가 복구되는 가능성을 낮추기 위해 시스템 복구를 비활성화한다.

3.11 윈도우 관리 이벤트 제거

[그림 19] 윈도우 관리 이벤트 제거

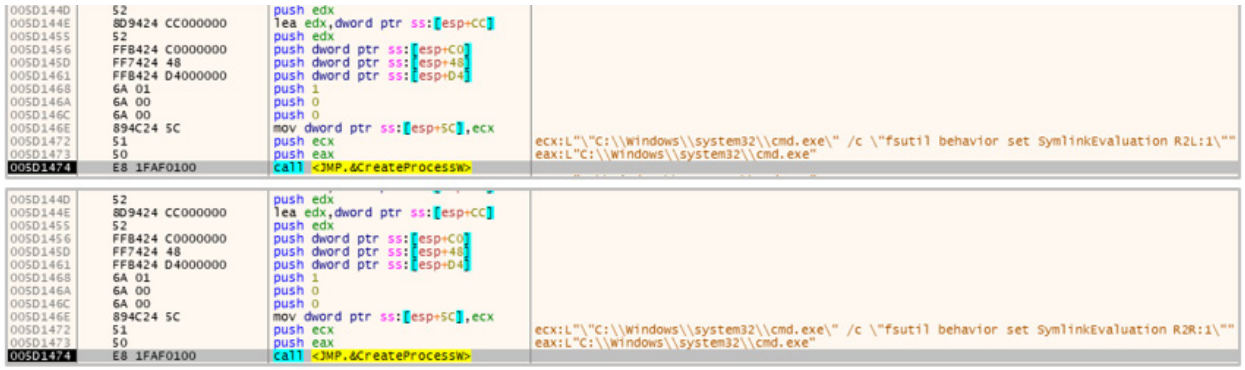
랜섬웨어가 동작하며 피해자 단말기에 남기는 흔적을 제거하기 위해
윈도우 관리 이벤트를 제거하는 행위를 수행한다.
윈도우 관리 이벤트가 제거되면 악성코드의 감염 원인, 피해 규모 등을 추적하는데 어려움이 발생할 수 있다.

3.12 네트워크로 연결 가능한 단말기 조회

[그림 20] 네트워크로 연결 가능한 단말기 정보 수집

랜섬웨어가 네트워크를 이용한 전파를 수행하기 전에
피해 단말기와 네트워크로 연결된 단말기를 확인하는 과정을 수행한다.
이때 ARP 테이블을 조회하며, ARP 테이블은 내부망에 존재하는 단말기의 네트워크 정보를 저장하고 있다.

3.13 심볼릭 링크 원격 연결 활성화



[그림 21] 심볼릭 링크 원격 연결 활성화

윈도우는 기본적으로 원격 단말기의 심볼릭 링크에 접근할 수 없도록 보안 설정되어 있다.
하지만 fsutil 명령을 이용하면 원격 단말기의 심볼릭 링크에 접근할 수 있도록 설정할 수 있다.

랜섬웨어가 이러한 행위를 수행하는 이유는 피해자 단말기에 연결된 원격 단말기에 대한 심볼릭 링크에 접근해 더 많은 파일을 암호화하기 위함으로 판단할 수 있다.

3.14 윈도우 자격 증명과 네트워크를 이용한 전파

[그림 22] MaxMpxCt 최대값 65535 설정

BlackCat 랜섬웨어는 피해를 확산시키기 위해 네트워크를 이용하여 전파된다.
이 과정에서 네트워크 최대 연결 수 제한으로 전파 속도가 늦춰지는 상황을 회피하기 위해 네트워크 연결에 관련된 레지스트리 값을 최대로 설정한다.

[그림 23] 네트워크 공유 서버를 이용한 전파

이후 피해 단말기에서 연결 가능한 네트워크 공유 서버를 열거하고,
랜섬웨어 설정에 가지고 있는 윈도우 자격 증명과 “NET USE” 명령을 이용해 공유 서버에 인증을 시도한다.
인증이 성공하면 네트워크 공유 드라이브가 연결되고 공유 서버의 파일이 암호화된다.

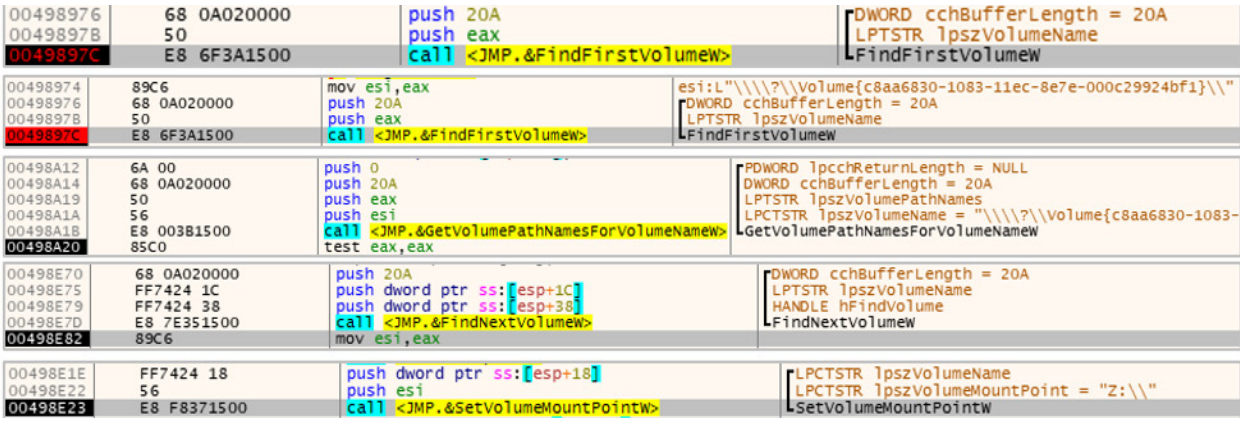
[그림 24] PSEXEC를 이용한 전파

"NET USE" 명령을 이용한 전파 방식 이외에도, 랜섬웨어가 TEMP 폴더에 생성한 PSEXEC 도구와 랜섬웨어 설정에 가지고 있는 윈도우 자격 증명을 이용해 네트워크 전파를 시도한다.
이 과정을 통해 윈도우 자격증명 인증에 성공하면 네트워크에 연결된 원격 단말기에 BlackCat 랜섬웨어 실행파일이 전송되고 피해가 확산된다.

[표 5] 랜섬웨어 설정에 존재하는 윈도우 자격 증명

"ctconventions\\administrator" : "9ThingThousandPortugal^"
"ctconventions\\kdanforth" : "I)eltaFlyer5"
"ctconventions\\whgadmin" : "@C232323c"
"ctconventions\\glackey" : "P@55Me2021$"
"ctconventions\\walkergroup" : "2EnjoyAnythingCreate"
"ctconventions\\ctcc" : "CTcc2629"
"ctconventions\\MasterAccount" : "micros000000"

3.15 숨겨진 볼륨 마운트



[그림 25] 피해자 단말기에 연결된 숨겨진 볼륨 마운트

[그림 26] 랜섬웨어에 의해 마운트 된 Y:, Z: 볼륨

랜섬웨어는 피해자 단말기에서 최대한 많은 데이터를 암호화하기 위해
숨겨진 볼륨을 조회, 마운트 하는 과정을 수행한다.
위 그림은 운영체제 볼륨으로 숨겨진 Y:, Z: 볼륨이 마운트 된 상황이다.

3.16 암호화 대상 파일 탐색

```
0049902F    E8 84341500     call <JMP.&GetLogicalDrives>
00499034    894424 0C       mov dword ptr ss:[esp+C],eax        C:, D:, Y:, Z:

005D2BF2    56              push esi                            [Drive]:\Users
005D2BF3    50              push eax
005D2BF4    E8 2711FFFF     call target.5C3D20
```

[그림 27] 피해자 단말기에 연결된 볼륨 조회

피해 단말기에 연결된 논리적 드라이브를 조회한다.
그 결과 반환받은 드라이브 경로에 “Users” 경로를 덧붙여 해당하는 경로가 존재하는지 확인한다.
“Users” 디렉토리를 조회하는 행위는 사용자 데이터를 우선적으로 암호화하기 위한 것으로 추정된다.

```
005D95DF    53              push ebx                            LPWIN32_FIND_DATA lpFindFileData
005D95E0    56              push esi                            LPCTSTR lpFileName = "Y:\\*"
005D95E1    E8 022E0100     call <JMP.&FindFirstFileW>          FindFirstFileW

005C792F    53              push ebx                            LPWIN32_FIND_DATA lpFindFileData
005C7930    57              push edi                            HANDLE hFindFile
005C7931    E8 C24A0200     call <JMP.&FindNextFileW>           FindNextFileW
```

[그림 28] 윈도우 API를 이용한 파일 탐색

조회된 볼륨 드라이버 최상위 디렉토리와 “Users” 디렉토리 하위에 존재하는 경로를 순회하며
암호화 대상 파일을 수집한다.

```
if ( GetFileInformationByHandle(a2, &FileInformation) )
{
  dwFileAttributes = FileInformation.dwFileAttributes;
  result = 0;
  if ( (FileInformation.dwFileAttributes & FILE_ATTRIBUTE_REPARSE_POINT) != 0 )
  {
    memset(OutBuffera, 0, 0x4000u);
    BytesReturned = 0;
    if ( DeviceIoControl(a2, FSCTL_GET_REPARSE_POINT, 0, 0, OutBuffera, 0x4000u, &BytesReturned, 0) )
    {
      result = OutBuffera[0];
    }
```

```
004CC8DF    50              push eax                            DWORD dwFileAttributes
004CC8E0    56              push esi                            LPCTSTR lpFileName = "C:\\Users\\Default User"
004CC8E1    E8 0AFD1100     call <JMP.&SetFileAttributesW>      SetFileAttributesW
```

[그림 29] 파일 속성에 따른 권한 설정

암호화 대상을 조회하는 과정에서 파일의 속성을 확인하는 루틴이 존재하며,
해당 파일이 링크 파일인 경우 파일을 직접 접근할 수 있도록 권한을 수정한다.

3.17 데이터 암호화

```
005D9290    6A 02           push 2                              ULONG dwFlags = 2
005D9292    6A 10           push 10                             ULONG cbBuffer = 10
005D9294    50              push eax                            PUCHAR pbBuffer
005D9295    6A 00           push 0                              BCRYPT_ALG_HANDLE hAlgorithm = NULL
005D9297    E8 1C340100     call <JMP.&BCryptGenRandom>         BCryptGenRandom
```

[그림 30] AES 암호화를 위한 키 생성

```
qmemcpy(v187, "nd 32-byexpa", sizeof(v187));
v173 = 1797285236;
qmemcpy(v180, "expand 3expa", sizeof(v180));
qmemcpy(v189, "nd 32-byte k2-bynd 32-byte k", sizeof(v189));
```

[그림 31] ChaCha20 암호화를 위한 키 생성

```
if ( *(_BYTE *)(a2 + 52) == 1 )
  mw_ConcatStringWithQuotes("ChaCha20Aes", p_Buffer, (char *)8);
else
  mw_ConcatStringWithQuotes("Aes", p_Buffer, (char *)3);
```

[그림 32] 2가지 종류의 파일 암호화 알고리즘

파일 암호화 방식은 2가지를 지원하며, AES 암호화를 기본으로 사용하지만
키 생성 등 암호화 과정에서 오류가 발생할 경우 ChaCha20 암호화 알고리즘을 사용한다.

Address	Hex								ASCII
00165230	7B 22 76 65	72 73 69 6F	6E 22 3A 30	2C 22 6D 6F	{"version":0,"mo
00165240	64 65 22 3A	22 46 75 6C	6C 22 2C 22	63 69 70 68	de":"Full","ciph
00165250	65 72 22 3A	22 41 65 73	22 2C 22 70	72 69 76 61	er":"Aes","priva
00165260	74 65 5F 6B	65 79 22 3A	5B 34 31 2C	31 30 2C 31	te_key":[41,10,1
00165270	39 36 2C 36	32 2C 31 33	33 2C 31 37	38 2C 35 2C	96,62,133,178,5,
00165280	31 34 32 2C	32 34 31 2C	32 32 2C 32	33 39 2C 36	142,241,22,239,6
00165290	32 2C 31 30	33 2C 31 38	39 2C 35 2C	32 31 37 5D	2,103,189,5,217]
001652A0	2C 22 64 61	74 61 5F 73	69 7A 65 22	3A 33 31 37	,"data_size":317
001652B0	30 33 30 34	2C 22 63 68	75 6E 6B 5F	73 69 7A 65	0304,"chunk_size
001652C0	22 3A 32 35	33 36 32 38	31 36 2C 22	66 69 6E 69	":25362816,"fini
001652D0	73 68 65 64	22 3A 66 61	6C 73 65 7D	00 00 00 00	shed":false}....

[그림 33] 암호화 파일 메타 데이터

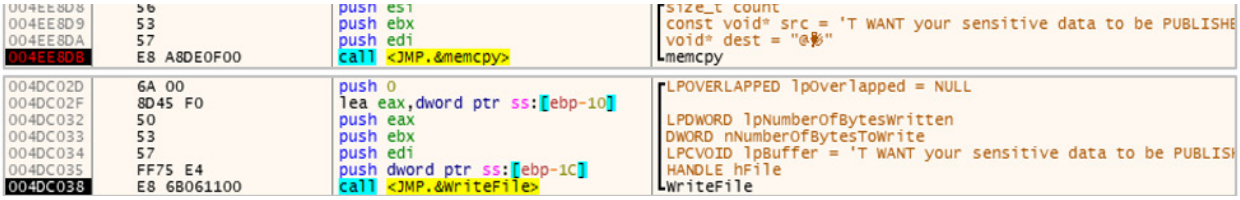
암호화된 파일을 복호화 하기 위한 메타 데이터는 위 그림과 같이 JSON 데이터이며,
파일 복호화에 필요한 암호화 키를 포함하고 있다.
이 데이터는 공격자의 RSA 공개키로 암호화 후 암호화된 파일의 끝 부분에 저장한다.

[표 6] 암호화 제외 폴더
"system volume information", "intel", "$windows.~ws", "application data", "$recycle.bin", "mozil-la", "$windows.~bt", "public", "msocache", "windows", "default", "all users", "tor browser", "pro-gramdata", "boot", "config.msi", "google", "perflogs", "appdata", "windows.old"

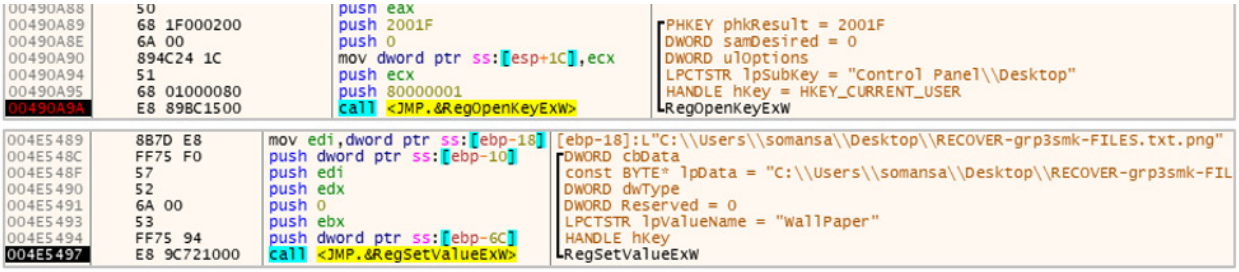
[표 7] 암호화 제외 파일
"desktop.ini", "autorun.inf", "ntldr", "bootsect.bak", "thumbs.db", "boot.ini", "ntuser.dat", "iconcache.db", "bootfont.bin", "ntuser.ini", "ntuser.dat.log"

[표 8] 암호화 제외 확장자
"themepack", "nls", "diagpkg", "msi", "lnk", "exe", "cab", "scr", "bat", "drv", "rtp", "msp", "prf", "msc", "ico", "key", "ocx", "diagcab", "diagcfg", "pdb", "wpx", "hlp", "icns", "rom", "dll", "msstyles", "mod", "ps1", "ics", "hta", "bin", "cmd", "ani", "386", "lock", "cur", "idx", "sys", "com", "deskthemepack", "shs", "ldf", "theme", "mpa", "nomedia", "spl", "cpl", "adv", "icl", "msu"

3.18 랜섬노트 생성 및 바탕화면 변경



[그림 34] 랜섬 노트 생성

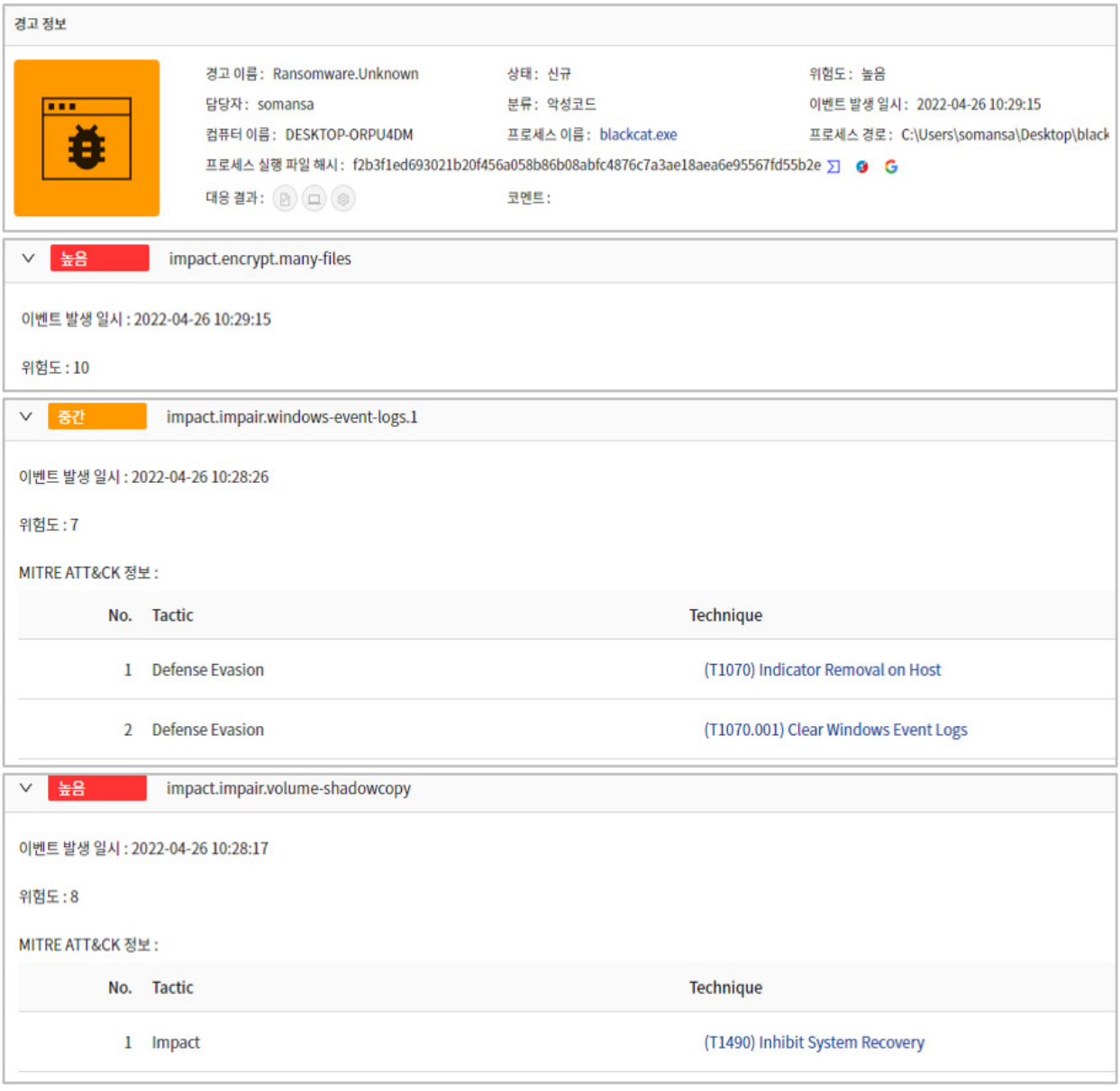


[그림 35] 피해자 단말기 바탕화면 변경

피해자 단말기의 모든 파일을 암호화 후 파일이 암호화되었다는 사실과
금전 지불 방법을 알리기 위한 랜섬 노트를 생성하고 바탕화면을 변경하는 행위를 수행한다.

4. Privacy-i EDR 탐지 정보

4.1 탐지 정보



[그림 36] Privacy-i EDR 탐지 정보

Privacy-i EDR은 파일 암호화, 윈도우 이벤트 로그 제거, 볼륨 섀도 카피 제거 등
사용자 단말기에서 발생하는 여러 가지 행위를 실시간으로 분석하여
BlackCat 랜섬웨어를 탐지 및 차단하였다.

5. 대응

1. 행위기반 악성코드 차단 솔루션 (Privacy-i EDR 등 EDR 솔루션, AV 솔루션)을 도입하여 차단한다.
2. 음란, 도박, 게임, P2P 사이트 등 악성코드 배포 온상이 되는 사이트 접속을 차단한다.
3. 망분리(논리적 망분리)를 구축하여 인터넷 접속을 통한 악성코드 유입을 원천차단한다.
4. 신뢰할 수 없는 메일의 첨부파일은 열지 않도록 한다.
5. USB 매체 사용을 차단하거나, PC로의 USB파일 복사를 차단한다.
6. OS 및 소프트웨어 보안 업데이트를 최신상태로 유지하도록 한다.

궁금하신 점이나 문의사항은 malware@somansa.com 으로 문의주십시오