

EventLog

made by 임승연

INDEX

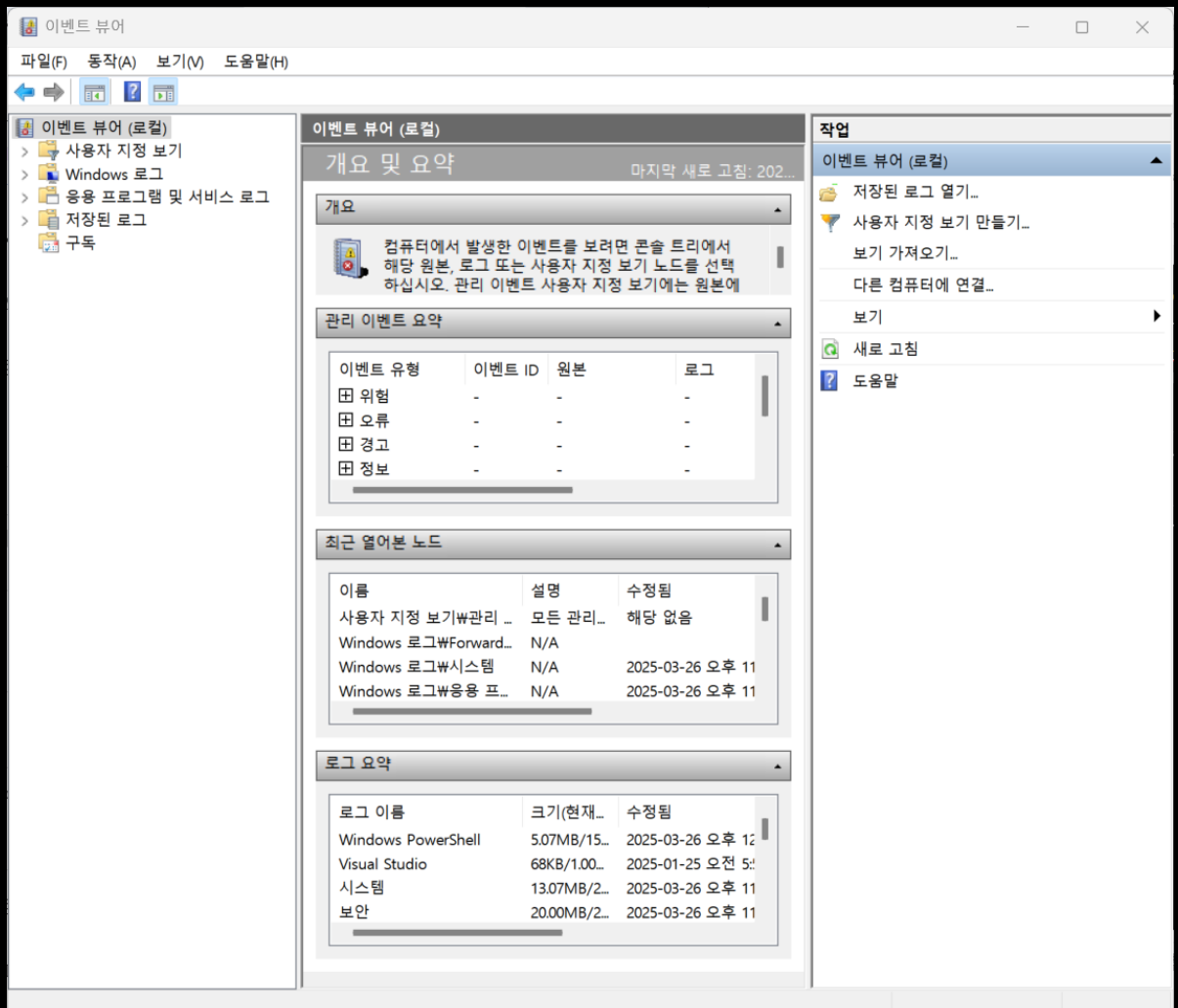
1. 개요

2. 구조와 분석 방법

1 개요

EventViewer

- 애플리케이션, 보안, 설정 및 시스템 이벤트를 포함한 필수 Windows 로그 이벤트의 범주화된 목록 제공
- 설치된 개별 애플리케이션 및 특정 Windows 구성 요소 범주에 대한 로그 그룹화 제공
- 이벤트 발생 시, 이벤트 원본에 대한 세부 정보와 이벤트 문제 해결에 도움이 될 수 있는 자세한 기술 정보 제공
- 구독을 사용하여, 여러 컴퓨터의 로그를 중앙 집중식 서버로 통합 가능
- 지정된 유형의 이벤트 발생 시, 특정 작업을 실행하도록 이벤트 뷰어 구성



1 개요

EventViewer

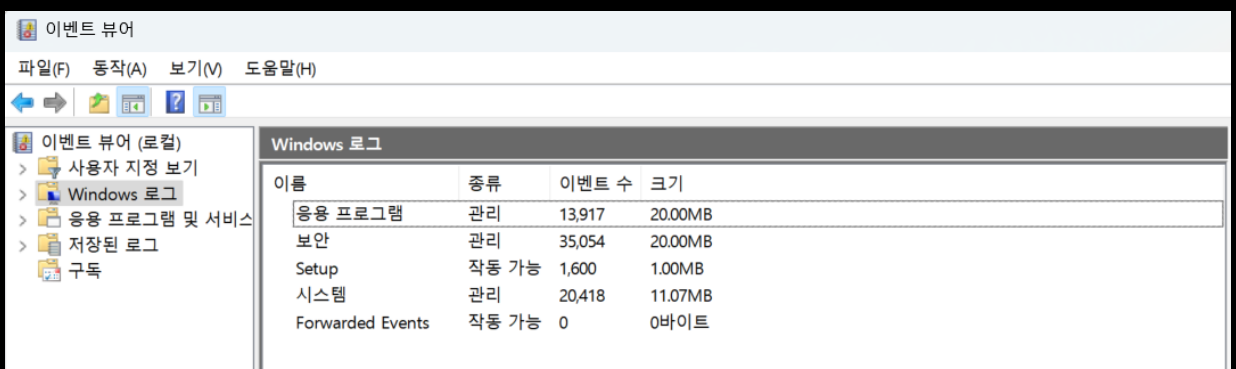
- 제공 정보
 - 이벤트에 대한 설명
 - 이벤트 ID 번호
 - 이벤트를 생성한 구성 요소 또는 하위 시스템
 - 정보, 경고 또는 오류 상태
 - 이벤트의 시간
 - 이벤트가 발생한 사용자의 이름
 - 이벤트가 발생한 컴퓨터의 이름
 - 이벤트 유형에 대한 자세한 내용이 포함된 Microsoft 지원 또는 Microsoft 기술 자료에 대한 링크

열 이름	설명
공급자 이름	이벤트 소스
이벤트 이름	해당 공급자가 지정한 이벤트
텍스트	이벤트의 공급자, 이벤트 이름 및 ID에 대한 설명
타임스탬프 (ms)	이벤트가 발생한 시간
공급자 GUID	이벤트 공급자의 ID
이벤트 ID	이벤트의 ID
프로세스 ID	이벤트가 발생한 프로세스
프로세스 이름	실행 중인 프로세스의 이름
스레드 ID	이벤트가 발생한 스레드의 ID

1 개요

EventLog

- Windows 운영 체제 내에서 발생하는 시스템 이벤트에 대한 정보를 제공
- 포함 요소
 - Windows 구성 요소
 - 설치된 애플리케이션에 대한 정보
 - 경고 및 오류 메시지
- 이벤트 수집 방법
 - Windows 방화벽에서 Windows 이벤트 로그 관리를 허용하는 인바운드 규칙 제작
- 각 로그의 속성
 - 로그 파일의 위치
 - 로그 파일의 최대 크기
 - 자동 백업 옵션
 - 로그에 대한 권한
 - 로그가 꽉 찼을 때 발생하는 동작



Windows 로그			
이름	종류	이벤트 수	크기
응용 프로그램	관리	13,917	20.00MB
보안	관리	35,054	20.00MB
Setup	작동 가능	1,600	1.00MB
시스템	관리	20,418	11.07MB
Forwarded Events	작동 가능	0	0바이트

1 개요

Eventlog

기본 제공 로그	설명 및 사용
응용 프로그램 로그 (Application)	애플리케이션에서 기록한 이벤트 로그 포함 SMTP (Simple Mail Trasfer Protocol)
보안 로그 (Security)	사용하도록 설정한 경우, 감사 결과를 보고 감사 이벤트는 이벤트에 따라 성공 또는 실패를 보고. 예를 들어, 사용자가 파일에 액세스할 수 있는지 여부에 따라 성공 또는 실패를 보고.
설정 로그 (Setup)	애플리케이션 설정과 관련된 이벤트가 포함
시스템 로그 (System)	Windows 구성 요소 및 서비스는 일반 이벤트를 기록하고 오류, 경고 또는 정보로 분류 Windows 운영 체제는 시스템 구성 요소가 기록하는 이벤트를 미리 결정
전달된 이벤트 (Forwarded Events)	Windows 구성 요소가 원격 컴퓨터에서 수집하는 이벤트 저장. 원격 컴퓨터에서 이벤트를 수집하기 위해서는 이벤트 구독을 생성해야 함
사용자 지정 로그 (CustomLog)	사용자 지정 로그를 만드는 애플리케이션에서 기록한 이벤트를 포함. 사용자 지정 로그를 사용하면 애플리케이션이 다른 애플리케이션에 영향을 주지 않고 보안 목적으로 로그의 크기를 제어하거나 ACL을 연결 가능.

1 개요

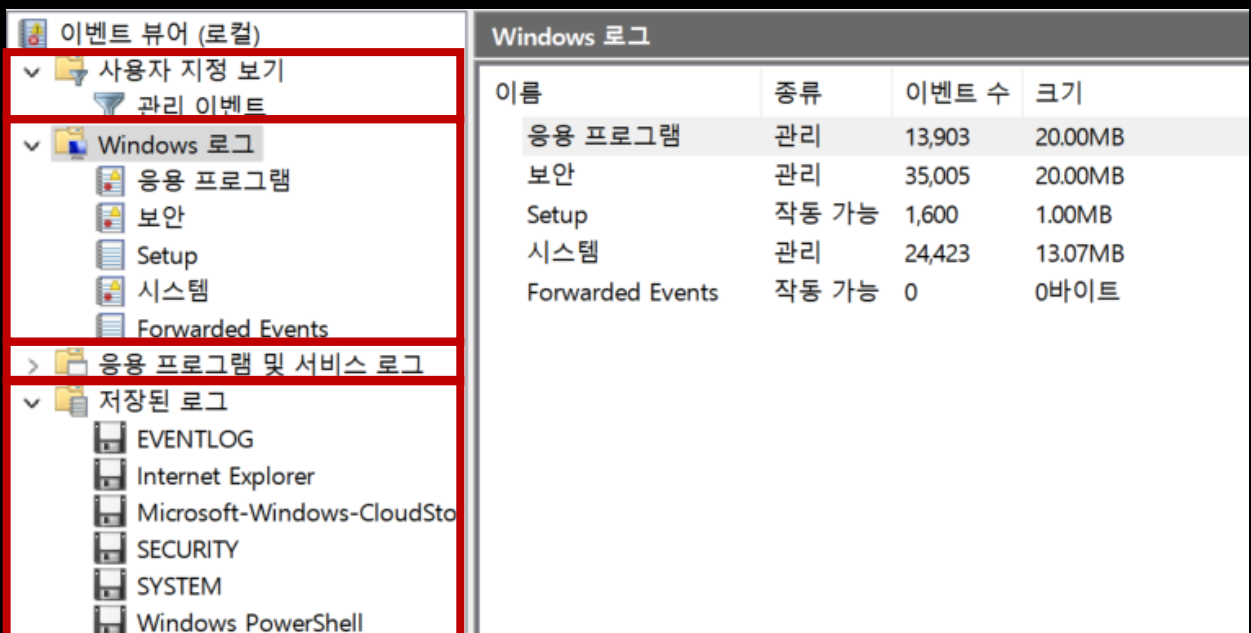
표준 로그

- Application
- Security
- System

사용자 지정 로그

- 사용자는 자신의 사용자 지정 로그 규칙의 트리거 기준을 설정 가능
- 이벤트 ID 입력 후 이벤트 소스 선택
- 표준 로그인 Application, Security, System 중 이벤트 소스를 선택 가능
- 타사 애플리케이션 로그를 지정할 수도 있음

< EventViewer >



The screenshot shows the Windows Event Viewer interface. The left-hand pane displays a tree view of event logs. The right-hand pane shows a list of events from the 'Windows 로그' (Windows Logs) category.

Left-hand Tree View:

- 이벤트 뷰어 (로컬)
 - 사용자 지정 보기
 - 관리 이벤트
 - Windows 로그
 - 응용 프로그램
 - 보안
 - Setup
 - 시스템
 - Forwarded Events
 - 응용 프로그램 및 서비스 로그
 - 저장된 로그
 - EVENTLOG
 - Internet Explorer
 - Microsoft-Windows-CloudSto
 - SECURITY
 - SYSTEM
 - Windows PowerShell

Right-hand List (Windows 로그):

이름	종류	이벤트 수	크기
응용 프로그램	관리	13,903	20.00MB
보안	관리	35,005	20.00MB
Setup	작동 가능	1,600	1.00MB
시스템	관리	24,423	13.07MB
Forwarded Events	작동 가능	0	0바이트

1 개요

응용프로그램 및 서비스 로그

- 시스템 전체에 영향을 미칠 수 있는 이벤트가 아닌 단일 애플리케이션이나 구성 요소의 이벤트를 저장
- 범주
 - 관리자
 - 작동
 - 분석 (Analytic)
 - 디버그

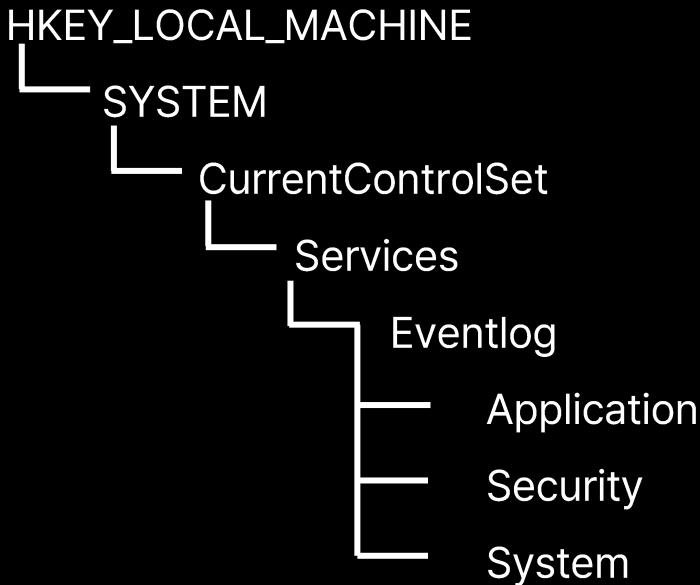
로그의 하위 유형	설명
관리자	이벤트 뷰어를 사용하여 문제를 해결하는 관리자와 지원 담당자에게 중요. 문제에 대응하는 방법에 대한 지침을 제공. 관리자 로그의 이벤트는 관리자가 작업할 수 있는 문제 및 정의된 솔루션을 표시
작동	운영 이벤트를 사용하여 문제 또는 발생을 분석 및 진단하고, 문제 또는 발생에 따라 도구나 작업을 트리거 가능
분석 및 디버그	문제를 추적하고 이벤트를 저장하고, 많은 양의 이벤트를 기록하는 경우가 많음. 개발자는 애플리케이션을 디버깅할 때, 디버그 로그 사용 분석 및 디버그 로그는 기본적으로 숨겨져 있어, 사용하지 않도록 설정됨

1 개요

이벤트 로깅 서비스

- Eventlog 레지스트리 키에 저장된 정보 사용
- Eventlog 키에는 로그라는 여러 하위 키가 포함됨
- 각 로그에는 애플리케이션이 이벤트 로그에 쓰고 읽을 때, 이벤트 로깅 서비스에서 리소스를 찾는 데 사용하는 정보가 포함됨

Eventlog 키의 구조



도메인 컨트롤러

- 도메인 사용자 계정 인증 시 계정 로그인 이벤트 생성
- 디렉토리 서비스 이벤트 기록

2 구조와 분석 방법

이벤트 로그 파일

- EVT와 EVTX의 차이

EVT	<ul style="list-style-type: none">System, Security, Application 로그에 모든 정보를 저장이벤트들이 저장되는 레코드(Record)와 헤더(Header)로만 이루어짐생성시간(Create Time)과 쓰기시간(Write Time)을 저장
EVTX	<ul style="list-style-type: none">System, Security, Application 외에도 다양한 로그로 관리하여 세분화EVT와 다르게 Chunk 개념이 도입되어<ul style="list-style-type: none">레코드들의 저장 포맷이 변경저장 시간도 이벤트 생성시간만 저장

이벤트 로그 파일의 종류

- 이벤트 로그 파일의 경로

File Extension	Path
EVT	C:/Windows/System32/config
EVTX	C:/Windows/System32/winevt/logs

2 구조와 분석 방법

이벤트 로그

- Windows에 설치되거나 새로운 프로그램이 설치될 경우, 그에 해당하는 이벤트 로그가 레지스트리 키에 등록된 후, 해당 이벤트가 발생할 경우 생성 자주 생성되는 Windows 이벤트 로그 (EVTX)
- Application.evtx : 전체적인 응용 프로그램들의 상태를 나타냄
- Security.evtx : 계정이나 권한, 시작, 이벤트 로그 상태 여부를 나타냄
- Setup.evtx : 윈도우 내의 보안 업데이트나 패키지 등의 설치 여부를 나타냄
- System.evtx : 시스템의 상태 여부를 나타냄

Log Name	Contents
Connection	Connect data Events
Error	Error Events
Hardware Events	Hardware error Events
Internet Explorer	Internet error Events
Microsoft-Windows-Bits-Clients%4Operaional	BITS (Background Intelligent Transfer Service) Events
Microsoft-Windows-DeviceSetupManager%4Operaional	DeviceSetupManager Events
Microsoft-Windows-PrintService%4Admin	Print Events
Microsoft-Windows-SmbClient%4Connectivity	SMBClient Events
Microsoft-Windows-User Profile Service%4Operational	User Profile Events
Microsoft-Windows-Windows Defender%4Operational	Windows Defender Events
Alerts	Microsoft Office Alerts Events
Microsoft-Windows-WindowsUpdateClient%4Operational	WindowsUpdate Events

2 구조와 분석 방법

EVTX 파일 형식

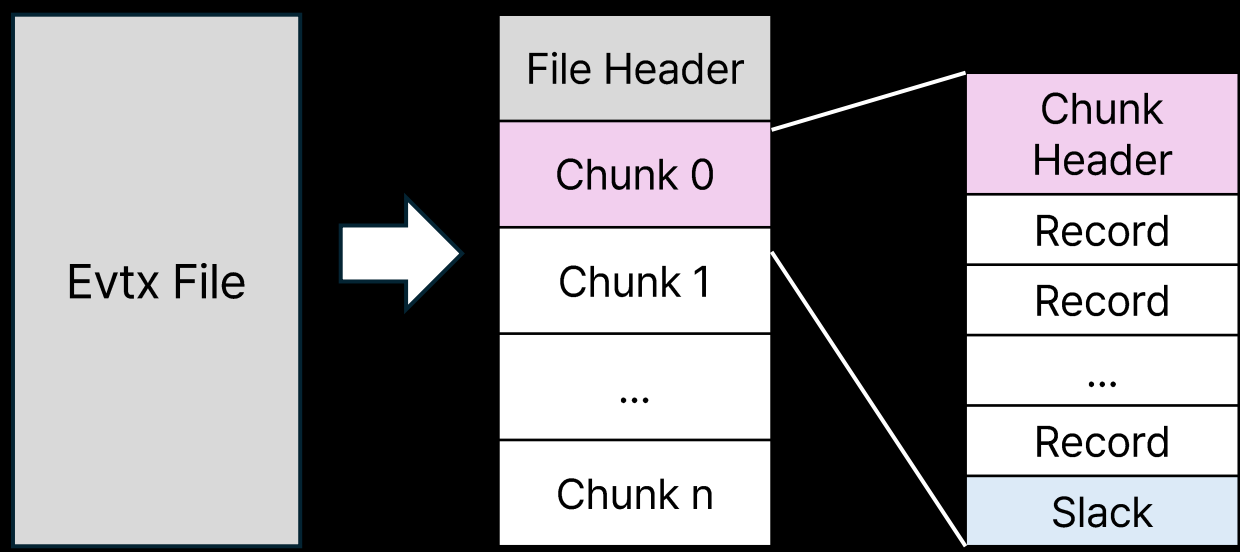
- %SystemRoot%/System32/winevt/Logs/
- 구성 요소
 - 로그 이름 (Channel) : 이벤트 로그 파일 이름
 - 원본 (Provider) : 서비스 공급자 이름
 - 작업 범주 (Task) : 원본에 의한 이벤트 분류
 - 수준 (Level) : 발생한 이벤트의 심각성 정도
 - 위험, 오류, 경고, 정보, 자세한 정보 표시
 - 로그된 날짜 (TimeCreated) : 로그가 기록된 날짜
 - 키워드 (Keywords) : 이벤트 분류
 - 사용자 (Version) : System 또는 Customer
 - 컴퓨터 (Computer) : 로그가 기록된 PC 이름
 - Opcode : 구성 요소가 수행하는 작업 범주 식별
- EVTX 파일 내 표현 방식

Character	Description
Byte Method	Little-Endian
Time Method	FileTime in UTC(UTC+9)
letter Method	ASCII string (extended ASCII) Unicode String. UTF-16 little-Endian Byte Order Mark(BOM)

2 구조와 분석 방법

EVTX 파일 구조

- Windows에서 특정 이벤트가 발생 시, 생성되는 정보를 Event Record에 저장하며, 여러 Event Record가 모여서 Chunk를 구성
- EVTX 파일 구성
 - File Header
 - 1개 이상의 Chunk
 - Chunk Header
 - 다수의 Record
 - Slack 영역



EVTX 구성 요소별 크기

Name	Size	Description
File Header	4,096 Byte	Fixed
Chunk	65, 536 Byte	Fixed
Chunk Header	512 Byte	Fixed
Record	-	Viable
Slack	-	Viable

2 구조와 분석 방법

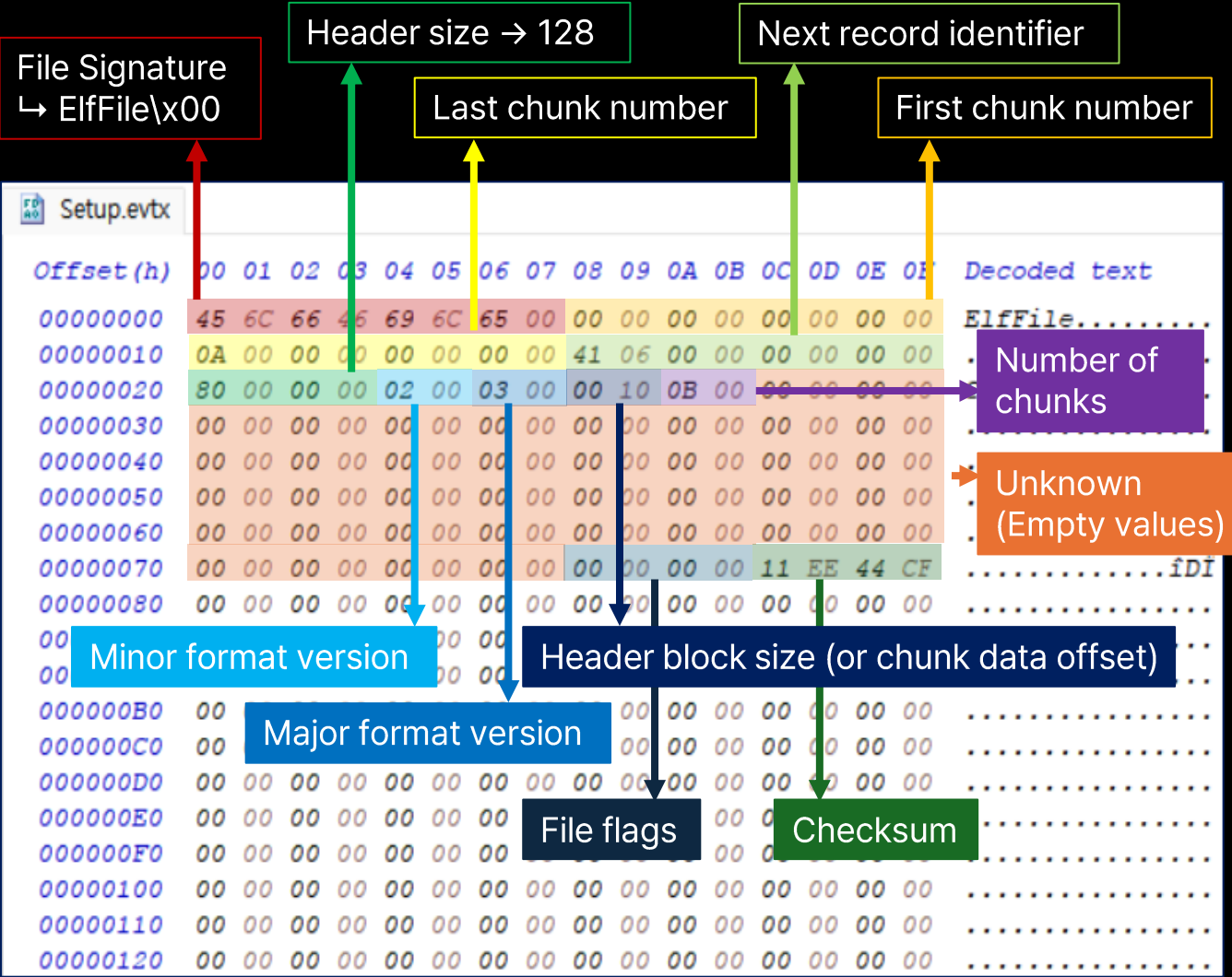
EVTX 파일 Header 구조

- EVTX 파일 Header의 Signature
 - ElfFile (45 6C 66 46 69 6C 65 00)
- First/Last Chunk Number
 - 소속된 Chunk의 처음과 마지막 Chunk 번호
- Next Record Identifier
 - 가장 처음 기록된 Record의 Identifier
- Header Size
 - File Header의 크기 = 0x80

Offset	Size	Value	Description
0	8	ElfFile\x00	Signature
8	8		First chunk number
16	8		Last chunk Number
24	8		Next record identifier
32	4	128	Header size
36	2		Minor format version
38	2		Major format version
40	2	4096	Header block size (or chunk data offset)
42	2		Number of chunks
44	76		Unknown (Empty values)
120	4		File flags
124	4		Checksum CRC32 of the first 120 bytes of the file header
128	3968		Unknown (Empty values)

2 구조와 분석 방법

EVTX 파일 Header 구조



- Format versions

Version (Major.Minor)	Description
3.1	Seen on Windows Vista and later
3.2	Seen on Windows 10 (2004) and later

- File flags

Value	Identifier	Description
0x0001		Is Dirty
0x0002		Is full

2 구조와 분석 방법

EVTX 파일의 Chunk 구조

- Chunk Header의 Signature
 - ElfChnk (45 6C 66 43 68 6E 6B 00)
- First/Last Event Record Number
 - EVTX 파일 Header의 Chunk Number와 유사
 - 해당 Chunk 내의 Event Record의 처음과 마지막 Record 번호
- First/Last Event Record Identifier
 - Chunk 내의 Record의 처음과 마지막 Identifier 값
- Header Size
 - Chunk Header의 Size = 0x80
- Last Event Record Offset
 - 해당 Chunk의 마지막 레코드 Offset
- Free Space Offset
 - Slack 공간의 시작부분 Offset
- Chunk Header : 두 개의 Checksum 가짐
 - 첫 번째 Checksum : CRC32#1 – Chunk 내의 Record 검증
 - 두 번째 Checksum : CRC32#2
- String Offset Array
 - Event Record 값 참조
- Template Ptf
 - Record의 Template의 Offset이 기록

2 구조와 분석 방법

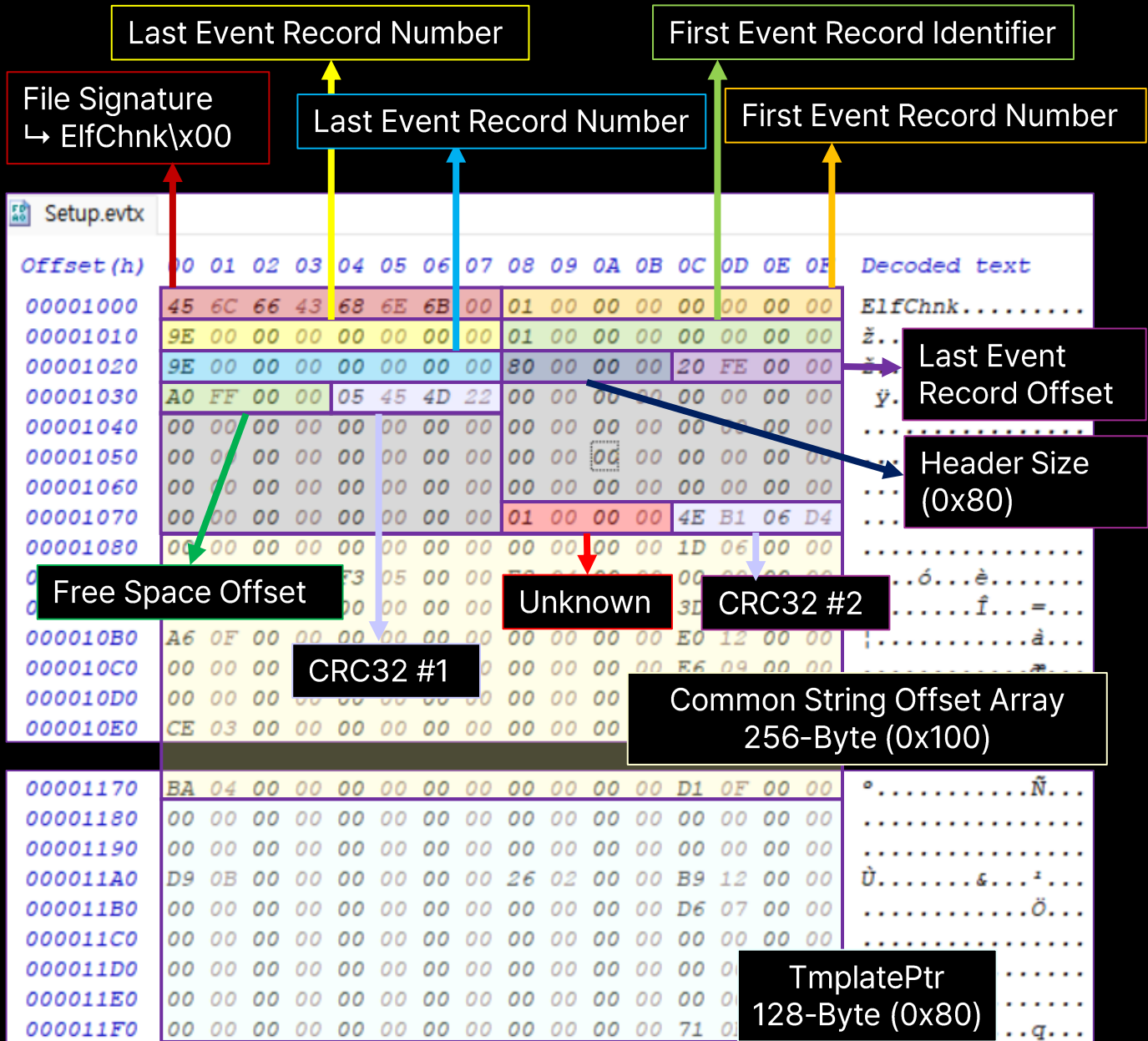
EVTX Chunk Header 구조

Offset	Size	Value	Description
0	8	ElfChnk\x00	Signature
8	8		First Event Record number
16	8		Last Event Record Number
24	8		First Event Record Identifier
32	4		Last Event Record Identifier
40	4	128	Header Size
44	4		Last Event Record Data Offset
48	4		Free Space Offset
52	4		Event Record Checksum (CRC32#1)
56	64		Empty Value
120	4		Flags
124	4		Checksum (CRC32#2)
128	256		String Offset Array (Chunk의 시작과 관련)
384	128		Template Ptr (Record의 해당 Template offset 값)

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
0x00	Signature ("ElfChnk\x00")								First Event Record Number							
0x10	Last Event Record Number								First Event Record Identifier							
0x20	Last Event Record Identifier								Header Size (0x80)				Last Event Record Offset			
0x30	Free Space Offset				CRC32 #1 (Record)				Empty Value							
0x40																
0x50																
0x60																
0x70																
...									Unknown				CRC32 #2			
0x170 0x180	Common String Offset Array 256-Byte (0x100)															
...																
0x1f0 0x200	TemplatePtr 128-Byte (0x80)															

2 구조와 분석 방법

EVTX Chunk Header 구조



- EVTX Chunk Header
 - 가장 먼저 저장된 Record가 있는 Chunk 정보를 가짐
- Checksum 이후 Header를 구성하는 두 가지 값이 추가적으로 존재
 - Chunk의 해당하는 Record들의 주소값을 참조하는 값

2 구조와 분석 방법

EVTX Record 구조

Offset	Size	Description
0	4	File Signature (0x2A2A0000)
4	4	Event Record Size
8	8	Event Record Identifier
16	8	Written Date and Time Contains a Filetime
24	...	Binary XML
...	4	Copy of Size

- EVTX Record File Signature
 - 2A 2A 00 00
- EVTX Record의 Size
 - 두 곳에 저장
 - Offset : 4 그리고 Record의 마지막 4 byte에 저장됨
- Event Record Identifier
 - Event Record의 식별번호
- Written Date and Time Contains a Filetime
 - Windows64 FileTime을 사용
 - UTC를 따르며, 국내에서 사용되는 컴퓨터의 경우 UTC+9:00
- Binary XML
 - 실제 이벤트와 관련된 정보는 Binary XML 형태로 저장

2 구조와 분석 방법

EVTX Record 구조

File Signature
↳ \x2A2A0000

Event Record Size
↳ 길이 : 8c8

Event Record Identifier

Written Date and Time

Binary XML

두 번째 레코드 시작 Signature

길이(h): 8C8

Offset(h) 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F Decoded text

00001200 2A 2A 00 00 C8 08 00 00 01 00 00 00 00 00 00 00 ***.È.....

00001210 6E B4 67 C7 A1 74 DB 01 0F 01 01 00 0C 01 50 D0 n'gÇ;tÛ.....PD

00001220 68 CF 26 02 00 00 00 00 00 00 50 D0 68 CF 4E 38 hI&.....PDhIÑ8

00001230 AC 81 D3 AA E6 D2 03 D8 40 61 AD 04 00 00 0F 01 -QÓ²æÒ.Ø@a.....

00001240 01 00 41 FF FF A1 04 00 00 4D 02 00 00 00 00 00 ..Äÿÿ;...M.....

00001250 00 BA 0C 05 00 45 00 76 00 65 00 6E 00 74 00 00 .°...E.v.e.n.t..

00001260 00 87 00 00 00 06 6A 02 00 00 00 00 00 00 BC 0F .‡....j.....‡.

00001270 05 00 78 00 6D 00 6C 00 6E 00 73 00 00 00 05 01 ..x.m.l.n.s.....

00001280 ...

00001AC0 00 00 00 00 C8 08 00 00 2A 2A 00 00 98 02 00 00È...***.~...

00001AD0 02 00 00 00 00 00 00 00 60 9B 80 B2 A2 74 DB 01`>€²tÛ.

00001AE0 0F 01 01 00 0C 01 50 D0 68 CF 26 02 00 00 12 00PDhI&.....

00001AF0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00PDhI&.....

00001B00 08 00 02 00 00 00 00 00 15 00 08 00 00 00 00PDhI&.....

00001B10 00 00 04 00 08 00 00 00 00 00 00 00 00 00 00PDhI&.....

00001B20 04 00 0C 00 13 00 00 00 00 00 00 00 00 00 00PDhI&.....

- EVTX Record File Signature
 - 2A 2A 00 00
- EVTX Record의 Size
 - 08 C8
- Event Record Identifier
 - 01
- Filetime
 - 2023-12-01 05:15:11
- Binary XML

References

- https://learn.microsoft.com/ko-kr/visualstudio/profiling/events-viewer?view=vs-2022&utm_source=chatgpt.com
- <https://learn.microsoft.com/ko-kr/windows/win32/eventlog/eventlog-key>
- <https://learn.microsoft.com/ko-kr/training/modules/manage-monitor-event-logs/2-describe-windows-server-event-logs>
- <https://learn.microsoft.com/en-us/windows/win32/eventlog/event-log-file-format>
- <https://isc.sans.edu/diary/25858>
- <https://docs.microsoft.com/en-us/windows/win32/etw/about-event-tracing>
- <https://support.kaspersky.com/help/keswin/12.8/ko-KR/235321.htm>
- [https://github.com/libyal/libevtx/blob/main/documentation/Windows%20XML%20Event%20Log%20\(EVTX\).asciidoc](https://github.com/libyal/libevtx/blob/main/documentation/Windows%20XML%20Event%20Log%20(EVTX).asciidoc)
- 최요한, 김준호, 박정흠, 이상진. (2021). Windows Event Trace Log 포렌식 해석 및 활용. 디지털포렌식연구, 15(3), 27-38.
- 강세림. (2019). 윈도우 이벤트 로그(EVTX) 분석 및 포렌식 활용방안. [석사학위논문, 국민대학교]
- 신용학. (2017). 윈도우 이벤트 로그 파일 (EVTX) 삭제 및 위변조에 대한 디지털 포렌식 복구 기술 연구 [석사학위논문, 국민대학교]