이스라엘 공격 목적으로
친 팔레스타인 단체에서 개발한 인프라 파괴 목적의

BiBi 와이퍼 악성코드

요약

1. 이스라엘-팔레스타인 간 사이버 공격 목적으로
   친 팔레스타인 세력이 개발한 **신종 BiBi 와이퍼 악성코드**
   이스라엘 보안기업이 자국 기업/기관 네트워크 조사 중 발견.
2. 일반적인 랜섬웨어는 데이터변조를 통한
   복호화 비용(몸값)을 획득하는 것을 목적으로 함.
   BiBi 와이퍼 악성코드는 랜섬웨어처럼 파일변경행위를 수행하나,
   전산 인프라 파괴에 목적이 있어 국가 간 사이버 테러에서 주로 포착됨.
3. 감염 대상의 네트워크를 통해 시스템에 침투,
   PC 내 파일들을 무작위 데이터로 덮어쓰는 방식으로 손상시킴.
   데이터와 운영 체제 모두 복구 불가능.

대응 방안

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

목차

1. 개요

1.1 배경

[그림 1] 친 팔레스타인 해커와 악성코드가 사용하는 확장자 BiBi

와이퍼(Wiper) 악성코드는 데이터 변조를 통한 수익획득을 목적으로 하는 것이 아닌
오로지 인프라 파괴하는 것을 목적으로 하기 때문에 국가 간 사이버 테러에 주로 사용된다.
또한 공격 대상의 시스템에 침투하여 공격 대상의 시스템을 마비시키고,
파괴하여 복구가 불가능한 피해를 입힌다.
지난 2022년 시작된 우크라이나 – 러시아 전쟁에서
러시아의 해킹 그룹이 와이퍼(Wiper) 악성코드를 사용해 우크라이나를 공격한 사례가 있다.

소만사 악성코드 분석 센터는 이번 이스라엘-팔레스타인 전투 간 사이버 공격 사례를
관찰하는 과정에서 친팔레스타인 세력이 개발한 신종 와이퍼(Wiper) 악성코드를 확인하였다.
해당 악성코드는 랜섬웨어와 동일한 파일 변경 행위를 수행하나,
금전적 이득을 취한 후 파일을 복호화하는 과정과 흔적이 없는 것이 특징이다.
즉 랜섬웨어를 빙자한 신종 와이퍼(Wiper)가 등장한 것이다.

본 와이퍼(Wiper) 악성코드는 BiBi라는 이름을 내부적으로 사용하고 있으며,
BiBi는 이스라엘 네타냐후 총리를 희화화하는 별명으로,
친 팔레스타인 공격 그룹이 이스라엘을 대상으로 배포 및 공격하고 있음을 확인할 수 있다.
또한 해당 공격에 앞서, 리눅스 버전의 BiBi 와이퍼(Wiper)도 확인이 되었는데
이는 이스라엘 케파르 사바의 컴퓨터 보안 서비스 업체인 Security Joes의 사고 대응 팀이
이스라엘 조직의 네트워크 침입을 조사하면서 발견했다.

소만사 악성코드 분석 센터는 본 공격에 대해 국가 간 사이버 전쟁에
와이퍼(Wiper) 악성코드가 사용됨을 확인하고, 신속히 악성코드 샘플을 확보해 분석하였다.
또한 본 악성코드 분석 보고서 발간을 통해
BiBi 와이퍼(Wiper) 악성코드의 특징과 대응 방안에 대해 서술하였다.

2. 정보

2.1 파일정보

Name	[BiBi].exe
Type	Windows PE 실행 파일
Behavior	BiBi Wiper Malware
SHA-256	40417e937cd244b2f928150cae6fa0eff5551fdb401ea072f6ecdda67a747e17

[표 1] 친팔레스타인 공격 그룹의 EXE 형태의 BiBi 와이퍼(Wiper) 악성코드

2.2 BiBi 와이퍼(Wiper) 악성코드 공격 방식

[그림 2] BiBi 와이퍼(Wiper) 악성코드 공격 방식과 실제 동작 모습

① 해커의 와이퍼 악성코드 제작 : 친팔레스타인 해커의 악성코드 제작
해커는 와이퍼 악성코드를 제작하고, 대상 국가에 대해 배포를 준비한다.

② 감염 대상 국가 / 시설 / 시스템 침투 : 이스라엘 Security Joes 자료에 의한 침입 방법
감염 대상의 네트워크를 통해 시스템에 침투하여 와이퍼 악성코드를 유포 / 배포한다.

③ 와이퍼 악성코드 감염 / 실행 : 경로 인자와 함께 실행 시 해당 경로만 파괴
와이퍼 악성코드를 대상 시스템 내에서 실행한다.

④ 시스템 파괴 및 복구 불가 : 전 시스템 파괴 1번을 1라운드로 칭하며 무한한 라운드로 파괴
시스템은 BiBi 와이퍼 악성코드에 의해 파괴되어 복구가 불가능한 피해를 입는다.
이 때, PC 내 파일들은 무작위 데이터로 덮어씌워지며, 복구가 불가능하게 파괴된다.

3. 분석

3.1 인자(CommandLine) 확인

```
00007FF7BB2E34C4   48:83EC 28          sub rsp,28
00007FF7BB2E34C8   FF15 92ED0000       call qword ptr ds:[<&GetCommandLineA>]
00007FF7BB2E34CE   48:8905 13FC0100    mov qword ptr ds:[7FF7BB3030E8],rax
00007FF7BB2E34D5   FF15 8DED0000       call qword ptr ds:[<&GetCommandLineW>]
00007FF7BB2E34DB   48:8905 0EFC0100    mov qword ptr ds:[7FF7BB3030F0],rax
00007FF7BB2E34E2   B0 01               mov al,1
00007FF7BB2E34E4   48:83C4 28          add rsp,28
00007FF7BB2E34E8   C3                  ret
```

[그림 3] GetCommandLine API를 통한 인자 확인

BiBi 와이퍼(Wiper) 악성코드는 공격 대상의 PC에서 실행되면,
가장 먼저 GetCommandLine API를 통해 인자를 확인한다.
인자는 경로가 될 수 있으며, 경로를 입력하고 BiBi 와이퍼(Wiper) 악성코드를 실행하면
대상 경로의 파일이 무작위 데이터에 의해 덮어씌워진다.

3.2 사용 API 주소의 동적 획득

```
00007FF7BB2DA34A   48:8905 EF7E0200    mov qword ptr ds:[<&CreateEventExW>],rax
00007FF7BB2DA351   FF15 E17C0100       call qword ptr ds:[<&GetProcAddress>]
00007FF7BB2DA357   48:8D15 9A920100    lea rdx,qword ptr ds:[7FF7BB2F35F8]          rdx:"CreateThreadpoolTimer", 00007FF7
00007FF7BB2DA35E   48:8BCB             mov rcx,rbx
00007FF7BB2DA361   48:8905 E07E0200    mov qword ptr ds:[<&CreateSemaphoreW>],rax
00007FF7BB2DA368   FF15 CA7C0100       call qword ptr ds:[<&GetProcAddress>]
00007FF7BB2DA36E   48:8D15 9B920100    lea rdx,qword ptr ds:[7FF7BB2F3610]          rdx:"CreateThreadpoolTimer", 00007FF7
00007FF7BB2DA375   48:8BCB             mov rcx,rbx
00007FF7BB2DA378   48:8905 D17E0200    mov qword ptr ds:[7FF7BB302250],rax
00007FF7BB2DA37F   FF15 B37C0100       call qword ptr ds:[<&GetProcAddress>]
00007FF7BB2DA385   48:8D15 9C920100    lea rdx,qword ptr ds:[7FF7BB2F3628]          rdx:"CreateThreadpoolTimer", 00007FF7
00007FF7BB2DA38C   48:8BCB             mov rcx,rbx
00007FF7BB2DA38F   48:8905 C27E0200    mov qword ptr ds:[7FF7BB302258],rax
00007FF7BB2DA396   FF15 9C7C0100       call qword ptr ds:[<&GetProcAddress>]
00007FF7BB2DA39C   48:8D15 9D920100    lea rdx,qword ptr ds:[7FF7BB2F3640]          rdx:"CreateThreadpoolTimer", 00007FF7
00007FF7BB2DA3A3   48:8BCB             mov rcx,rbx
00007FF7BB2DA3A6   48:8905 B37E0200    mov qword ptr ds:[7FF7BB302260],rax
00007FF7BB2DA3AD   FF15 857C0100       call qword ptr ds:[<&GetProcAddress>]
00007FF7BB2DA3B3   48:8D15 A6920100    lea rdx,qword ptr ds:[7FF7BB2F3660]          rdx:"CreateThreadpoolTimer", 00007FF7
00007FF7BB2DA3BA   48:8BCB             mov rcx,rbx
00007FF7BB2DA3BD   48:8905 A47E0200    mov qword ptr ds:[7FF7BB302268],rax
00007FF7BB2DA3C4   FF15 6E7C0100       call qword ptr ds:[<&GetProcAddress>]
00007FF7BB2DA3CA   48:8D15 A7920100    lea rdx,qword ptr ds:[7FF7BB2F3678]          rdx:"CreateThreadpoolTimer", 00007FF7
00007FF7BB2DA3D1   48:8BCB             mov rcx,rbx
00007FF7BB2DA3D4   48:8905 957E0200    mov qword ptr ds:[7FF7BB302270],rax
00007FF7BB2DA3DB   FF15 577C0100       call qword ptr ds:[<&GetProcAddress>]
00007FF7BB2DA3E1   48:8D15 A8920100    lea rdx,qword ptr ds:[7FF7BB2F3690]          rdx:"CreateThreadpoolTimer", 00007FF7
00007FF7BB2DA3E8   48:8BCB             mov rcx,rbx
00007FF7BB2DA3EB   48:8905 867E0200    mov qword ptr ds:[7FF7BB302278],rax
00007FF7BB2DA3F2   FF15 407C0100       call qword ptr ds:[<&GetProcAddress>]
00007FF7BB2DA3F8   48:8D15 A9920100    lea rdx,qword ptr ds:[7FF7BB2F36A8]          rdx:"CreateThreadpoolTimer", 00007FF7
00007FF7BB2DA3FF   48:8BCB             mov rcx,rbx
00007FF7BB2DA402   48:8905 777E0200    mov qword ptr ds:[7FF7BB302280],rax
00007FF7BB2DA409   FF15 297C0100       call qword ptr ds:[<&GetProcAddress>]
00007FF7BB2DA40F   48:8D15 AA920100    lea rdx,qword ptr ds:[7FF7BB2F36C0]          rdx:"CreateThreadpoolTimer", 00007FF7
00007FF7BB2DA416   48:8BCB             mov rcx,rbx
```

[그림 4] GetProcAddress API를 통한 동적 API 주소 획득

BiBi 와이퍼(Wiper) 악성코드는 GetProcAddress API를 사용하여
시스템 파괴 행위에 사용할 API의 주소를 획득하는데,
이러한 행위는 시그니처 기반 AV의 탐지를 피하기 위함이다.
본 행위는 사용할 API 목록을 숨겨 AV의 탐지를 회피하는 전형적인 모습이다.

3.3 논리 디스크 목록 획득

```
00007FF7BB2D5C1D   48:8B49 F8          mov rcx,qword ptr ds:[rcx-8]
00007FF7BB2D5C21   48:2BC1             sub rax,rcx
00007FF7BB2D5C24   48:83C0 F8          add rax,FFFFFFFFFFFFFFF8
00007FF7BB2D5C28   48:83F8 1F          cmp rax,1F
00007FF7BB2D5C2C   0F87 83040000       ja bibi.7FF7BB2D6085
00007FF7BB2D5C32   E8 454C0000         call bibi.7FF7BB2DA87C
00007FF7BB2D5C37   FF15 C3C30100       call qword ptr ds:[<&GetLogicalDrives>]
00007FF7BB2D5C3D   8945 DF             mov dword ptr ss:[rbp-21],eax
00007FF7BB2D5C40   BB 1A000000         mov ebx,1A
00007FF7BB2D5C45   83FB 02             cmp ebx,2
00007FF7BB2D5C48   0F84 68010000       je bibi.7FF7BB2D5DB6
00007FF7BB2D5C4E   48:8D45 DF          lea rax,qword ptr ss:[rbp-21]
00007FF7BB2D5C52   0FA318              bt dword ptr ds:[rax],ebx
00007FF7BB2D5C55   0F83 5B010000       jae bibi.7FF7BB2D5DB6
```

[그림 5] GetLogicalDrives API를 통한 논리 디스크 목록 획득

이후 파괴 대상 시스템의 논리 디스크 목록을 획득하는데,
이 때 사용하는 API는 GetLogicalDrives를 사용한다.
이를 통해 파괴 대상의 시스템 내 존재하는 논리 디스크 목록을 획득한다.

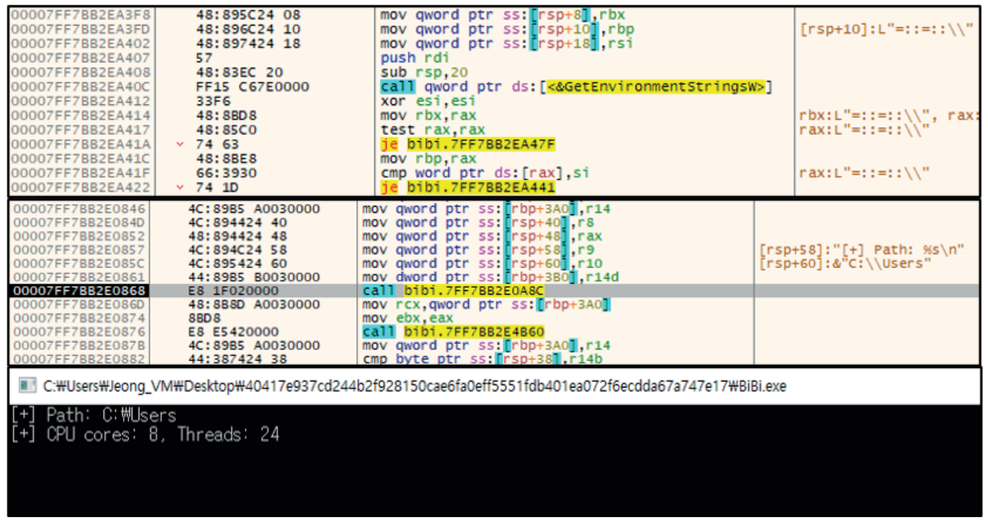
3.4 드라이브 타입 확인

```
00007FF7BB2D5C67   8D43 41             lea eax,qword ptr ds:[rbx+41]
00007FF7BB2D5C6A   4C:896D CF          mov qword ptr ss:[rbp-31],r13
00007FF7BB2D5C6E   8845 BF             mov byte ptr ss:[rbp-41],al
00007FF7BB2D5C71   C645 C0 00          mov byte ptr ss:[rbp-40],0
00007FF7BB2D5C75   41:B8 02000000      mov r8d,2
00007FF7BB2D5C7B   48:8D15 8A640200    lea rdx,qword ptr ds:[7FF7BB2FC10C]          rdx:":\\", 00007FF7BB2FC10C:":\\"
00007FF7BB2D5C82   48:8D4D BF          lea rcx,qword ptr ss:[rbp-41]
00007FF7BB2D5C86   E8 650E0000         call bibi.7FF7BB2D6AF0
00007FF7BB2D5C8B   48:8D4D BF          lea rcx,qword ptr ss:[rbp-41]
00007FF7BB2D5C8F   48:837D D7 10       cmp qword ptr ss:[rbp-29],10
00007FF7BB2D5C94   48:0F434D BF        cmovae rcx,qword ptr ss:[rbp-41]
00007FF7BB2D5C99   FF15 69C30100       call qword ptr ds:[<&GetDriveTypeA>]
00007FF7BB2D5C9F   8D48 FE             lea ecx,qword ptr ds:[rax-2]
00007FF7BB2D5CA2   F7C1 FAFFFFFF       test ecx,FFFFFFFA
```

[그림 6] GetDriveType API를 통한 논리 디스크 타입 확인

GetDriveTypeA API를 이용하여 이전에 획득한 논리 디스크들의 타입을 확인한다.
해당 행위를 통해 파괴 대상 디스크의 세부 사항을 확인하여
더욱 정밀한 디스크 내 파일들의 파괴 행위가 가능하다.

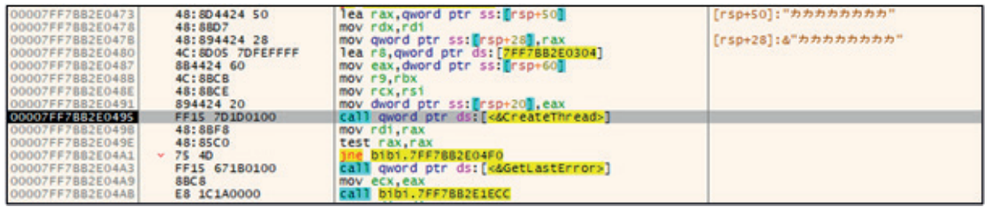
3.5 주요 경로 획득



[그림 7] GetEnvironmentStrings API를 통한 주요 경로 획득

파괴 대상이 되는 주요 경로를 획득하는데, 주요 경로는 C:\Users와 그 하위 경로가 대상이 된다.

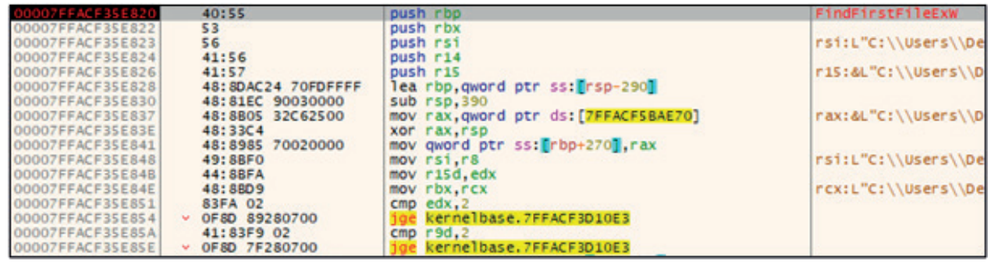
3.6 멀티스레딩을 통한 신속한 파괴 행위 수행



[그림 8] CreateThread API를 통한 멀티스레딩 수행

CreateThread API를 호출하여 여러 개의 스레드를 생성하고 실행하는데,
이는 멀티스레딩을 이용한 신속한 시스템 파괴 행위를 수행하기 위함이다.
각각의 스레드는 경로 확인과 파일 파괴 등 기능을 분담하여 신속한 시스템 파괴 행위를 수행한다.

3.7 파괴 대상 경로 탐색



[그림 9] FindFirstFIleEx API를 통한 파괴 대상 경로 탐색

FindFirstFileEx API를 이용하여 파괴할 파일을 찾아 이전에 확인했던
C:\Users 및 그 하위 경로를 탐색 시도한다.

3.8 파일 데이터 덮어쓰기 및 이름 변경

```
00007FFACF357860  48:83EC 58              sub rsp,58                                  CreateFileW
00007FFACF357864  44:8B9424 88000000      mov r10d,dword ptr ss:[rsp+88]
00007FFACF35786C  41:8BC2                 mov eax,r10d
00007FFACF35786F  25 B77F0000             and eax,7FB7
00007FFACF357874  C74424 30 20000000      mov dword ptr ss:[rsp+30],20                20:' '
00007FFACF35787C  894424 34               mov dword ptr ss:[rsp+34],eax
00007FFACF357880  41:8BC2                 mov eax,r10d

00007FFACF358580  48:895C24 10            mov qword ptr ss:[rsp+10],rbx               WriteFile
00007FFACF358585  48:897424 18            mov qword ptr ss:[rsp+18],rsi
00007FFACF35858A  4C:894C24 20            mov qword ptr ss:[rsp+20],r9
00007FFACF35858F  57                      push rdi
00007FFACF358590  48:83EC 60              sub rsp,60

00007FFACF37A300  48:83EC 38              sub rsp,38                                  MoveFileExW
00007FFACF37A304  48:836424 28 00         and qword ptr ss:[rsp+28],0                 [rsp+28]:&L"C:\\Users\\fElELSlikX.BiBi1"
00007FFACF37A30A  45:33C9                 xor r9d,r9d
00007FFACF37A30D  44:894424 20            mov dword ptr ss:[rsp+20],r8d
00007FFACF37A312  45:33C0                 xor r8d,r8d
00007FFACF37A315  E8 16000000             call <kernelbase.MoveFileWithProgressTransacted
00007FFACF37A31A  48:83C4 38              add rsp,38
00007FFACF37A31E  C3                      ret
00007FFACF37A31F  CC                      int3
```

[그림 10] 파일 데이터 덮어쓰기 및 이름 변경

경로 상의 모든 파일에 대해 임의의 데이터를 덮어씌우는 행위를 반복한다.
해당 작업을 통해 파일은 복구 자체가 불가능한 상태로 변경되며, 이름과 확장자가 변경된다.

3.9 볼륨 섀도우 카피 복사본 제거 및 주요 보안 설정 해제

```
00007FFACF3377A0  40:53                   push rbx                                    CreateProcessInternalW
00007FFACF3377A2  56                      push rsi
00007FFACF3377A3  57                      push rdi
00007FFACF3377A4  41:54                   push r12
00007FFACF3377A6  41:55                   push r13
00007FFACF3377A8  41:56                   push r14
00007FFACF3377AA  41:57                   push r15
00007FFACF3377AC  B8 D0140000             mov eax,14D0
00007FFACF3377B1  E8 3D8C0800             call <JMP.&__chkstk>
00007FFACF3377B6  48:2BE0                 sub rsp,rax
00007FFACF3377B9  48:8B05 B0362800        mov rax,qword ptr ds:[7FFACF5BAE70]
00007FFACF3377C0  48:33C4                 xor rax,rsp
00007FFACF3377C3  48:898424 C0140000      mov qword ptr ss:[rsp+14C0],rax
00007FFACF3377CB  48:89A424 F8050000      mov qword ptr ss:[rsp+5F8],rsp
00007FFACF3377D3  49:8BC1                 mov rax,r9
00007FFACF3377D6  48:898424 48030000      mov qword ptr ss:[rsp+348],rax
00007FFACF3377DE  48:8BF1                 mov rsi,rcx
00007FFACF3377E1  8B9C24 40150000         mov ebx,dword ptr ss:[rsp+1540]

1: rcx 0000000000000000
2: rdx 0000000000000000
3: r8 0000027F37A52F70 L"cmd.exe /c wmic shadowcopy delete"
4: r9 0000000000000000
5: [rsp+28] 0000000000000000

1: rcx 0000000000000000
2: rdx 0000000000000000
3: r8 0000027F37A649B0 L"cmd.exe / c bcdedit / set {default} bootstatuspolicy ignoreallfailures"
4: r9 0000000000000000
5: [rsp+28] 0000000000000000

1: rcx 0000000000000000
2: rdx 0000000000000000
3: r8 0000027F37A649B0 L"cmd.exe /c bcdedit /set {default} recoveryenabled no"
4: r9 0000000000000000
5: [rsp+28] 0000000000000000
```

[그림 11] 볼륨 섀도우 카피 복사본 제거 및 주요 보안 설정 해제

CreateProcessInternal API를 사용하여 CMD.exe 유틸을 실행해
볼륨 섀도우 복사본 삭제 행위와 주요 보안 설정을 비활성화한다.

	cmd.exe /c wmic shadowcopy delete
	볼륨 섀도우 카피 복사본 제거
	cmd.exe /c bcdedit /set {default} bootstatuspolicy ignoreallfailures
	Windows 오류 복구 알림창 표시 해제
	cmd.exe /c bcdedit /set {default} recoveryenabled no
	Windows 복구 모드 사용 해제

[표 2] BiBi 와이퍼(Wiper) 악성코드의 시스템 보안 훼손 명령

3.10 파일 공유 위반 프로세스 종료를 위한 Rstrtmgr DLL 로드

```
00007FFACF358820   40:55          push rbp                    LoadLibraryExW
00007FFACF358822   53             push rbx
00007FFACF358823   57             push rdi
00007FFACF358824   48:8BEC        mov rbp,rsp
```

[그림 12] 파일 공유 위반 프로세스 종료를 위한 Rstrtmgr.dll 로드

파일 파괴 행위 중 이미 열린 파일로서, 공유 위반이 발생하여 파괴 행위가 불가능하면
해당 파일을 열고 있는 프로세스를 종료 시키기 위해 위와 같이 Rstrtmgr.dll을 로드한다.
해당 DLL 내의 API들을 이용하여 공유 위반이 발생한 프로세스를 찾고 종료할 수 있다.

3.11 파일 공유 위반 프로세스 종료를 위한 Rstrtmgr DLL 로드

```
00007FFACAD93FC0   48:895C24 08     mov qword ptr ss:[rsp+8],rbx    RmStartSession
00007FFACAD93FC5   48:897424 10     mov qword ptr ss:[rsp+10],rsi
00007FFACAD93FCA   57               push rdi

00007FFACAD93900   48:895C24 08     mov qword ptr ss:[rsp+8],rbx    RmEndSession
00007FFACAD93905   57               push rdi
00007FFACAD93906   48:83EC 20       sub rsp,20
00007FFACAD9390A   8BF9             mov edi,ecx
00007FFACAD9390C   E8 57FEFFFF      call rstrtmgr.7FFACAD93768
00007FFACAD93911   48:85C0          test rax,rax

00007FFAD1B2C250   4C:8BD1          mov r10,rcx                     ZwOpenProcess
00007FFAD1B2C253   B8 26000000      mov eax,26                      26:'&'
00007FFAD1B2C258   F60425 0803FE7F 01  test byte ptr ds:[7FFE0308],1
00007FFAD1B2C260   75 03            jne ntdll.7FFAD1B2C265
00007FFAD1B2C262   0F05             syscall
00007FFAD1B2C264   C3               ret
00007FFAD1B2C265   CD 2E            int 2E
00007FFAD1B2C267   C3               ret

00007FFAD1B2C310   4C:8BD1          mov r10,rcx                     NtTerminateProcess
00007FFAD1B2C313   B8 2C000000      mov eax,2C                      2C:','
00007FFAD1B2C318   F60425 0803FE7F 01  test byte ptr ds:[7FFE0308],1
00007FFAD1B2C320   75 03            jne ntdll.7FFAD1B2C325
00007FFAD1B2C322   0F05             syscall
00007FFAD1B2C324   C3               ret
```

[그림 13] 파일 공유 위반 프로세스 종료

Rstrtmgr.dll 내 API들을 이용하여 파일 공유 위반이 발생한 프로세스들을 찾고
ZwOpenProcess를 호출하여 핸들을 얻은 후, ZwTerminateProcess API를 이용해 종료한다.
이를 통해 공유 위반이 발생 하여도 모든 파일을 파괴할 수 있는 기능을 수행한다.

3.12 시스템 내 파일 파괴

[그림 14] 파일 데이터 파괴 및 이름과 확장자 변경

모든 파일은 파괴되고, 파일명은 임의의 이름으로 변경된 후 확장자는 BiBi + [숫자]로 변경된다.

3.13 시스템 내 파일 파괴 (다중 파괴)

[그림 15] 파괴 행위를 수행한 라운드 횟수를 기재

위와 같이 대상 폴더 및 CPU와 Thread, 파괴된 파일 개수 등을 실시간으로 볼 수 있도록 한다.
위 사진의 라운드는 시스템이 몇 번의 파괴가 이뤄졌는지를 나타내며,
종국에는 시스템 내 모든 파일들이 다중 파괴 후 복구 불가능하도록 훼손된다.

3.14 파일 파괴 모습

[그림 16] 파괴 행위가 수행된 후 파일의 내용이 파괴된 모습

위 사진과 같이 정상적인 문자열을 갖고 있던 텍스트 문서는
BiBi 와이퍼(Wiper)의 시스템 파괴 행위 수행 후
임의의 데이터로 덮어씌워져 복구가 불가능하도록 변경되었다.

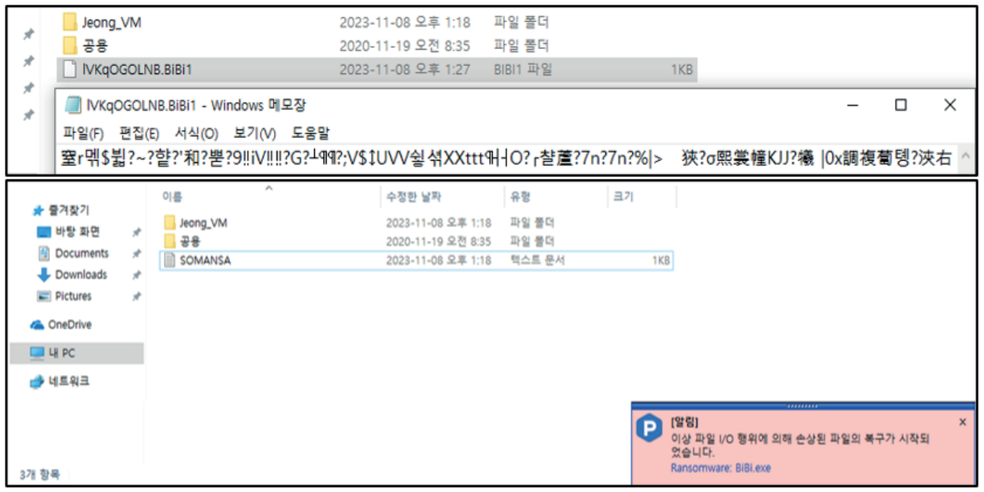
4. Privacy-i EDR 탐지 정보

4.1 이상 파일 행위 탐지 기능을 통한 악성코드 탐지/차단

[그림 17] BiBi 와이퍼(Wiper)의 이상 파일 행위 탐지 알림 화면

Privacy-i EDR은 이상 파일 행위로 탐지 기능을 이용하여
BiBi 와이퍼(Wiper) 악성코드의 시스템 파괴 행위를 탐지하고 차단하였다.

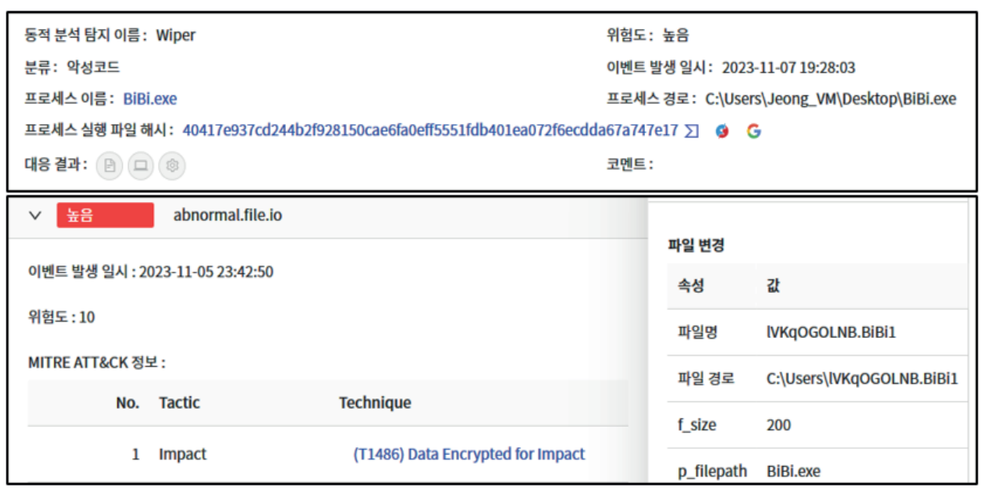
4.2 실시간 백업/복구 기능을 통한 파일 복원



[그림 18] BiBi 와이퍼가 손상시킨 파일을 실시간 복구 기능으로 복구하는 화면

Privacy-i EDR의 실시간 백업 및 복구 기능을 통해 파괴된 파일에 대해 복구를 진행하는 모습이다.

4.3 이상 파일 행위 탐지



[그림 19] BiBi 와이퍼(Wiper)의 이상 파일 행위 탐지 로그

Privacy-i EDR은 이상 파일 행위를 탐지하고 차단할 수 있으며,
이러한 방법을 통해 알려지지 않았던 BiBi 와이퍼(Wiper)에도 대응이 가능함을 확인할 수 있다.

5. 대응

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

6. 별 첨

6.1 국내 피해 현황

기업명	설명
-	식별된 피해 사례 없음

[표 3] 국내 기업 피해 현황

6.2 국제 피해 현황

기업명	설명
-	식별된 피해 사례 없음

[표 4] 국제 기업 피해 현황

궁금하신 점이나 문의사항은 malware@somansa.com 으로 문의주십시오