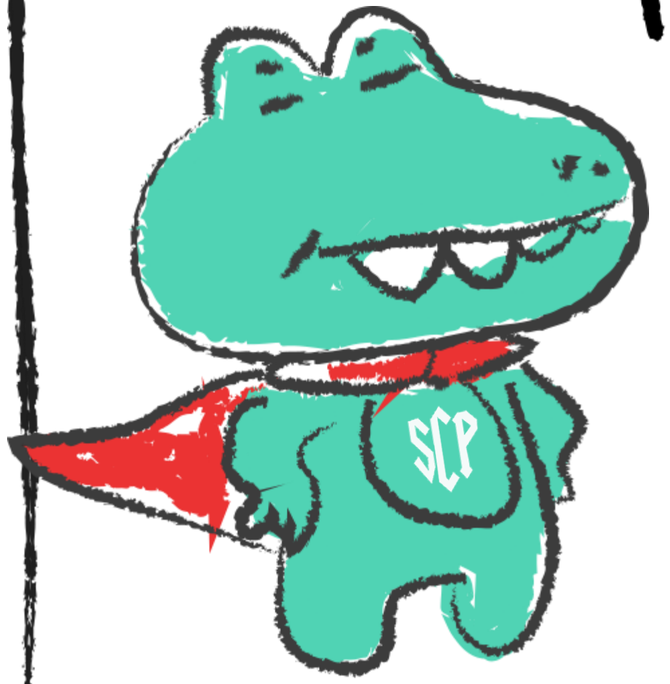
Template 분석

- Prefetch -



01.20 장혜선

목차

첫번째 Prefetch

두번째 Decompress

세번째 Template 분석

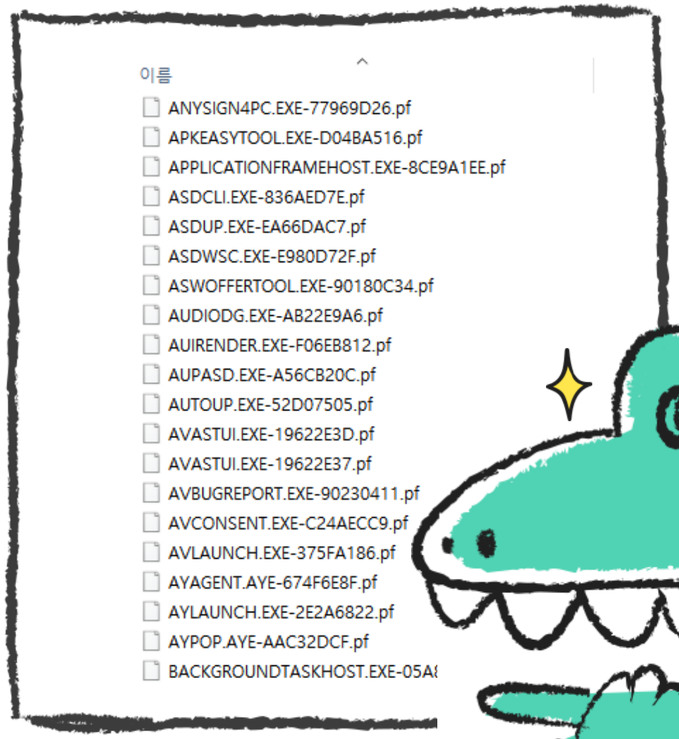
1. Prefetch

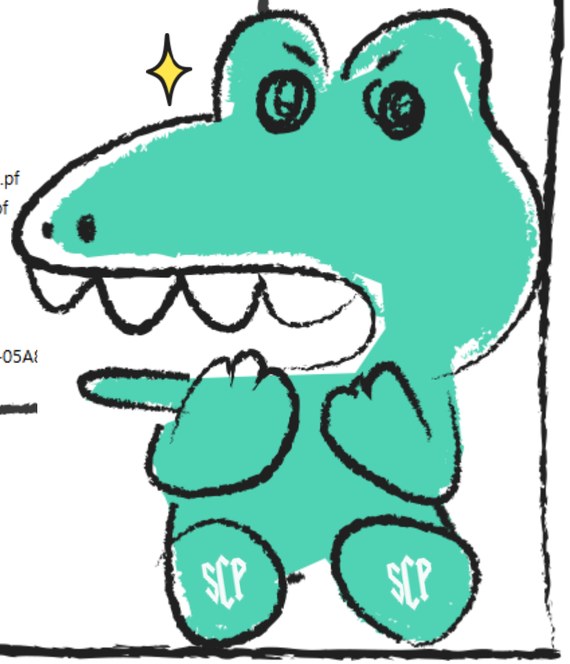
- **Window XP** 버전에서 도입
- Windows 운영체제에서 부팅/응용 실행 시 프로세스 적재 **가속화**를 위한 기능
- 같은 실행 파일이어도 다른 위치에서 처음 실행 시 프리패칭 수행

Ref. https://br4gon.tistory.com/2

1. Prefetch

- 개수 제한
  - Windows XP, Vista, 7은 **128**개
  - Windows 8 이상은 **1024**개
- **가장 오래 사용하지 않은** 응용프로그램의 프리패치 파일 삭제
- %SystemRoot%₩Prefetch
- [실행파일명]-[파일 경로에 대한 해쉬].pf





1. Prefetch – 알 수 있는 정보

실행 파일 정보
(파일명, 경로 등)

실행 정보
(시간, 횟수)

볼륨 정보

실행파일이 참조하는
파일 및 디렉토리

2. Decompress

Win 10부터는 **MAM** 형태로 압축되어 있다.

ALNOTICE.EXE-B091854C.pf

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	4D	41	4D	04	C8	BD	00	00	A5	A7	A6	AA	B9	B7	BA	BB	MAM.È½..¥§¦ª¹·º»
00000010	A9	B7	BB	BB	B9	B7	BB	DB	B9	B7	CB	CB	A9	B7	AB	B9	©·»»¹·»Û¹·ËË©·«¹
00000020	A9	A7	AA	BA	A9	C7	BB	BA	99	87	89	99	98	A7	99	99	©§ª°©Ç»°™‡‰™˜§™™
00000030	A8	97	98	99	99	A7	B8	AA	B9	B7	CB	DB	B9	B7	BC	CB	¨—˜™™§¸ª¹·ËÛ¹·¼Ë
00000040	C9	B7	BB	DB	C9	B7	BB	0B	D8	B7	DB	CB	B9	C7	DB	AB	É·»ÛÉ·».Ø·ÛË¹ÇÛ«
00000050	C9	B7	BB	CD	DA	C7	BB	BC	C9	C7	DB	DB	DA	B7	DD	CB	É·»ÍÚÇ»¼ÉÇÛÛÚ·ÝË
00000060	DA	C7	B0	DC	CA	C7	DB	DC	B9	C7	BB	BB	CB	B7	BB	CB	ÚÇ°ÜÊÇÛÜ¹Ç»»Ë·»Ë
00000070	BA	D8	B0	DB	BA	C8	CD	BA	BA	D7	0C	BB	BA	07	BB	BB	°Ø°Û°ÈÍ°°×.»°.»»
00000080	AA	D8	BB	DB	CA	C8	0B	8B	0D	00	00	00	00	00	00	00	ªØ»ÛÊÈ.‹.....
00000090	00	00	00	00	00	00	00	D0	89	0D	0B	00	00	00	00	D0	.....Ð‰.....Ð
000000A0	87	A8	B2	00	00	00	C0	B0	87	C9	C5	00	C0	00	D0	D0	‡¨²...À°‡ÉÅ.À.ÐÐ
000000B0	96	BB	A7	00	D9	00	00	D0	98	9B	99	D0	AB	C0	B0	90	–»§.Ù..Ð˜›™Ð«À°.
000000C0	88	A8	88	CC	BC	DD	9C	80	87	A9	88	BD	B0	CC	90	8D	ˆ¨ˆÌ¼Ýœ€‡©ˆ½°Ì..
000000D0	87	99	88	CC	B0	DC	90	80	87	98	98	D0	0B	CD	B0	8B	‡™ˆÌ°Ü.€‡˜˜Ð.Í°‹
000000E0	88	A8	A9	00	D0	DB	BD	80	90	A8	B9	0B	CD	0A	BD	9D	ˆ¨©.ÐÛ½€.¨¹.Í.½.
000000F0	90	AA	BB	DB	CC	0B	CC	AD	B0	DC	D0	00	0D	00	00	B0	.ª»ÛÌ.Ì.°ÜÐ....°
00000100	D0	BB	7D	0C	00	00	00	00	48	EE	33	42	D3	D3	5E	D9	Ð»}.....Hî3BÓÓ^Ù

ALNOTICE.EXE-B091854C.pf

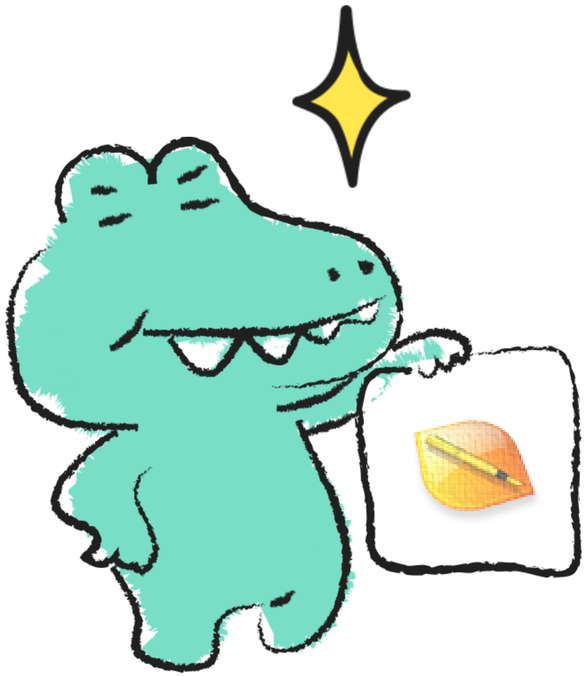
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	1E	00	00	00	53	43	43	41	11	00	00	00	68	B0	00	00	...SCCA....h°..
00000010	41	00	4C	00	4E	00	4F	00	54	00	49	00	43	00	45	00	A.L.N.O.T.I.C.E.
00000020	2E	00	45	00	58	00	45	00	00	00	00	00	00	00	00	00	..E.X.E.....
00000030	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000040	00	00	00	00	00	00	00	00	00	00	00	00	4C	85	91	B0	.....L…‘°
00000050	00	00	00	00	28	01	00	00	4F	00	00	00	08	0B	00	00	....(...O.....
00000060	71	0B	00	00	90	66	00	00	48	2E	00	00	60	95	00	00	q....f..H...`•..
00000070	01	00	00	00	08	1B	00	00	25	00	00	00	01	00	00	00	.....%.....
00000080	6F	57	8A	A1	A7	07	D8	01	C2	EC	DA	55	8E	07	D8	01	oWŠ¡§.Ø.ÂìÚUŽ.Ø.
00000090	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000A0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000B0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000C0	00	00	00	00	00	00	00	00	02	00	00	00	01	00	00	00	.....
000000D0	00	00	00	00	D8	94	00	00	84	00	00	00	00	00	00	00	....Ø”..„.....
000000E0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
000000F0	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000100	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....

Ref. https://gist.github.com/EricZimmerman/95be73f6cd04882e57e6

3. Template 분석

✓ C언어와 유사한 템플릿 전용 언어

✓ 자료형으로 범위를 잡음



0050h: 00 00 00 00 28 01 00 00 4F 00 00 00 08 0B 00 00 ....(...O.....

Template Results - PF.bt

Name	Value	Start	Size
struct Win10_FileInfomation FileInfo		54h	D8h
uint MetricsOffset	128h	54h	4h

3. Template 분석 - Prefetch

※실제 작성된 코드 순서와 상이함.

File Header

```
typedef struct {
    Version version;
    char Signature[4]; //SCCA
    byte Unknown[4];
    int FileSize;
    wchar_t FileName[30];
    int Hash <format=hex>;
    Flag flag;
} FileHeader <optimize=true>;
```

Name	Value	Start	Size
struct FileHeader header		0h	54h
enum Version version	Windows10 (30)	0h	4h
char Signature[4]	SCCA	4h	4h
byte Unknown[4]		8h	4h
int FileSize	45160	Ch	4h
wchar_t FileName[30]	ALNOTICE.EXE	10h	3Ch
int Hash	B091854Ch	4Ch	4h
enum Flag flag	Application (0)	50h	4h

Ref. https://www.sweetscape.com/010editor/repository/files/PF.bt

3. Template 분석 - Prefetch

※실제 작성된 코드 순서와 상이함.

```
typedef enum <int> {
    Windows10 = 0x1E,
    Windows8x = 0x1A,
    Windows7orVista = 0x17,
    WindowsXPor2003 = 0x11
} Version;
```

```
typedef struct {
    Version version;
    char Signature[4]; //SCCA
    byte Unknown[4];
    int FileSize;
    wchar_t FileName[30];
    int Hash <format=hex>;
    Flag flag;
} FileHeader <optimize=true>;
```

```
0000h: 1E 00 00 00 53 43 43 41 11 00 00 00 68 B0 00 00  ....SCCA....h°..
0010h: 41 00 4C 00 4E 00 4F 00 54 00 49 00 43 00 45 00  A.L.N.O.T.I.C.E.
0020h: 2E 00 45 00 58 00 45 00 00 00 00 00 00 00 00 00  ..E.X.E.........
0030h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
0040h: 00 00 00 00 00 00 00 00 00 00 00 00 4C 85 91 B0  ............L…‘°
```

Template Results - PF.bt

Name	Value
struct FileHeader header	
enum Version version	Windows10 (30)

```
typedef enum <int> {
    Boot = 0x01,
    Application =0x00
} Flag;
```

```
0050h: 00 00 00 00 28 01 00 00 4F 00 00 00 08 0B 00 00  ....(...O...
0060h: 71 0B 00 00 90 66 00 00 48 2E 00 00 60 95 00 00  q....f..H...
```

Template Results - PF.bt

Name	Value
enum Flag flag	Application (0)

3. Template 분석 - Prefetch

※실제 작성된 코드 순서와 상이함.

```
//
//                    MAIN

LittleEndian();
FileHeader header;
if(header.version == 30)
{
    Win10_FileInfomation FileInfo;
    FSeek(FileInfo.MetricsOffset);
    Win10_FileMetricsArray FileMetrics;
    FSeek(FileInfo.ChainsOffset);
    Win10_TraceChainArray TraceChain;
    FSeek(FileInfo.NameStringOffset);
    Win10_FileNameStrings FileName;
    FSeek(FileInfo.VolumesInformationOffset);
    Win10_VolumeInformation VolumeInfo;
    Trailing Data;
}
```

MAIN

- 윈도우 버전에 따라 명칭과 구조가 다름.
- Win 10, Win 8, Win 7, Win XP

Name
struct FileHeader header
struct Win10_FileInfomation FileInfo
struct Win10_FileMetricsArray FileMetrics
struct Win10_TraceChainArray TraceChain
struct Win10_FileNameStrings FileName
struct Win10_VolumeInformation VolumeInfo
struct Trailing Data

3. Template 분석 - Prefetch

※실제 작성된 코드 순서와 상이함.

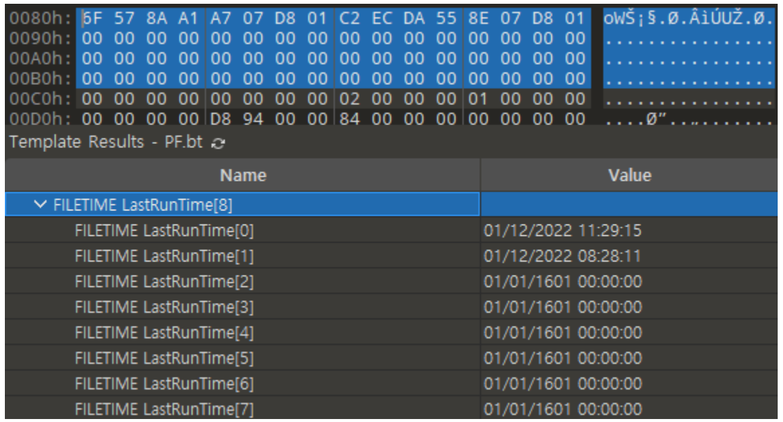
```
typedef struct {
    uint MetricsOffset <format=hex>;
    uint MetricsvolumeentryNum;
    uint ChainsOffset <format=hex>;
    uint TraceChainsvolumeentryNum;
    uint NameStringOffset <format=hex>;
    uint NameStringSize;
    uint VolumesInformationOffset <format=hex>;
    uint VolumesNum;
    uint VolumesInformarionSize;
    uint64 Unknown1;
    FILETIME LastRunTime[8];
    uint64 Unknown2;
    uint RunCount;
    uint Unknown3;
    uint Unknown4;
    uint64 Unknown5[11];
} Win10_FileInfomation;
```

File Information

Name	Value	Start	Size
struct Win10_FileInfomation FileInfo		54h	D8h
uint MetricsOffset	128h	54h	4h
uint MetricsvolumeentryNum	79	58h	4h
uint ChainsOffset	B08h	5Ch	4h
uint TraceChainsvolumeentryNum	2929	60h	4h
uint NameStringOffset	6690h	64h	4h
uint NameStringSize	11848	68h	4h
uint VolumesInformationOffset	9560h	6Ch	4h
uint VolumesNum	1	70h	4h
uint VolumesInformarionSize	6920	74h	4h
uint64 Unknown1	4294967333	78h	8h
FILETIME LastRunTime[8]		80h	40h
uint64 Unknown2	0	C0h	8h
uint RunCount	2	C8h	4h
uint Unknown3	1	CCh	4h
uint Unknown4	0	D0h	4h
uint64 Unknown5[11]		D4h	58h

※실제 작성된 코드 순서와 상이함.

3. Template 분석 - Prefetch



```
0080h: 6F 57 8A A1 A7 07 D8 01 C2 EC DA 55 8E 07 D8 01  oWŠ¡§.Ø.ÂìÚUŽ.Ø.
0090h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00A0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00B0h: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ................
00C0h: 00 00 00 00 00 00 00 00 02 00 00 00 01 00 00 00  ................
00D0h: 00 00 00 00 D8 94 00 00 84 00 00 00 00 00 00 00  ....Ø”..........
```

Template Results - PF.bt

Name	Value
FILETIME LastRunTime[8]	
FILETIME LastRunTime[0]	01/12/2022 11:29:15
FILETIME LastRunTime[1]	01/12/2022 08:28:11
FILETIME LastRunTime[2]	01/01/1601 00:00:00
FILETIME LastRunTime[3]	01/01/1601 00:00:00
FILETIME LastRunTime[4]	01/01/1601 00:00:00
FILETIME LastRunTime[5]	01/01/1601 00:00:00
FILETIME LastRunTime[6]	01/01/1601 00:00:00
FILETIME LastRunTime[7]	01/01/1601 00:00:00

File Information

- 시간 형식으로 출력
- 마지막으로 실행한 시각
- 8개

Ref.

※실제 작성된 코드 순서와 상이함.

3. Template 분석 - Prefetch

```
typedef struct {
    local int num1;
    for (num1 =0; num1 < FileInfo.MetricsvolumeentryNum; num1++)
        Win10_Entry_1 array;
} Win10_FileMetricsArray;
```

File metrics array

Name	Value
struct Win10_FileInfomation FileInfo	
uint MetricsOffset	128h
uint MetricsvolumeentryNum	79

✓ metricsvolume entry num

Name
struct Win10_Entry_1 array[74]
struct Win10_Entry_1 array[75]
struct Win10_Entry_1 array[76]
struct Win10_Entry_1 array[77]
struct Win10_Entry_1 array[78]

✓ metrics array entry

※실제 작성된 코드 순서와 상이함.

3. Template 분석 - Prefetch

```
typedef struct{
    byte FileReferenceMFTEntryIndex[6];
    byte SequenceNumber[2];
} FileReference;
```

```
typedef struct {
    uint PrefetchStartTime;
    uint PrefetchDuration;
    uint AveragePrefetchDuration;
    uint FileNameStringOffset;
    uint FileNameStringNumber;
    uint Flags;
    FileReference Data;
} Win10_Entry_1;
```

```
typedef struct {
    local int num1;
    for (num1 =0; num1 < FileInfo.MetricsvolumeentryNum; num1++)
        Win10_Entry_1 array;
} Win10_FileMetricsArray;
```

File metrics array

Name	Value
struct FileReference Data	
byte FileReferenceMFTEntryIndex[6]	↑F↓
byte SequenceNumber[2]	┐

Name	Value
struct Win10_Entry_1 array[0]	
uint PrefetchStartTime	0
uint PrefetchDuration	5
uint AveragePrefetchDuration	5
uint FileNameStringOffset	0
uint FileNameStringNumber	61
uint Flags	256
struct FileReference Data	

※실제 작성된 코드 순서와 상이함.

3. Template 분석 - Prefetch

```
typedef struct {
    uint TotalBlockLoadCount;
    uint UnKnown;
} Win10_Entry_2;
```

```
typedef struct {
    local int num2;
    for (num2 =0; num2 < FileInfo.TraceChainsvolumeentryNum; num2++)
        Win10_Entry_2 array;
} Win10_TraceChainArray;
```

Trace Chain Array

Name	Value
uint ChainsOffset	B08h
uint TraceChainsvolumeentryNum	2929

✓ Trace Chains volume entry Num

Name	Value
> struct Win10_Entry_2 array[2924]	
> struct Win10_Entry_2 array[2925]	
> struct Win10_Entry_2 array[2926]	
> struct Win10_Entry_2 array[2927]	
> struct Win10_Entry_2 array[2928]	

✓ Trace Chains array entry

3. Template 분석 - Prefetch

※실제 작성된 코드 순서와 상이함.

```
typedef struct {
    uint TotalBlockLoadCount;
    uint UnKnown;
} Win10_Entry_2;
```

```
typedef struct {
    local int num2;
    for (num2 =0; num2 < FileInfo.TraceChainsvolumeentryNum; num2++)
        Win10_Entry_2 array;
} Win10_TraceChainArray;
```

Trace Chain Array

Name	Value
struct Win10_TraceChainArray TraceChain	
struct Win10_Entry_2 array[0]	
uint TotalBlockLoadCount	2830
uint UnKnown	4294950914

※실제 작성된 코드 순서와 상이함.

3. Template 분석 - Prefetch

```
typedef struct {
    while (FTell() - FileInfo.NameStringOffset < FileInfo.NameStringSize)
        Entry_3 array;
} Win8_FileNameStrings;
```

Filename String

Name	Value
struct Win10_FileInfomation FileInfo	
uint MetricsOffset	128h
uint MetricsvolumeentryNum	79
uint ChainsOffset	B08h
uint TraceChainsvolumeentryNum	2929
uint NameStringOffset	6690h → 26256
uint NameStringSize	11848

✓ Name String Offset

Name	Value
struct Win10_FileInfomation FileInfo	
uint MetricsOffset	128h
uint MetricsvolumeentryNum	79
uint ChainsOffset	B08h
uint TraceChainsvolumeentryNum	2929
uint NameStringOffset	6690h
uint NameStringSize	11848

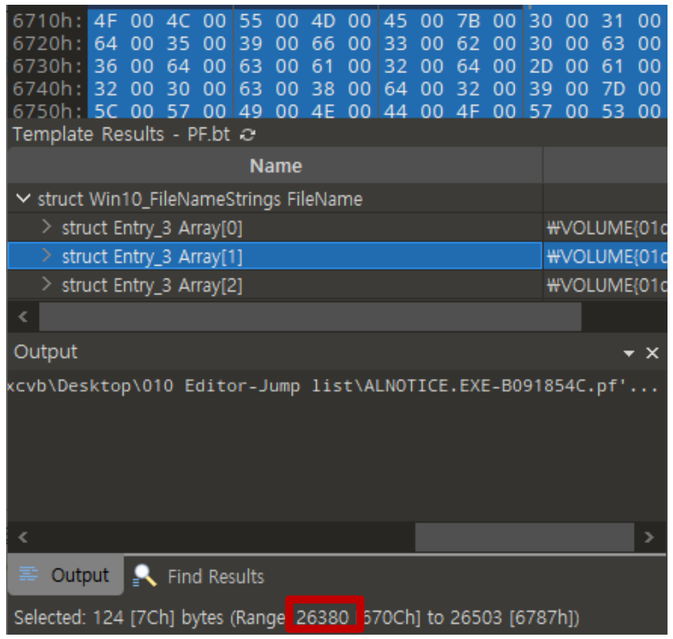
✓ Name String Size

※실제 작성된 코드 순서와 상이함.

3. Template 분석 - Prefetch

```
typedef struct {
    while (FTell() - FileInfo.NameStringOffset < FileInfo.NameStringSize)
        Entry_3 array;
} Win8_FileNameStrings;
```

Filename String



- 현 위치 : 26380
- 오프셋 : 26256
- 크기 : 11848

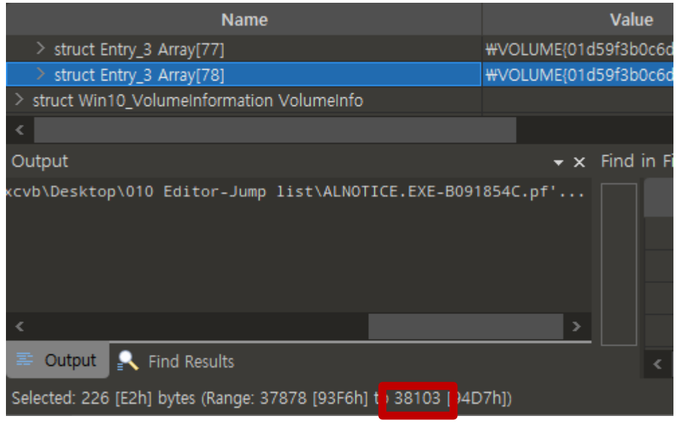
$$\underbrace{26{,}380 - 26{,}256}_{124} < 11{,}848$$

3. Template 분석 - Prefetch

※실제 작성된 코드 순서와 상이함.

```
typedef struct {
    while (FTell() - FileInfo.NameStringOffset < FileInfo.NameStringSize)
        Entry_3 array;
} Win8_FileNameStrings;
```

Filename String



- 현 위치 : 38104
- 오프셋 : 26256
- 크기 : 11848

$$\underbrace{38{,}104 - 26{,}256}_{11848} < 11{,}848$$

3. Template 분석 - Prefetch

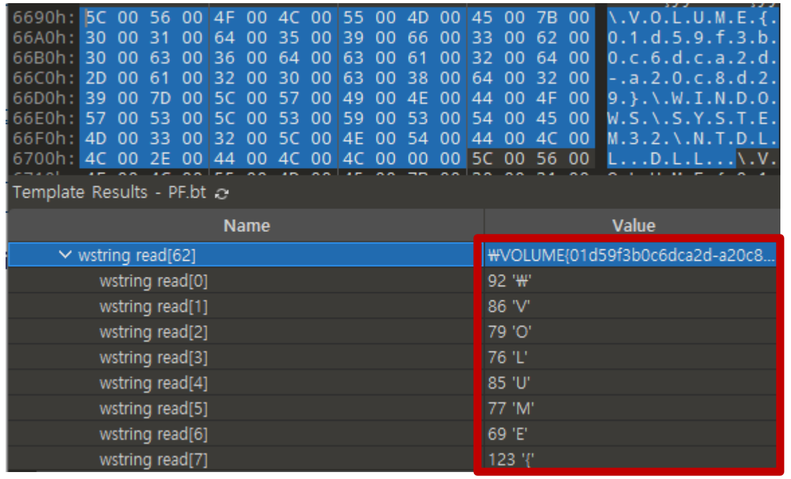
※실제 작성된 코드 순서와 상이함.

```
wstring ReadWstring(Entry_3 &read)
{
    return read.read;
}
```

```
typedef struct {
    wstring read;
} Entry_3 <read=ReadWstring>;
```

```
typedef struct {
    while (FTell() - FileInfo.NameStringOffset < FileInfo.NameStringSize)
        Entry_3 array;
} Win8_FileNameStrings;
```

Filename String



※실제 작성된 코드 순서와 상이함.

3. Template 분석 - Prefetch

Volume Information

```
typedef struct {
    Win10_VolumeInformationEntry VolumeEntry;
    Win10_VolumeInformationData VolumeData;
} Win10_VolumeInformation;
```

Name
struct Win10_VolumeInformation VolumeInfo
struct Win10_VolumeInformationEntry VolumeEntry
struct Win10_VolumeInformationData VolumeData

※실제 작성된 코드 순서와 상이함.

3. Template 분석 - Prefetch

```
typedef struct {
    local int64 now = FTell();
    uint VolumeDevicePathOffset <format=hex>;
    uint NumberVolumeDevicePath;
    FILETIME VolumeCreateTime;
    uint VolumeSerialNumber <format=hex>;
    uint FileReferencesOffset <format=hex>;
    uint FileReferencesDataSize;
    uint DirectoryStringsOffset <format=hex>;
    uint NumberDirectoryStrings;
    uint Unknown1;
    uint64 UnKnown2[3];
    uint CopyNumberDirectoryStrings;
    uint64 UnKnown3[3];
    uint Unknown;
} Win10_VolumeInformationEntry;
```

Volume Information Entry

Name	Value
struct Win10_VolumeInformation VolumeInfo	
struct Win10_VolumeInformationEntry VolumeEntry	
uint VolumeDevicePathOffset	60h
uint NumberVolumeDevicePath	34
FILETIME VolumeCreateTime	11/20/2019 00:40:03
uint VolumeSerialNumber	A20C8D29h
uint FileReferencesOffset	A8h
uint FileReferencesDataSize	1040
uint DirectoryStringsOffset	4B8h
uint NumberDirectoryStrings	37
uint Unknown1	91
uint64 UnKnown2[3]	
uint CopyNumberDirectoryStrings	37
uint64 UnKnown3[3]	
uint Unknown	6684729

※실제 작성된 코드 순서와 상이함.

3. Template 분석 - Prefetch

Volume Information Data

```
typedef struct {
    FSeek(VolumeInfo.VolumeEntry.now + VolumeInfo.VolumeEntry.VolumeDevicePathOffset);
    wstring VolumeDevicePath;
    FSeek(VolumeInfo.VolumeEntry.now+ VolumeInfo.VolumeEntry.FileReferencesOffset);
    Win10_FileReferences FileReferences;
    FSeek(VolumeInfo.VolumeEntry.now + VolumeInfo.VolumeEntry.DirectoryStringsOffset);
    DirectoryStrings DirStrings;
} Win10_VolumeInformationData;
```

Name	Value
struct Win10_VolumeInformation VolumeInfo	
struct Win10_VolumeInformationEntry VolumeEntry	
uint VolumeDevicePathOffset	96
uint NumberVolumeDevicePath	34
FILETIME VolumeCreateTime	11/20/2019 00:40:03
uint VolumeSerialNumber	2718731561
uint FileReferencesOffset	168
uint FileReferencesDataSize	1040
uint DirectoryStringsOffset	1208
uint NumberDirectoryStrings	37
uint Unknown1	91
uint64 UnKnown2[3]	
uint CopyNumberDirectoryStrings	37
uint64 UnKnown3[3]	
uint Unknown	6684729

※실제 작성된 코드 순서와 상이함.

3. Template 분석 - Prefetch

Volume Information Data

```
typedef struct {
    FSeek(VolumeInfo.VolumeEntry.now + VolumeInfo.VolumeEntry.VolumeDevicePathOffset);
    wstring VolumeDevicePath;
    FSeek(VolumeInfo.VolumeEntry.now+ VolumeInfo.VolumeEntry.FileReferencesOffset);
    Win10_FileReferences FileReferences;
    FSeek(VolumeInfo.VolumeEntry.now + VolumeInfo.VolumeEntry.DirectoryStringsOffset);
    DirectoryStrings DirStrings;
} Win10_VolumeInformationData;
```

Name	Value
struct Win10_VolumeInformation VolumeInfo	
struct Win10_VolumeInformationEntry VolumeEntry	
uint VolumeDevicePathOffset	96
uint NumberVolumeDevicePath	34
FILETIME VolumeCreateTime	11/20/2019 00:40:03
uint VolumeSerialNumber	2718731561
uint FileReferencesOffset	168
uint FileReferencesDataSize	1040
uint DirectoryStringsOffset	1208
uint NumberDirectoryStrings	37
uint Unknown1	91
uint64 UnKnown2[3]	
uint CopyNumberDirectoryStrings	37
uint64 UnKnown3[3]	
uint Unknown	6684729

※실제 작성된 코드 순서와 상이함.

3. Template 분석 - Prefetch

```
typedef struct {
    FSeek(VolumeInfo.VolumeEntry.now + VolumeInfo.VolumeEntry.VolumeDevicePathOffset);
    wstring VolumeDevicePath;
    FSeek(VolumeInfo.VolumeEntry.now+ VolumeInfo.VolumeEntry.FileReferencesOffset);
    Win10_FileReferences FileReferences;
    FSeek(VolumeInfo.VolumeEntry.now + VolumeInfo.VolumeEntry.DirectoryStringsOffset);
    DirectoryStrings DirStrings;
} Win10_VolumeInformationData;
```

```
typedef struct {
    uint UnKnown;
    uint NumberOfFileReference;
    uint64 Unknown;
    local int num = VolumeInfo.VolumeEntry.FileReferencesDataSize - 16;
    byte ArrayOfFileReference[num];
} Win10_FileReferences;
```

Volume Information Data

- FileReferences

Name	Value
uint FileReferencesDataSize	1040

Name	Value
struct Win10_VolumeInformationData VolumeData	
> wstring VolumeDevicePath[35]	₩VOLUME{01d59f3b0c6dca2d-a20c8...
v struct Win10_FileReferences FileReferences	
uint UnKnown	3
uint NumberOfFileReference	128
uint64 Unknown	15199975164936280
> byte ArrayOfFileReference[1024]	

-16

3. Template 분석 - Prefetch

※실제 작성된 코드 순서와 상이함.

```
typedef struct {
    FSeek(VolumeInfo.VolumeEntry.now + VolumeInfo.VolumeEntry.VolumeDevicePathOffset);
    wstring VolumeDevicePath;
    FSeek(VolumeInfo.VolumeEntry.now+ VolumeInfo.VolumeEntry.FileReferencesOffset);
    Win10_FileReferences FileReferences;
    FSeek(VolumeInfo.VolumeEntry.now + VolumeInfo.VolumeEntry.DirectoryStringsOffset);
    DirectoryStrings DirStrings;
} Win10_VolumeInformationData;
```

```
typedef struct {
    uint16 StringNumberOfCharacters <format=hex>;
    wstring DirectoryString;
} DirectoryStrings;
```

Volume Information Data

- DirectoryString

struct DirectoryStrings DirStrings	
uint16 StringNumberOfCharacters	48
wstring DirectoryString[49]	₩VOLUME{01d59f3b0c6dca2d-a20c8...

※실제 작성된 코드 순서와 상이함.

3. Template 분석 - Prefetch

Trailing Data

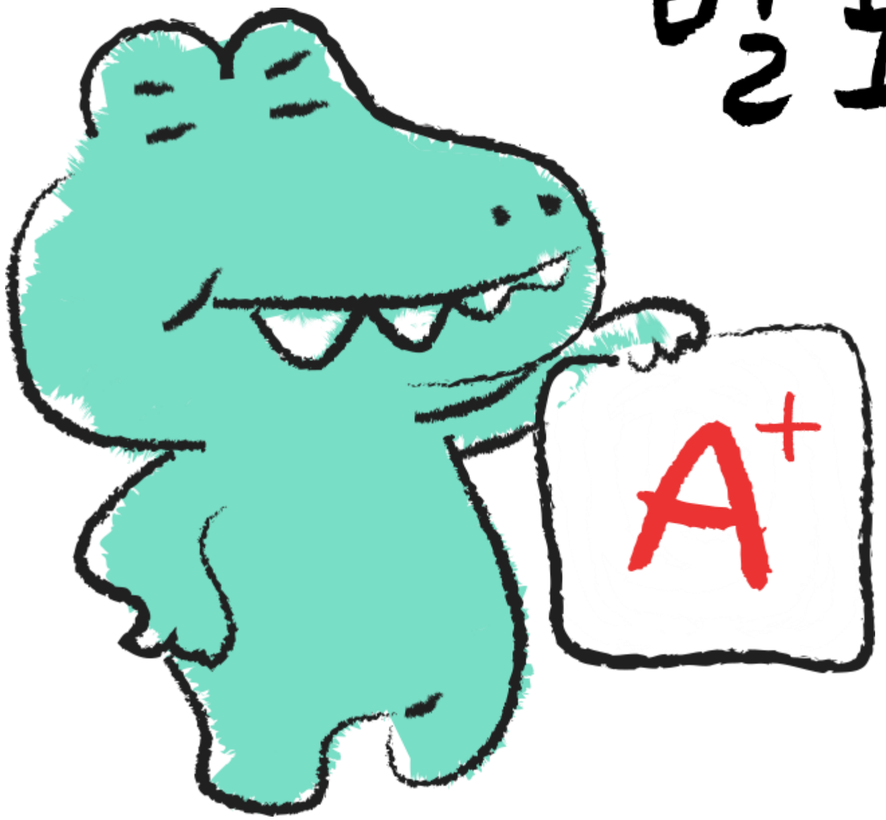
```
typedef struct {
    local int num;
    for (num = 0; num < VolumeInfo.VolumeEntry.NumberDirectoryStrings; num++)
        TrailingData Data;
} Trailing;

typedef struct {
```

```
typedef struct {
    uint16 StringNumberOfCharacters <format=hex>;
    wstring DirectoryString;
} TrailingData;
```

```
typedef struct {
    uint16 StringNumberOfCharacters <format=hex>;
    wstring DirectoryString;
} DirectoryStrings;
```

Name	Value
struct Trailing Data	
struct TrailingData Data[0]	
uint16 StringNumberOfCharacters	54
wstring DirectoryString[55]	₩VOLUME{01d59f3b0c6dca2d-a20c8...



발표를 들어주셔서
감사합니다

Ref.