https://doi.org/10.33778/kcsa.2020.20.5.027

XML기반 Windows Event Log Forensic 도구 설계 및 구현

김 종 민*, 이 동 휘**

요 약

Windows Event Log에는 시스템의 전반적인 동작들을 정의하고 있는 Log이며, 해당 파일에는 사용자의 여러 행위 및 이상 징후를 탐지할 수 있는 데이터가 저장되어 있다. 하지만 행위마다 Event Log가 발생함으로써, 로그들을 분석할 때, 상당한 시간이 소요된다. 따라서 본 연구에서는 NSA에서 발표한 “Spotting the Adversary with Windows Event Log Monitoring”의 주요 Event Log 목록을 바탕으로 XML 기반한 Event Log 분석 도구를 설계 및 구현 하였다.

XML-based Windows Event Log Forensic tool design and implementation

Jongmin Kim*, DongHwi Lee**

ABSTRACT

The Windows Event Log is a Log that defines the overall behavior of the system, and these files contain data that can detect various user behaviors and signs of anomalies. However, since the Event Log is generated for each action, it takes a considerable amount of time to analyze the log. Therefore, in this study, we designed and implemented an XML-based Event Log analysis tool based on the main Event Log list of "Spotting the Adversary with Windows Event Log Monitoring" presented at the NSA.

Key words : Windows Event Log, XML, NSA, Forensic

접수일(2020년 11월 30일), 수정일(1차: 2020년 12월 22일), 게재확정일(2020년 12월 31일)

* 동신대학교/융합정보보호학과 교수(주저자)
** 동신대학교/융합정보보호학과 교수(교신저자)

1. 서 론

전 세계적으로 PC가 보급되면서 가정에서 사용하는 개인용 PC 및 기업에서 사용하는 PC의 대부분의 OS는 Windows 운영체제를 사용하고 있다.

Windows 운영체제의 증가로 인해 해당 운영체제에 대한 위협이 발생하게 되면, 필연적으로 분석을 실시하며, 분석 Log들은 메모리, 네트워크, 레지스트리, Windows Event Log 등 여러 가지들이 있다.

Windows Event Log는 윈도우에서 정립한 동작에 대하여 Application. System, Security Log와 같이 세 가지 로그들을 기록하고 있는데, 이러한 Windows Event Log는 포렌식 분석에서 유용하게 사용할 수 있다.

따라서 본 연구에서는 NSA에서 발표한 “Spotting the Adversary with Windows Event Log Monitoring”의 주요 Event Log를 목록화 및 분석하여 XML 기반 Event Log 분석도구를 설계하였으며, 이를 구현하기 위해 Evnet Log를 정립 및 자동화 분석 도구를 구현하였다.

2. 관련연구

2.1 Windows Event Log

Windows 시스템은 Application Log, Security Log, System Log와 같이 세 가지 로그를 이벤트에 기록하며, OS 구성에 따라 Directory Service Log, File Replication Service Log, DNS Server Log가 추가될 수가 있다[1]. 주요 이벤트 별 특징은 <표 1>과 같다.

<표 1> 윈도우 시스템 Event Log 종류[2][3]

Event Log	설명
Application	응용 프로그램이 기록한 다양한 이벤트가 저장되며, 기록되는 이벤트는 해당 제품의 개발자에 의해 결정된다. ex) 안티바이러스 제품의 경우 악성코드 탐지 및 업데이트를 기록한다. 일반 응용프로그램의 경우 활성화 여부와 성공 여부 등에 대한 정보를 기록한다.
Security	유효하거나 유효하지 않은 로그온 시도 및 파일 생성, 열람, 삭제 등의 리소스 사용에 관련된 이벤트를 기록한다. 감사로그 설정을 통해 다양한 보안 이벤트 저장이 가능하다.
System	Windows 시스템 구성요소가 기록하는 이벤트로 시스템 부팅 시 드라이버가 로드 되지 않는 경우와 같이 구성요소의 오류를 이벤트에 기록한다.

2.2 Windows Event Log 구성요소

이벤트 로그는 이벤트 뷰어를 통해 확인할 수 있으며, 각 로그들은 메타데이터와 메시지를 확인 할 수 있다.

메타데이터는 Channel, Provider, Task, Level, TimeCreated, Keywords, EventRecordID, Version, Computer, Opcode로 구성되어 있으며, (그림 1)과 같다.

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
  - <System>
      <Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-
        5478-4994-a5ba-3e3b0328c30d}" />
      <EventID>4799</EventID>
      <Version>0</Version>
      <Level>0</Level>
      <Task>13826</Task>
      <Opcode>0</Opcode>
      <Keywords>0x8020000000000000</Keywords>
      <TimeCreated SystemTime="2020-12-15T10:42:46.9748136Z" />
      <EventRecordID>30874</EventRecordID>
      <Correlation ActivityID="{afc0bea3-ce85-0002-36bf-c0af85ced601}" />
      <Execution ProcessID="976" ThreadID="222892" />
      <Channel>Security</Channel>
      <Computer>DESKTOP-C5M61OM</Computer>
      <Security />
    </System>
  - <EventData>
      <Data Name="TargetUserName">Backup Operators</Data>
      <Data Name="TargetDomainName">Builtin</Data>
      <Data Name="TargetSid">S-1-5-32-551</Data>
      <Data Name="SubjectUserSid">S-1-5-18</Data>
      <Data Name="SubjectUserName">DESKTOP-C5M61OM$</Data>
      <Data Name="SubjectDomainName">WORKGROUP</Data>
      <Data Name="SubjectLogonId">0x3e7</Data>
      <Data Name="CallerProcessId">0x31654</Data>
      <Data Name="CallerProcessName">C:\Windows\System32
        \svchost.exe</Data>
    </EventData>
  </Event>
```

(그림 1) 이벤트로그 파일의 XML

3. 제안하는 방법

본 연구에서 NSA의 “Spotting the Adversary with Windows Event Log Monitoring”주요 Event Log 목록을 바탕으로 XML 기반한 Event Log 분석 도구를 설계 및 구현하기 위해, 주요 Event Log를 분석하였다.

3.1 주요 Event Log

NSA의 주요 Event Log는 16개의 이벤트 카테고리로 나누어져 있으며, <표 2>과 같이 16개의 카테고리로 나누어져 있다.

<표 2> 최상위 항목 요소[3][4]

카테고리			
Clearing Event Logs	Account Usage	Remote Desktop Logon Detection	Windows Defender Activities
Application Crashes	Software & Service Installation	External Media Detection	Pass the Hash Detection
AppLocker	System or Service Failures	Windows Update Errors	Kernel Driver Signing
Group Policy Errors	Mobile Device Activities	Printing Services	Windows Firewall

<표 3>는 16개의 카테고리의 상위 단계에 종속된 하위항목에 대하여 정리한 것이다.

<표 3> 하위항목 요소[3][4]

General Event Descriptions	General Event IDs
Account and Group Activities	4728, 4732, 4634, 4735, 4740, 4756
Application Crashes and Hangs	1000 and 1002
Windows Error Reporting	1001
Blue Screen of Death(BSOD)	1001
Windows Defender Errors	1005, 1006, 1008, 1010, 2001, 2003, 2004, 3002, 5008
Windows Integrity Errors	3001, 3002, 3003, 3004, 3010 and 3023
EMET Crash Logs	1 and 2
Windows Firewall Logs	2004, 2005, 2006, 2009, 2033
MSI Packages Installed	1022 and 1033
Windows Update Installed	2 and 19
Windows Service Manager Errors	7022, 7023, 7024, 7026, 7031, 7032, 7034
Group Policy Errors	1125, 1127, 1129
AppLocker and SRP Logs	865, 866, 867, 868, 882, 8003, 8004, 8006, 8007
Windows Update Errors	20, 24, 25, 31, 34, 35
Hotpatching Error	1009
Kernel Driver and Kernel Driver Signing Errors	5038, 6281, 219
Log Clearing	104 and 1102
Kernel Filter Driver	6
Windows Service Installed	7045
Program Inventory	800, 903, 904, 905, 906, 907, 908
Wireless Activities	8000, 8001, 8002, 8003, 8011, 10000, 10001, 11000, 11001, 11002, 11004, 11005,

	11006, 11010, 12011, 12012, 12013
USB Activities	43, 400, 410
Printing Activities	307

3.2 XML 구조

XML 문서는 XML 선언 부분으로 문서구조를 정의하는 프롤로그와 실제 XML 문서의 내용을 입력하는 도큐먼트 인스턴스 부분으로 크게 두 부분으로 구성되어 있다. 또한 XML 문서의 구조는 전통적인 DTD와 새로운 XML Schema에 의하여 기술된다[5].

3.2.1 DTD(Document Type Definition)[5][6]

DTD는 문서 내에서 사용할 태그들을 정의하기 위한 일련의 문서 구조에 대한 규칙이다. DTD는 문서에 사용할 수 있는 태그, 문서에 나타나는 태그들의 순서와 출현 횟수, 태그에서 허용되는 애트리뷰트 등을 계층적 구조로 규정한다.

XML은 HTML처럼 고정적인 태그를 갖지 않고 콘텐츠에 따라 융통성 있고 자유로운 태그에 의하여 정의된다. 데이터 교환을 위해 XML을 사용하고자 하는 기업이나 단체는 그 자신의 독자적인 DTD들을 정의하여 사용한다. 엘리먼트, 애트리뷰트와 같이 문서를 구성하는 기본요소들로부터 파라미터 엔티티, 조건절 등을 사용하여 여러 상황에 따라 맞춤화 할 수 있는 기능을 제공한다.

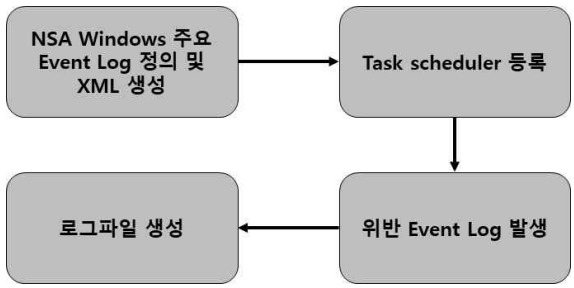
3.2.2 XML Schema[5][7]

DTD의 확장성, 데이터 타입, 통합, 재사용 등의 문제점을 해결하기 위하여 2004년 W3C에 의하여 표현력이 풍부하고, 설명하지 않아도 직관적으로 알 수 있고, 서술적이지 않고 간결하고, 인터넷에서 사용될 수 있고, 다양한 XML 관련 스펙과 협력할 수 있고, 정보 처리에 상호 운용(interoperable)할 수 있고, XML에 의하여 표현되는 새로운 타입의 XML Schema가 개발되었다.

XML Schema는 그 자체가 XML 문서이며 XML과 동일한 문법을 이용함으로써 개발자에게 보다 친숙하다.

스트링, 정수, 날짜 등의 풍부한 데이터 타입을 지원하며, 또한 사용자 정의 타입을 정의하는 기능을 제공한다. 또한 기존 타입에서 확장하거나 축소하여 변형된 새로운 타입을 생성할 수도 있고, 데이터 무결성을 지원하는 메카니즘을 제공한다.

3.3 분석 도구 설계 및 구현 과정



(그림 2) Windows 주요 Event Log 분석 도구 구현 절차

(그림 2)는 본 연구의 분석 도구 설계 및 구현과정을 나타낸 것이다.

첫 번째로 NSA Windows 주요 Event Log 들을 분석한 후 목록화 하여 XML 파일로 생성을 하게된다.

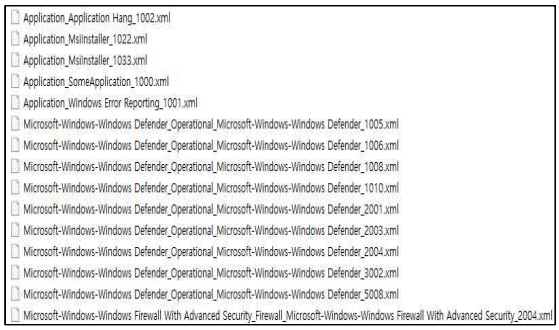
두 번째로 생성된 XML들을 Task scheduler에 등록을 한다. Task scheduler에 등록을 하게되면 등록된 Event Log들이 발생되면 자동적으로 Log가 쌓이게 되는 것이다.

4. Windows 주요 Event Log 분석 도구 설계 및 구현

4.1 주요 Event Log 정의 및 XML 생성

주요 이벤트를 목록화 하여, 이벤트 뷰어에서 XML을 추출한 다음, 특정 폴더에 XML파일을 저장한다.

(그림 3)은 주요 이벤트 로그에 대한 XML 파일 목록이다.



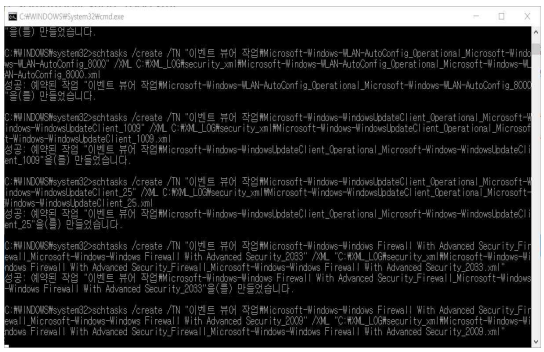
(그림 3) 주요 이벤트 로그에 대한 XML 목록

4.2 Task scheduler 등록

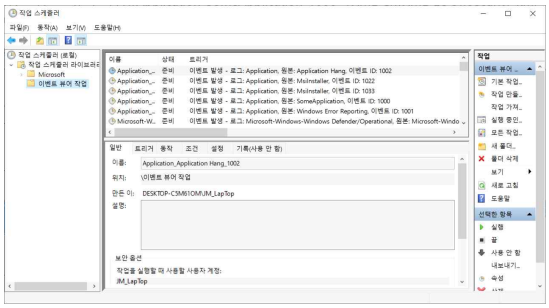
주요 이벤트 로그에 대한 XML을 목록화 하였다면, 작업 스케쥴러에 등록하며, (그림 4)는 작업 스케쥴러에 등록하기 위한 bat 파일이다.

(그림 4) task_scheduler_command.bat 파일

작업 스케쥴러에 자동으로 등록하기 위해 배치 파일 실행하게 되면 (그림 5)와 같이 진행되며, 완료와 동시에 (그림 6)과 같이 작업 스케줄러에 업로드 된다.



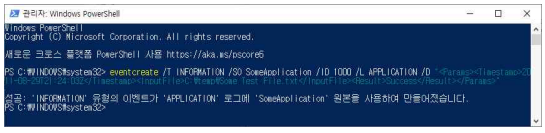
(그림 5) task_scheduler_command.bat 파일 실행



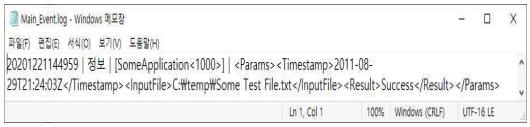
(그림 6) 작업 스케줄러 업로드

4.3 Event Log 발생 및 Log 기록

작업 스케줄러에 업로드 되고나면 최종적으로 업로드 된 주요 Event들이 발생을 하게 되면 Main_Event.log라는 이름으로 이벤트가 발생할 때마다 기록하게 된다.



(그림 7) task_scheduler에 등록된 Event 발생



(그림 8) ain_Event.log 내용

(그림 7)은 Eventlog를 발생시키기 위한 명령어이고, 이벤트가 발생하면 Main_Event.log라는 이름으로 저장되며, (그림 8)과 같이 로그에 대한 정보가 입력된다.

5. 결론

XML은 다양한 종류의 DATA에 적용될 수 있는 유연성과 확장성을 가지고 있기 때문에 여러 계층의 응용프로그램에서 적용될 수 있고, 웹뿐만 아니라 다양한 분야에서 사용되고 있다.

본 논문에서는 NSA에서 발표한 “Spotting the

Adversary with Windows Event Log Monitoring”의 주요 Event Log를 목록화 하였으며, Event Log들의 XML 파일로 추출하여, XML 기반한 Event Log 분석 도구를 설계 및 분석하였다.

Event Log에서는 보안과 관련된 상당한 정보를 가지고 있으며, 분석자가 Event Log를 분석하기 위해서는 Event Viewer를 실행해 하나하나씩 분석을 해야 하는 불편함이 있었다. 하지만 본 논문에서 제안한 Event Log 분석 도구를 사용하게 된다면, 주요 Event Log만 등록한다면, 해당 Event Log에 대해서 실시간으로 분석을 할 수 있어 신속한 분석이 가능하다.

참고문헌

[1] 국가사이버안전센터, "정보보안 관리실태 평가 소개", 한국정보보호학회논문지, Vol. 23, No. 5, pp. 9-11, 2013.

[2] Minasi, Mark, Gibson, Darril, Finn, Aidan, Henry, “Mastering Windows Server 2008 R2”, Wiley, p. 921, 2012. 08.

[3] 김종민, 김동민, 이동휘, "제어시스템의 내부자 위협 탐지를 위한 Event Log 타당성 및 중요도분석에 관한 연구", 융합보안논문지, Vol. 18, No. 3, pp. 77-85, 2018.

[4] Spotting the Adversary with Windows Event Log Monitoring: An Analysis of NSA Guidance, February 28, 2013.

[5] 김태권, “XML 문서 처리에 관한 연구”, 정보과학회논문지, Vol. 43, No. 4, pp.489-496, 2016.

[6] Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau, "Extensible Markup Language (XML) 1.0 (Fifth Edition)," [Online]. Available: http://www.w3.org/TR/xml, 2008.

[7] Shudi S. Gao, C. M. Sperberg-McQueen, Henry S. Thompso, "W3C XML Schema Definition Language(XSD) 1.1 Part 1: Structures," [Online]. Available: http://www.w3.org/TR/xmlschema11-1, 2012.

〔저자소개〕

김 종 민 (Jongmin Kim)
2015년 경기대학교 산업보안학박사
현 재 동신대학교 에너지융합대학
에너지응용학부 융합정보보안전공 교수

email : dyuo1004@dsu.ac.kr

이 동 휘(DongHwi Lee)
2007년 경기대학교 정보보호박사
2011년~2012년 University of Colorado Denver, Dept. of Computer Science and Engineering
현 재 동신대학교 에너지융합대학
에너지응용학부 융합정보보안전공 교수

email : dhclub@dsu.ac.kr