KIPS Tr. Comp. and Comm. Sys.
Vol.5, No.4 pp.95~102 pISSN: 2287-5891

http://dx.doi.org/10.3745/KTCCS.2016.5.4.95

Digital Forensic Indicators of Compromise Format(DFIOC) and Its Application

Lee Min Wook[†] · Yoon Jong Seong[††] · Lee Sang Jin[†††]

ABSTRACT

Computer security incident such as confidential information leak and data destruction are constantly growing and it becomes threat to information in digital devices. To respond against the incident, digital forensic techniques are also developing to help digital incident investigation. With the development of digital forensic technology, a variety of forensic artifact has been developed to trace the behavior of users. Also, a diversity of forensic tool has been developed to extract information from forensic artifact. However, there is a issue that information from forensic tools has its own forms. To solve this problem, it needs to process data when it is output from forensic tools. Then it needs to compare and analyze processed data to identify how data is related each other and interpret the implications. To reach this, it calls for effective method to store and output data in the course of data processing. This paper aims to propose DFIOC (Digital Forensic Indicators Of Compromise) that is capable of transcribing a variety of forensic artifact information effectively during incident analysis and response. DFIOC, which is XML based format, provides "Evidence" to represent various forensic artifacts in the incident investigation. Furthermore, It provides "Forensic Analysis" to report forensic analysis result and also gives "Indicator" to investigate the trace of incidence quickly. By logging data into one sheet in DFIOC format for forensic analysis process, it is capable of avoiding unnecessary data processing. Lastly, since collected information is recorded in a normalized format, data input and output becomes much easier as well as it will be convenient to use for identification of collected information and analysis of data relationship.

Keywords : Incident Response, Digital Forensic, Forensic Artifacts Collecting Format, Indicator of Compromise(IOC)

디지털 포렌식 기반의 침해 지표 포맷 개발 및 활용 방안

이 민 욱[†] · 윤 종 성[††] · 이 상 진[†††]

요 약

기밀 정보 유출, 데이터 파괴 등 디지털 기기에 저장된 정보를 위협하는 침해사고가 계속해서 증가하고 있다. 이와 함께 디지털 침해 사고를 조사하기 위하여 디지털 포렌식 기술 또한 계속해서 발전해 왔다. 디지털 포렌식 기술의 발전으로 인하여 사용자의 행위를 추적할 수 있는 다양한 포렌식 아티팩트들이 발견되었으며, 포렌식 아티팩트로부터 정보를 추출하기 위한 다양한 포렌식 도구가 개발되었다. 하지만 포렌식 도구에서 출력하는 정보는 각기 다른 양식을 갖고 있다. 따라서 포렌식 도구에서 출력하는 정보를 다시 가공해야 하는 작업이 필요하다. 가공된 데이터는 데이터 간의 정보를 비교 분석하여 연관관계를 도출하고 그 의미를 파악해야 한다. 이를 위하여 데이터를 가공하는 작업에서 데이터의 저장과 출력을 효과적으로 하기 위한 방안이 필요하다. 본 논문에서는 침해사고 조사 분석시 필요한 다양한 포렌식 아티팩트 정보를 효과적으로 기술할 수 있는 디지털 포렌식 침해지표 작성 포맷 DFIOC(Digital Forensic Indicators Of Compromise)를 제안한다. DFIOC는 XML 기반의 포맷이며 침해사고 조사에 필요한 다양한 포렌식 아티팩트 정보를 Evidence로 표현하여 기술할 수 있다. 또한 포렌식 분석 결과를 기록하는 Forensic Analysis를 제공하고 있으며, 침해 흔적을 기록하기 위하여 Indicator 항목을 제공하고 있다. 포렌식 분석 과정에 필요한 데이터를 DFIOC 포맷의 문서 하나로 기록할 수 있게 됨으로써 불필요한 데이터 가공이 발생하지 않게 된다. 또한 정규화된 포맷을 통해 수집된 정보를 기록하기 때문에 입출력이 쉬워지며 수집된 정보를 확인하고 상호 연관관계 분석에 활용하기 쉬워진다.

키워드 : 침해 사고 분석, 디지털 포렌식, 포렌식 아티팩트 수집 포맷, 침해지표

1. 서 론

사용자 개인정보, 기업의 문서 등 다양한 형태의 중요 정보들이 컴퓨터에 저장되어 있으며 저장된 데이터 양이 계속해서 증가하고 있다. 이에 따라 사이버 침해 사고로 인한 피해 또한 점점 증가하고 있다[1]. 사이버 침해 사고에 대응하기 위하여 디지털 포렌식 기술이 계속해서 발전해 왔으며, 디지털 포렌식 기술의 발전과 함께 침해 흔적을 조사할 수 있는 다양하고 유용한 포렌식 아티팩트들이 발견되어 조사에 활용되고 있다. 다양한 포렌식 아티팩트를 분석하기 위해서는 신뢰성 있게 포렌식 아티팩트를 분석하고 정보를 출력해주는 도구를 사용해야 한다[2]. 각기 다른 도구에서 출

※ 이 논문은 2016년도 국방과학연구소의 지원으로 수행되었음 (계약번호 : UD140078ED).
† 준 회 원 : 고려대학교 정보보호학과 석사과정
†† 비 회 원 : 고려대학교 정보보호학과 석·박사통합과정
††† 종신회원 : 고려대학교 교수
Manuscript Received : March 10, 2016
First Revision : April 7, 2016
Accepted : April 11, 2016
* Corresponding Author : Lee Sang Jin(sangjin@korea.ac.kr)

력된 정보 양식이 다르기 때문에 추출된 정보들 간의 연관관계 분석을 위하여 포렌식 조사 과정 중 정보의 가공이 필연적으로 발생하게 된다. 포렌식 조사 과정은 데이터 수집 및 조사 단계. 분석 단계, 보고서 작성 단계로 이루어진다[3]. 각각의 단계에서 사용하는 데이터 처리 방법이 다르기 때문에 매번 데이터 가공을 수행하게 된다. 이 같은 정보의 가공은 침해시스템으로부터 수집된 데이터의 양이 증가할수록 조사관에게 더욱 더 많은 부하가 된다. 침해 사고의 발생 빈도와 규모가 증가함에 따라 많은 양의 아티팩트를 분석하기 위해 좀 더 효율적으로 데이터를 저장하고 활용하는 방안이 요구되고 있다.

디지털 포렌식 조사 과정에서 불필요한 데이터 가공이 발생하는 이유는 침해 사고를 조사하는데 있어 표준화된 데이터 처리 방안이 준비되어 있지 않기 때문이다. 따라서 표준화된 데이터 기술 방식을 개발하고 활용함으로써 분석과정에 발생하는 불필요한 데이터를 가공하는 작업들을 보다 단순화 하고 효율적으로 처리할 수 있다. 본 논문에서는 많은 양의 데이터를 가공하고 저장할 뿐만 아니라 분석 결과를 저장하고 침해 흔적을 기술할 수 있는 XML 기반의 표준화된 포맷인 Digital Forensic Indicators Of Compromise (DFIOC)를 소개한다.

2. 관련 연구

침해 사고 흔적을 정형화된 포맷으로 기술하고 활용하고자하는 다양한 연구가 존재한다. CybOX는 MITRE에서 2011년 공개한 XML 기반의 포맷이다. CybOX는 시스템에서 관측 가능한 정보를 Object는 표현하여 기술하도록 설계되어 있으며, Object는 레지스트리, 파일, 네트워크 정보와 같이 시스템에서 표현 가능한 개체를 의미하는 것으로 침해 시스템에서 발견되는 다양한 정보들을 기술 가능하도록 설계되어 있다[4]. CybOX를 기반으로 다양한 응용포맷이 발표되고 있으며, 포렌식 관점에서 침해 사고를 정규화하여 작성하기 위해 DFAX가 개발되었다[5-6]. DFAX는 침해시스템에서 발견된 흔적을 CybOX로 작성하고, 작성된 정보를 사용하여 정보가 외부로 유출된 것인지 또는 유입된 것인지와 같은 포렌식 관점의 정보를 기술할 수 있도록 개량되었다. 또한 침해시스템을 조사한 조사관 정보나 포렌식 조사를 요청한 사람의 인적 정보 등 디지털 포렌식을 수행하는데 필요한 항목들이 추가되어 있다.

하지만 CybOX는 포렌식 조사를 위해 설계되지 않았다. 따라서 AMCache, Shellbag과 같은 중요한 포렌식 아티팩트를 Object로 명확히 구분하고 있지 않으며, 포렌식 관점에서 분석에 활용할 수 있는 다양한 정보들을 구체화 하여 표현하고 있지 않다. DFAX는 CybOX 기반으로 개발되었기 때문에 CybOX의 이러한 문제점을 그대로 상속하고 있다.

OpenIOC는 Mandiant에서 2008년에 발표된 침해지표 포맷이다. 침해지표는 침해 시스템의 분석 결과로 확인된 내용을 기록하여 향후 탐지 시그니처로 활용하기 위한 포맷이다[7]. 파일의 해시 값, 크기, 경로 또는 레지스트리 등 침해 시스템에서 확인할 수 있는 다양한 종류의 정보를 표현할 수 있다. OpenIOC를 통해 기록된 정보는 또 다른 침해 사고 조사 시 동일한 침해 사고 흔적이 존재하는지 파악하기 위한 용도로 사용한다. 따라서 OpenIOC는 백신의 악성코드 시그니처와 유사한 탐지 규칙으로 침해 흔적을 표현할 수 있지만 침해 시스템으로부터 수집된 정보를 분석에 활용할 수 있을 만큼 포렌식 아티팩트를 충분히 표현하진 못한다.

DFXML은 저장장치에 저장되어 있는 Raw 데이터를 표현하는 것에 특화되어 있는 포맷으로 2009년 Garfinkel에 의해 개발되었다[8-9]. DFXML은 침해 사고 의심 시스템에서 수집한 저장장치를 분석하여 파일의 오프셋 단위까지 정교하게 기술 가능하도록 설계되어있을 뿐만 아니라, 저장장치에서 삭제된 데이터의 위치까지 기술 가능하도록 설계되어 있다. 하지만 DFXML 포맷은 낮은 수준의 정보를 표현하기에 적합하게 개발되어 있어 파일간의 관계를 기술하거나, 로그인, USB 연결과 같은 시스템 상에서 발생한 이벤트간의 연관관계를 표현하는 것에는 한계가 있다.

DFAX는 분석 결과를 요약하여 정규화된 디지털 포렌식 보고서를 작성하기 위한 목적으로 개발되어 있으며, OpenIOC는 침해 시스템 분석결과를 침해지표로 정리하여 탐지 시그니처를 작성하고 작성된 탐지 시그니처를 활용하여 침해 시스템을 진단하기 위한 목적으로 개발되었다. 또한 DFXML은 침해 시스템에서 수집된 정보를 정밀히 기록하기 위한 목적을 갖고 있다. 결과적으로 기존 개발된 포맷들은 데이터의 수집, 분석 및 보고서와 침해지표를 작성하는 과정 모두를 포괄하도록 개발되어 있지 않으며, 디지털 포렌식 조사 과정 중 일부만을 처리할 수 있도록 개발되어 있다.

3. DFIOC(Digital Forensic Indicators Of Compromise)

침해 시스템을 조사하기 위해서는 침해 시스템으로부터 다양한 포렌식 아티팩트를 수집하고 분석해야 한다[10]. 포렌식 아티팩트는 Shellbag, 사용자 웹 사용 기록 등 다양한 종류가 있으며, IT 기기 활용 증가에 따라 포렌식 아티팩트에 저장된 정보의 양은 점점 많아지고 있다. 따라서 많은 양의 데이터를 분석하기 위하여 정형화된 데이터 처리 양식이 필요하다. XML(Extensible Markup Language)은 다양한 데이터를 기록하고 교환하는 것에 있어서 혼동이 없도록 명확한 데이터의 표현을 위하여 개발된 언어이다. Digital Forensic Indicators Of Compromise(DFIOC)는 XML을 기반으로 설계하였으며, 수집된 포렌식 아티팩트를 명확히 기록하고 분석에 활용할 수 있도록 설계하였다. Fig. 1은 DFIOC 구조와 활용을 나타낸 그림이다. DFIOC는 침해 시스템에서 수집된 다양한 포렌식 아티팩트 정보를 기록할 수 있다. Evidence는 포렌식 아티팩트와 포렌식 아티팩트의 추출 정보이다. 추출 정보는 단순히 수집된 그대로 저장할 수 있으며 포렌식 관점에서 의미를 추가하여 저장할 수 있다.

외부저장장치 연결, 실행 흔적과 같은 포렌식 관점의 정보를 포함하는 것은 조사관이 DFIOC에 기록된 정보를 더 쉽게 이해하고 분석할 수 있도록 한다. 수집된 정보를 활용하여 분석이 완료되면 Forensic Analysis 항목에 Evidence 간의 관계를 기록하여 포렌식 분석 보고서를 작성한다. 보고서 작성 완료 후 다음 단계에서는 Indicator 항목을 작성할 수 있다. Indicator 항목은 향후 다른 침해사고가 발생하였을 때 유사한 침해사고를 빠르게 확인하기 위하여 Evidence 논리 조합을 기록한 침해 흔적 탐지 규칙이다.

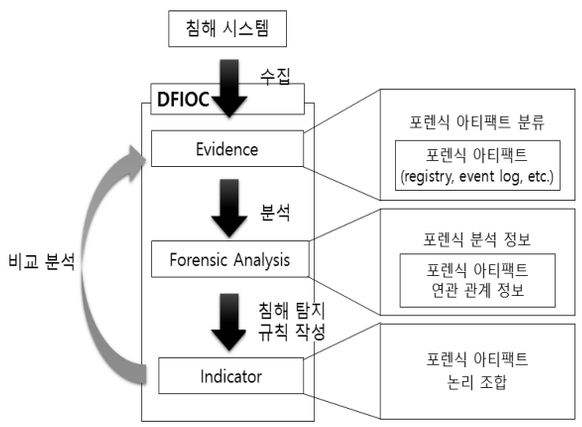

Fig. 1. DFIOC Abstract & Usage

3.1 Evidence

Evidence는 침해 시스템에서 수집되는 다양한 포렌식 아티팩트 또는 포렌식 아티팩트로부터 추출된 정보를 의미한다. DFIOC Evidence는 Windows XP부터 Windows 8까지 발견되어 포렌식 조사에 활용되고 있는 포렌식 아티팩트를 기록할 수 있는 항목을 정의하고 있다. 또한 수집되는 정보 중 불필요하게 중복되어 수집되는 데이터를 막기 위하여 각각의 Evidence에 고유 ID가 부여되어 중복된 정보를 입력할 필요가 있는 경우 ID 값을 참조하여 기록할 수 있다. 뿐만 아니라 ID 값은 포렌식 분석 후 Forensic Analysis를 작성할 때 이미 기록된 Evidence의 ID만을 참조하여 기술할 수 있도록 하고 있다. 이를 통해 반복해서 중복된 데이터를 입력하거나 오타로 인해 잘못된 데이터의 입력을 막을 수 있다.

Table 1은 DFIOC에서 표현 가능한 Evidence를 설명한 표이다. DFIOC Evidence는 다양한 포렌식 아티팩트 정보를 묶는 하나의 엘리먼트를 의미하며, 표는 Evidence 엘리먼트의 하위 6개의 엘리먼트 설명하고 있다. 다양한 포렌식 아티팩트 정보를 기록하기 위하여 많은 엘리먼트가 설계되었으며, 활용 목적과 수집 위치에 따라 각각의 엘리먼트는 6개로 분류된 엘리먼트의 하위 엘리먼트에 기록하도록 설계되었다. 조사할 침해시스템을 파악하기 위한 목적으로 수집되는 기본적인 정보는 SystemBasicInfo 엘리먼트의 하위 엘

Table 1. DFIOC Evidence Element

Element	하위 Element	설명	Element	하위 Element	설명
SystemBasic Info	HardwareInfo	시스템의 하드웨어 정보	UserSystem Activities	WebHistoryInfo	웹 페이지 접속 흔적
	OSInfo	시스템의 OS 정보		DownloadInfo	다운로드 흔적
	UserAccounts	시스템에서 사용 중인 계정정보		RecentDocInfo	최근 문서 열람 흔적
	UserGroups	시스템에서 사용 중인 유저 그룹 정보		ExternalDriveInfo	외부 저장장치 연결 흔적
	DiskInfo	시스템에서 사용 중인 디스크 및 파티션 정보		ShellBagInfo	폴더 변경 흔적
SystemSetting Info	NetworkConfig	시스템의 네트워크 설정 정보		PrefetchInfo	프로그램 실행 흔적
	InstalledApplication Info	시스템에 설치된 프로그램 정보		AMCacheInfo	프로그램 실행 흔적
	AutorunsInfo	자동 실행 프로그램 정보		CompatibilityInfo	프로그램 실행 흔적
	Services	시스템에 등록된 서비스 프로그램 정보		IconCacheInfo	프로그램 사용 흔적
	UpdateInfo	시스템의 업데이트 정보		EventLogInfo	프로그램 실행 흔적
	DriverInfo	시스템에 설치된 드라이버 정보		NtfsJrnlInfo	파일 관련 작업 흔적
	FirewallInfo	시스템 방화벽 설정 정보	SystemLive Info	ProcessInfo	활성 상태의 프로세스 정보
	SharedForlderInfo	공유폴더 정보		NetworkActivities	활성 네트워크 세션 및 네트워크 캐쉬 정보
	HostsFileInfo	윈도우 호스트 파일의 정보			
EvidenceFiles	File	Evidence 일반 파일 정보		Packets	네트워크 패킷 정보
	ExecutableFile	Evidence 실행 파일 정보	SecurityEvents	SecurityEvent	보안 이벤트 탐지 흔적

리먼트로 기록된다. 침해 시스템에서 사용된 프로그램 설치 및 설정 관련 정보는 SystemSettingInfo 엘리먼트 하위에 기록되어 사용자가 시스템을 사용한 환경을 파악하기 위한 목적으로 활용된다. 침해 시스템이 활성 상태일 때 활성 메모리에서 추출되는 정보를 수집하기 위하여 SystemLiveInfo 엘리먼트를 설계하였으며, 사용자 행위를 추적하기 위한 목적의 정보는 UserSystemActivities 엘리먼트 하위에 설계되어있다. 또한 침해 시스템에 저장된 실행 파일 정보 및 다양한 파일들의 기본적인 정보들을 수집하기 위하여 EvidenceFiles 엘리먼트를 설계하였다. 실행 파일은 Win32, Win64의 윈도우 실행 파일에 대한 정보를 작성할 수 있으며, 일반 파일은 문서 파일, 그림 파일 등 다양한 파일을 표현하기 위한 것으로 시간 정보를 중점적으로 표현할 수 있도록 설계하였다. 마지막으로 침해 시스템이 외부와 통신한 과정에서 발생한 보안 이벤트를 확인하기 위하여 보안 이벤트 정보를 SecurityEvents 엘리먼트로 분류하여 수집하도록 하고 있다.

작성 가능한 Evidence는 지금까지 발견된 포렌식 아티팩트를 기반으로 설계하였으며, 목적에 따라 분류하여 침해 시스템으로부터 수집된 정보를 찾기 쉽도록 하였다. 포렌식 아티팩트는 지속적으로 발견되고 있으며, 향후 추가적인 포렌식 아티팩트가 발견될 경우 Evidence를 추가하여 DFIOC를 사용할 수 있다.

3.2 Forensic Analysis

Forensic Analysis는 침해 시스템으로부터 수집된 Evidence를 분석하여 관계를 규정하기 위한 목적으로 설계된 항목이다. Forensic Analysis에 필요한 기본 정보로서 수사관 정보, 수사 날짜, 기간 등 포렌식 조사 정보를 기입할 수 있으며, Evidence 간의 관계를 기술하여 보고서를 작성할 수 있도록 설계하였다.

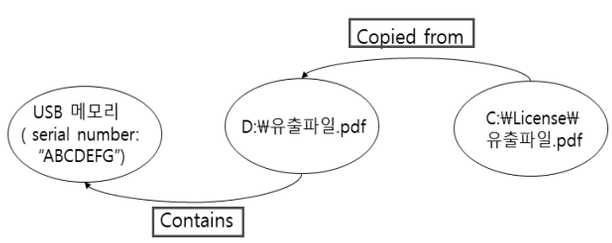

Fig. 2. Forensic Analysis Relation

Fig. 2는 Forensic Analysis의 예로 USB 메모리를 통해 파일이 유출된 분석 내용을 DFIOC의 Forensic Analysis로 나타낸 예이다. 침해시스템 조사 결과 Licence.pdf 파일이 복사되었음을 의미하는 “Copied from”과 Licence.pdf 파일이 USB에 저장되어 있다는 “Contains” 관계 기술어를 통하여 기록하고 있다. 이 같은 방식으로 특정 파일이 어떤 경로로 유출되었는지, 또는 악성코드가 어떤 경로로 감염되었는지 기록할 수 있다.

위와 같이 DFIOC Forensic Analysis는 “Copied from”, “Contains”와 같은 Evidence들 간의 관계 기술어를 정의하고 사전에 목록화하여 모호하게 표현될 가능성을 제거하고 있으며, 포렌식 분석 정보를 충실히 표현하도록 설계하였다.

3.3 Indicator

Indicator는 침해 흔적을 탐지 규칙으로 작성하여 향후 유사한 유형의 침해 사고가 발생하였을 때 빠르게 침해 흔적을 탐지하기 위한 항목이다. Indicator는 침해 의심 시스템을 분석 후 발견된 Evidence를 침해 흔적으로 규정하여 AND, OR의 논리 조합으로 기록한다. AND는 나열된 것이 모두 나타나야 함을 의미하며 OR는 나열된 것 중 하나만 해당되어도 됨을 의미한다. 또한 AND, OR의 단순한 논리 조합 매치로 인한 오탐률을 줄이기 위하여 DFIOC는 침해 흔적의 중요도(weight)를 기록할 수 있도록 포맷을 제공하고 있다. 중요도 정보는 Indicator에 기록된 Evidence들을 확실한 정보인지 불확실한 정보인지 수치로 기록하여 유연하게 침해 흔적을 탐지할 수 있다. 또한 자동화된 분석 가능 여부(Inspectable, Uninspectable)를 기록하여 작성된 Indicator를 다른 침해 의심 시스템을 조사할 때 조사관이 직접 확인해야 하는 정보인지 자동화 하여 탐지할 수 있는 정보인지 구분하여 침해 의심 시스템 조사에 활용할 수 있다.

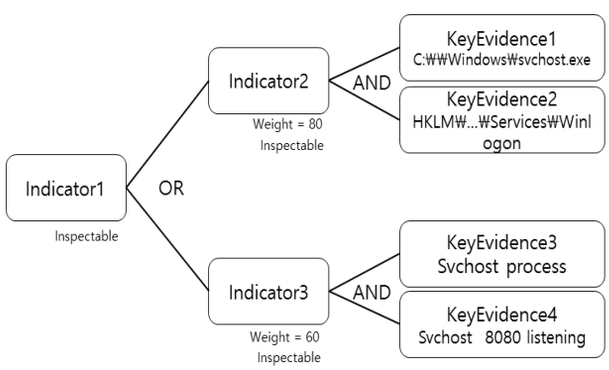

Fig. 3. Indicator XML Schema

Fig. 3은 Indicator를 단순화하여 나타낸 예다. C:\\\Windows 경로에 svchost.exe 파일이 존재하고 Winlogon 서비스가 등록된 것이 동시에(Indicator2) 확인되거나, svchost 프로세스가 8080 포트를 열고(Indicator3) 있다면 침해 흔적으로 판단할 수 있다고 작성되어 있다. Indicator2나 Indicator3은 둘 중 하나만 확인되어도 침해흔적으로 판단할 수 있으므로 OR를 사용하여 조합하였다. 또한 Indicator2와 Indicator3는 자동화된 분석이 가능하다는 의미로 Inspectable을 기록하였으며, 침해 중요도 정보로 Indicator2는 높은 확률로 동일한 침해 흔적인 것으로 판단할 수 있기 때문에 80으로 기록하였으며, Indicator3은 프로세스가 포트를 열고 있는 것은 동일한 정보가 수집되더라도 다른 침해 상황일 가능성이 있기 때문에 60의 낮은 중요도를 주었다. 이와 같이 작성된 Indicator를 사용하여 조사관은 침해 사고 발생 시 DFIOC에

수집된 Evidence와 기존에 작성된 Indicator를 비교 분석하여 빠르게 유사 침해 사고 여부를 파악할 수 있다.

4. DFIOC를 활용한 침해 사고 조사

이 장에서는 DFIOC 포맷의 활용을 확인하기 위하여 한 가지 시나리오를 가정하고 DFIOC 포맷의 적용과 활용에 대해 논의하고자 한다. 아래는 악성코드에 감염된 침해 시스템의 시나리오이다.

* 악성코드에 시스템이 감염
* 악성코드에 감염된 USB 메모리 연결을 통해 감염
* 악성코드는 시작프로그램으로 등록
* 악성코드 동작 중 8888 포트를 오픈

4.1 침해 시스템으로부터 Evidence 수집

시나리오의 침해 시스템으로부터 Evidence 정보를 수집한다면 USB 연결 흔적, 악성 프로그램 실행 흔적, 프로세스 동작 및 포트 오픈 상태가 수집될 것이다. Fig. 4는 침해시스템에서 수집한 정보를 DFIOC 포맷으로 기록한 내용이다. USB 메모리 연결이 이루어지면 외부저장장치 연결 흔적이 수집되므로 ExternalDriveInfo에 USB 메모리 연결 흔적이 기록된다. 또한 USB 메모리로부터 악성 프로그램에 감염되었으므로 외부 저장장치 경로에서 프로그램의 실행 흔적과 감염된 시스템에서 악성 프로그램이 실행된 흔적이 AMCache에 남게 된다. 또한 프로세스가 동작 중이며 네트워크가 활성화되어 있으므로 NetworkActivities 항목과 ProcessInfo 항목에 악성 프로그램의 동작 상태가 기록된다.

4.2 DFIOC Forensic Analysis 기록

침해 시스템에서 수집한 내용을 기반으로 포렌식 분석을 진행할 수 있으며 분석된 내용을 DFIOC 포맷의 Forensic Analaysis 항목에 기록할 수 있다. Fig. 5는 DFIOC Forensic Analaysis 항목에 분석 정보를 기록한 내용이다. 각각의 Evidence의 ID를 참조 정보로 사용하여 Evidence 간의 관계를 기록하고 있다. DFIOC에 수집된 Evidence에서 AMCache는 프로그램의 최초 실행 흔적을 확인할 수 있다. AMCache에는 C드라이브의 Malware.exe 프로그램이 처음 실행된 시간(AMCache-0000-00000-000001)과 E 드라이브의 Unknown1.exe 프로그램의 최초 실행 시간(AMCache-0000-00000-000000)이 인접한 시간에 실행된 것이 확인되므로 이를 통해 AMCache에 남겨진 기록이 서로 연관된 기록임을 알 수 있으며 두 개의 Evidence를 관련된 Evidence로 판단하여 “Log A related with B”로 기록할 수 있다. 또한 Unknown1.exe 프

```
<EvidencePackage> <Evidence>
        <NetworkActivities>
                <NetworkSessionList>  <NeworkSession id ="NetworkSession-0000-000000-000000">
                        <Process IDREF = "Process-0000-000000-000000"/>
                        <LocalPort>     8888 </LocalPort>
                </NeworkSession>  </NetworkSessionList>
        </NetworkActivities>
        <ProcessInfo><!-- 감염되어 동작중인 파일-->
                <Process ID="Process-0000-000000-000000"> <PEFile IDREF = "ExecutableFile-0000-000000-000000"/>
                <PortList> <NetworkSession IDREF="NetworkSession-0000-000000-000000"/> </PortList>
                </Process>
        </ProcessInfo>
        <UserSystemActivities>
                <AMCacheInfo>
                        <AMCache ID ="AMCache-0000-000000-000000"> <!-- USB 저장 파일이 실행 된 시각 -->
                                <FullPath>E:₩₩Utils₩Unknown1.exe</FullPath>
                                <FileSize>25088</FileSize>
                                <FirstRunTime>2014-12-04T12:25:30</FirstRunTime>
                        </AMCache>
                        <AMCache ID ="AMCache-0000-000000-000001"> <!-- USB 저장 파일이 실행 후 바로 실행된 파일 시각 -->
                                <FullPath>C:₩ProgramData₩Microsoft₩Windows₩Start Menu₩Programs₩StartUp₩Malware.exe</FullPath>
                                <FileSize>25088</FileSize>
                                <FirstRunTime>2014-12-04T12:25:35</FirstRunTime>
                                <File IDREF = "ExecutableFile-0000-000000-000000"/>
                        </AMCache>
                </AMCacheInfo>
                <ExternalDriveInfo>
                        <ExternalDrive ID = "ExternalDrive-0000-000000-0000000">
                                <DevideName>Mal USB</DevideName>
                                <DriveLetter>E</DriveLetter>
                                <SerialNumber>"07000708452117208D40"</SerialNumber>
                                <FirstConnectTime>2012-06-04T10:15:30</FirstConnectTime>
                        </ExternalDrive>
                </ExternalDriveInfo>
        </UserSystemActivities>
        <EvidenceFiles> <EvidenceFile>  <ExecutableFile ID = "ExecutableFile-0000-000000-000000">
                <File>
                        <FileName>      Malware.exe     </FileName>
                        <FilePath>      C:₩ProgramData₩Microsoft₩Windows₩Start Menu₩Programs₩StartUp₩Malware.exe </FilePath>
                        <FileHashValue> <SHA1Value>     1234567890ABCDEF01234567890ABCDEF0123456 </SHA1Value> </FileHashValue>
                </File>
        </ExecutableFile> </EvidenceFile>       </EvidenceFiles>
</Evidence> </EvidencePackage>
```

Fig. 4. Collected Forensic Artifact from Incident System

```
<EvidenceAnalysis>
  <AnalysisTitle>  침해 시스템 수사 </AnalysisTitle>
  <Analyst>
        <Name> 홍길동 </Name>
        <Role> 수사관 </Role>
  </Analyst>
  <StartDateTime> 2014-04-11T13:00:00 </StartDateTime>
  <EndDateTime> 2014-04-12T13:00:00 </EndDateTime>
  <CommentAndRelation>
    <EvidenceA> AMCache-0000-000000-000000  </EvidenceA>
    <EvidenceB> AMCache-0000-000000-000001  </EvidenceB>
    <EvidenceRelation> Log A related with B </EvidenceRelation>
    <Comment> A는 B에 의해 실행된 실행 흔적 </Comment>
  </CommentAndRelation>
  <CommentAndRelation>
    <EvidenceA> ExecutableFile-0000-000000-000001 </EvidenceA>
    <EvidenceB> ExternalDrive-0000-000000-000000  </EvidenceB>
    <EvidenceRelation>  A induced from B  </EvidenceRelation>
    <Comment> A(Malware.exe)는 B(Mal USB)로부터 유입된 파일 </Comment>
  </CommentAndRelation>
</EvidenceAnalysis>
```

Fig. 5. Forensic Analysis from DFIOC Evidence

로그램은 Mal USB(ExternalDrive-0000-00000-000000)의 연결 시간에 바로 실행되었으며, Unknown1.exe이 실행된 시간과 인접한 시간에 Malware.exe(ExecutableFile-0000-00000-000000)이 최초로 실행되었다. 이 정보를 바탕으로 Malware.exe는 Unknown1.exe 프로그램이 실행되었을 때 생성되었으며 Mal USB를 통해 유입된 것을 알 수 있다. 따라서 이 정보를 표현하기 위하여 Malware.exe 프로그램의 Evidence 정보와 Mal USB 메모리 연결 정보 Evidence를 “A induced from B” 관계로 기록할 수 있다.

4.3 DFIOC Indicator 작성

DFIOC는 분석된 내용을 바탕으로 침해 흔적의 특징을 추출하여 Indicator 항목에 침해 지표로 침해 탐지 시그니처를 만들 수 있다. 침해 지표로 삼을 수 있는 첫 번째 특징은 시작프로그램으로 등록된 악성코드이다. 시작프로그램에 존재하는 악성코드의 해시 값을 통하여 동일한 침해 사고 여부를 판단할 수 있다. 두 번째로는 시작프로그램에서 실행된 프로세스가 특정 포트를 열고 있는 것으로 침해 흔적을 삼을 수 있다. Fig. 6은 확인된 침해 흔적을 Indicator로 기록한 그림이다. 두 가지 특징 모두 자동화된 탐지가 가능하기 때문에 Indicatorcategory 속성에 Insepectable로 표시하여 자동화된 탐지가 가능함을 표기하였으며, 해시 값의 경우 침해 지표로 가장 확실한 정보이므로 가중치(weight)를 100으로 높게 표기하였다. 프로세스와 네트워크 정보 조합을 통한 판단은 두

```
<Indicator composition="OR" >
       <Indicator composition="AND" weight=60 IndicatorCategory="inspectable">
              <KeyEvidence>
                     <Content  Condition = "Contains"><!--시작프로그램에서 실행된 악성코드-->
                            <EvidencePath> "Evidence/ProcessInfo/Process/PEFile/File/FilePath"      </EvidencePath>
                            <ContentValue>"Start Menu"</ContentValue>
                     </Content>
                     <InernalReferenceEvidence IDREF ="Process-0000-000000-000000">
              </KeyEvidence>
              <KeyEvidence>
                     <Content Condition = "Equals"><!--프로세스에서 동작중인 네트워크 행위-->
                            <EvidencePath>  "Evidence/ProcessInfo/Process/PortList/NetworkSession/LocalPort" </EvidencePath>
                     </Content>
                     <ContentValue>"8888"</ContentValue>
                     <InernalReferenceEvidence IDREF ="Process-0000-000000-000000">
              </KeyEvidence>
       </Indicator>
       <Indicator weight=100 IndicatorCategory="inspectable">
              <KeyEvidence><!--현재 실행되고 있는 프로세스의 파일 정보-->
                     <Content condition="Equals" Type="Specific" IsRegEX="No">
                            <EvidencePath>
                                   "Evidence/ProcessInfo/Process/PEFile/File/FileHashValue/SHA1Value"
                            </EvidencePath>
                            <ContentValue>
                                   "1234567890ABCDEF01234567890ABCDEF0123456"
                            </ContentValue>
                     </Content>
                     <InernalReferenceEvidence IDREF ="Process-0000-000000-000000">
              </KeyEvidence>
       </Indicator>
</Indicator>
```

Fig. 6. DFIOC Indicator From Incident System

정보가 동시에 나타나야 하므로 AND 조합으로 기입하였으며 오탐 가능성이 있으므로 가중치는 60으로 낮게 표기하였다. 또한 해시 Indicator와 프로세스와 네트워크 정보의 Indicator는 둘 중 하나만 탐지되더라도 침해 시스템으로 의심이 가능하므로 두 개의 Indicator를 OR로 조합하였다. 작성된 Indicator는 다른 침해 의심 시스템을 조사해야 할 때 동일한 흔적이 존재하는지 비교 분석함으로써 침해 의심 시스템을 빠르게 분석할 수 있다.

지금까지 악성코드 감염 시나리오를 가정하여 DFIOC 포맷의 Evidence 항목에 수집된 정보를 수집하고, 수집된 정보를 바탕으로 분석하여 분석 결과를 Forensic Analysis 항목에 기록하였으며, 침해 지표를 Indicator 항목에 작성하는 과정을 확인하였다. XML 기반으로 작성되었기 때문에 수집된 정보를 혼동 없이 정확히 표현할 수 있었으며, XML 하나의 포맷 상에서 포렌식 분석 결과와 침해 흔적을 작성하였기 때문에 불필요한 정보 가공 없이 Evidence의 ID를 참조하는 것만으로 분석 내용을 정리할 수 있었으며, 침해 탐지 시그니처를 작성할 수 있었다.

5. 결론 및 향후 과제

침해 시스템 조사는 다양한 포렌식 아티팩트를 수집하고 포렌식 도구를 사용하여 분석하며, 포렌식 도구에서 출력한 정보를 활용하여 분석이 이루어진다. 하지만 서로 다른 양식으로 출력되는 포렌식 아티팩트 분석 정보는 서로 연관관계 분석이 어렵기 때문에 동일한 양식으로 정보의 가공이 필요하다. 이와 같은 불필요한 작업은 포렌식 아티팩트 수집, 분석 결과 작성 전반에 걸쳐 발생하게 되며, 조사할 데이터의 양이 많을수록 조사관에게 더욱 더 많은 부하가 가해지게 된다. 이 같은 문제를 해결하기 위하여 표준화된 포렌식 아티팩트 정보의 작성 방안으로 Digital Forensic Indicators Of Compromise(DFIOC) 포맷을 개발하고 소개하였다.

DFIOC 포맷은 침해 시스템에서 수집 가능한 모든 정보를 하나의 XML로 통합하여 분석함으로써 침해 시스템에 존재하는 다양한 데이터간의 상호 연관 관계를 보다 쉽게 분석할 수 있도록 구성되었다. 또한 침해 사고 분석이 완료된 후 분석된 내용을 Forensic Analysis에 기록하고, 침해 흔적을 Indicator로 기술할 수 있다. 작성된 Indicator는 다른 침해 사고에서 기존과 유사한 침해 사고인지 빠르게 분석하여 판단할 수 있는 침해 지표로써 활용할 수 있다.

DFIOC 포맷을 사용함으로써 침해 시스템의 다양한 정보를 통합된 환경에서 분석할 수 있는 기반이 마련된다. 하지만 현재 DFIOC를 지원할 수 있는 도구의 부재로 인해 DFIOC의 활용이 어렵다. 따라서 기존의 침해 사고 분석도구를 DFIOC의 포맷에 적용하기 위한 연동 도구가 필요하다. 또한 수집된 정보를 사용자가 손쉽게 조작하여 데이터를 확인할 수 있도록 DFIOC 포맷을 활용한 분석 환경 개발과 Indicator 기반의 침해 사고 분석 도구 개발이 향후 과제로 남아 있다.

References

[1] Alessandro Guarino, “Digital Forensics as a Big Data Challenge,” *StudioAG, ISSE 2013 Securing Electronic Business Processes*, Vol.6, pp.197-203, 2013.

[2] Yinghua Guo, Jill Slay, and Jason Beckett, “Validation and verification of computer forensic software, toolsdSearching Function,” *Digital Investigation*, Vol.6, pp.S12-S22, Sep., 2009.

[3] Karen Kent, Suzanne Chevalier, Tim Grance, and Hung Dang, “Guide to Integrating Forensic Techniques into Incident Response,” NIST SP800-86 Notes, Aug., 2006.

[4] MITRE [Internet], https://cyboxproject.github.io.

[5] Eoghan Casey, Greg Back, and Sean Barnum, “Leveraging CybOX to standardize representation and exchange of digital forensic information,” *Digital Investication*, Vol.12, pp.102-110, Mar., 2015.

[6] Eoghan Casey, Greg Back, Sean Barnum [Internet], https://github.com/DFAX/dfax.

[7] Mandiant [Internet], http://www.openioc.org.

[8] Simson Garfinkel, “Digital forensics XML and the DFXML toolset,” Vol.8. pp.161-174. Feb., 2012.

[9] Simson Garfinkel [Internet], https://github.com/simsong/dfxml.

[10] Stephen Larson, “Book Review: The Basics of Digital Forensics: The Primer For Getting Started in Digital Forensics,” *Journal of Digital Forensics, Security and Law*, Vol.9, No.1, pp.83-85, 2014.

이 민 욱

e-mail : zwmin1616@korea.ac.kr

2014년 경기대학교 전자공학과(학사)

2014년～현 재 고려대학교 정보보호학과 석사과정

관심분야 : Digital Forensic, Reverse Engineering

윤 종 성

e-mail : yoonjs53@naver.com

2005년 공군사관학교 전산과학과(학사)

2013년～현 재 고려대학교 정보보호학과 석・박사통합과정

관심분야 : Digital Forensic, Information Security

이 상 진

e-mail : sangjin@korea.ac.kr

1987년 고려대학교 수학과(학사)
1989년 고려대학교 수학과(석사)
1994년 고려대학교 수학과(박사)
1989년~1999년 ETRI 선임연구원
1999년~현 재 고려대학교 교수
2008년~현 재 고려대학교 디지털포렌식연구센터 센터장
관심분야 : Digital Forensic, Steganography, Hash Function