Journal of the KIMST, Vol. 26, No. 2, pp. 179-187, 2023
ISSN 1598-9127(Print) · 2636-0640(Online)
DOI https://doi.org/10.9766/KIMST.2023.26.2.179

Research Paper 정보·통신 부문

사이버공격 추적시스템 운용아키텍처

안 재 홍*,1)

1) 국방과학연구소 국방첨단과학기술연구원 사이버기술센터

Cyberattack Tracing System Operational Architecture

Ahn, Jae-hong*,1)

1) *Defense Cyber Technology Center, Agency for Defense Development, Korea*

(Received 5 October 2022 / Revised 9 February 2023 / Accepted 27 February 2023)

Abstract

APT cyber attacks have been a problem for over a past decade, but still remain a challenge today as attackers use more sophisticated techniques and the number of objects to be protected increases. 'Cyberattack Tracing System' allows analysts to find undetected attack codes that penetrated and hid in enterprises, and to investigate their lateral movement propagation activities. The enterprise is characterized by multiple networks and mass hosts (PCs/servers). This paper presents a data processing procedure that collects event data, generates a temporally and spatially extended provenance graph and cyberattack tracing paths. In each data process procedure phases, system design considerations are suggested. With reflecting the data processing procedure and the characteristics of enterprise environment, an operational architecture for CyberAttack Tracing System is presented. The operational architecture will be lead to the detailed design of the system.

Key Words : Cyber Attack Trace, APT, Enterprise, Provenance Graph

1. 서 론

APT(Advanced Persistent Threat) 공격은 엔터프라이즈 내부 네트워크 보안에 심각한 위협이다. 네트워크 장비나 컴퓨터에서 백신과 같은 시그니처(signature) 기반으로 악성코드를 탐지하고 차단하는 기술과 EDR (Endpoint Detection & Response)처럼 호스트 이벤트 분석을 기반으로 알려진 악성행위를 식별하거나 몇 가지 행위 간 관계로 악성행위를 추론하는 방법으로 탐지 및 차단하는 노력을 많이 하고 있지만, 엔터프라이즈로의 침투는 여전히 발생할 수 있다. 침투된 악성코드는 내부 네트워크를 통해 전파되어 확산하고, 최종 공격 목적을 달성하기 전까지 지속적인 위협으로 은거한다. 따라서 침투한 사이버 위협으로 인해 내부 시스템이 마비, 파괴, 유출, 변조되는 피해를 받기 전에 반드시 이를 발견해야 한다.

사이버공격 추적시스템은 탐지하지 못하고 엔터프

* Corresponding author, E-mail: addseman@add.re.kr

라이즈에 침투 후 은거하고 있는 공격코드를 발견하고, 그 공격코드의 내부 전파경로를 조사하는 시스템이다. 이 시스템은 엔터프라이즈 내부 호스트(PC, 서버)에서 발생하는 프로세스를 중심으로 이벤트를 수집하여, 행위 간 종속관계를 표현하는 프로비넌스 그래프(provenance graph)를 생성하고, 이 프로비넌스 그래프에서 위협이 되는 악성코드를 발견하고, 호스트 간 전파경로와 호스트 내에서 실행한 일련의 공격 의심행위를 식별한다.

사이버공격 추적시스템과 EDR은 호스트에서 실행 이벤트를 수집하여 악성행위를 분석하는 점에서 공통점이 있다. 그러나 ERD은 단일 호스트에서 이미 알려진 행위를 수집·저장·분석하여 실시간 또는 몇십 초 수준의 근 실시간으로 탐지하고 대응하는 데 중점이 있고, 사이버공격 추적시스템은 다단계 네트워크로 구성된 엔터프라이즈의 모든 호스트를 대상으로 이미 침투한 사이버 위협의 존재 여부를 확인하고 공격피해 범위 및 전파경로를 조사분석하는 과정에서 활용한다는 점에서 차별점이 있다. 이외 사이버공격 추적시스템은 침투에 성공한 공격자가 자신의 행위를 숨기기 위해 파일명 변조 또는 레지스트리 정보 삭제 등의 행위도 조사분석 과정에서 발견할 수 있게 한다.

본 논문에서는 사이버공격을 추적할 수 있는 이벤트 데이터를 각 호스트에서 수집하고, 이를 시간적 공간적으로 확대된 프로비넌스 그래프로 변환하여 중앙서버에 저장하고, 공격경로를 분석하는 데이터 처리 절차를 제시한다. 그리고 이 절차와 엔터프라이즈 환경 특성을 반영하여, 사이버공격 추적시스템 운용아키텍처를 제시한다.

논문의 구성은 2장 관련 연구에서 공격 탐지 및 의심행위를 식별하는데 널리 활용하는 공격행위 탐지 프레임워크와 이벤트 로그의 종속관계를 표현하는 프로비넌스 그래프 개념, 장기간 연구개발하고 있는 프로비넌스 그래프 처리 공개SW 기능 그리고 엔터프라이즈 환경 특성에 대해 살펴본다. 3장에서는 프로비넌스 그래프 데이터 처리절차와 사이버공격 추적시스템 운용아키텍처를 제시한다. 4장에서는 제시한 운용아키텍처의 유효성에 대한 논의와 사이버공격 추적시스템의 보안 활용 가능 분야를 제시 후 5장 결론으로 마무리한다.

2. 관련연구

MITRE ATT&CK 프레임워크는 전세계에서 관측한 공격 행위 패턴을 전술, 기법, 절차(TTPs)로 분류하여 정리한 지식베이스이다. 이것은 엔터프라이즈 공격을 300개 이상의 서브기법으로 세분화하여 정의하고 있다. 이 지식베이스는 민간과 공공 부문을 비롯한 사이버 보안 제품과 서비스를 위한 위협 모델과 방법론의 기반으로 활용된다[1]. 가트너 그룹 조사에 의하면 EDR (Endpoint Detection & Response) 상용 제품 중 상위 10개가 이 프레임워크를 활용할 정도로 세계적으로 보편화된 공격행위 모델이다[2]. 또한 ATT&CK CAR(Cyber Analytic Repository)는 공격 모델을 기반으로 개발된 분석 지식베이스이다. 이는 공격에 대한 가설적 설명, 호스트, 네트워크, 프로세스 관련 분석 정보, 그리고 참조할 전술과 기법, 용어, 분석 자동화 구현을 위한 슈도코드, 분석을 실행할 수 있는 단위시험 정보를 포함하고 있다[3]. 이를 기반으로 공격 의심행위 분석에 유용하게 활용할 수 있다.

프로비넌스 그래프(Provenance Graph)는 데이터 보안 모니터링에 효과적인 접근법이다[4]. 이를 이용하여 엔터프라이즈 내 사이버공격 전파경로와 유입 원천을 식별 가능하게 한다. Yuanzaho Gao의 연구에 의하면 프로비넌스 그래프가 차세대 탐지 및 대응 매카니즘으로 잠재력을 가지고 있다고 하며 몇 가지 장점을 다음과 같이 제시한다[5]. 먼저, 이벤트를 프로비넌스 그래프로 만들어서 시스템 객체 간 상호관계를 표현함으로써 시스템 실행을 가시화한다. 두 번째로, 의미 인식과 견고한 탐지를 가능하게 하여 공격자가 로그 정보를 공간적 시간적으로 변조하기 어렵게 한다. 이외 풍부한 의미를 제공하여 분석가에게 더 효율적인 조사분석을 제공한다. 세 번째는 모든 실행 기록을 유지하여, 장기간 은닉해 있는 APT를 분석하게 한다. 이를 위해 시스템 실행 기록은 침입 유입 지점을 추적하고 영향을 이해하는 데 필수사항이다.

W3C(World Wide Web Consortium)는 프로비넌스 표준에 대해 그 핵심 요소 구성을 Fig. 1처럼 엔티티, 엑티비티, 에이전트로 정의한다[6]. 엔티티는 물리적, 디지털, 개념적 또는 다른 종류의 실체를 뜻하며, 엔티티는 서로 다른 속성을 갖는 것으로 설명될 수 있다. 엔티티의 인스턴스는 프로세스, 파일, 레지스트리 등이다. 엑티비티는 엔티티가 존재하게 되는 방법과 해당 속성이 새 엔티티가 되는지를 의미하며, 기존 엔

티티를 이용하여 새로운 엔티티를 생성하기도 한다. 액티비티의 인스턴스는 생성하다, 수정하다, 삭제하다, 종료하다 등이다. 에이전트는 액티비티와의 연관관계로, 액티비티에 의해 생성된 엔티티는 해당 에이전트에 속하게 된다. 에이전트의 인스턴스는 사용자, 실행 SW, 프로세스 등이다.

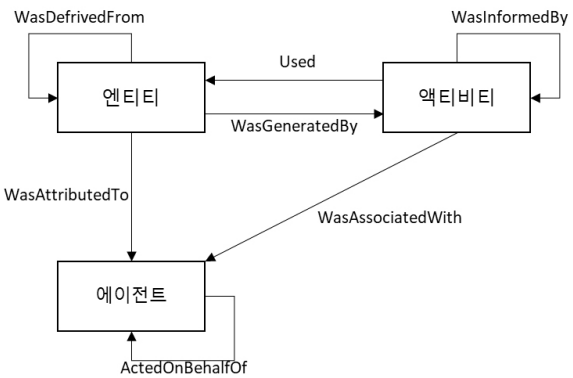

Fig. 1. A high level overview of the structure of PROV

Ashish Gehani의 연구 결과인 SPADE(Support for Provenance Auditing in Distributed Environments)는 여러 도메인에서 관리되는 컴퓨팅 그리드(grid) 환경에서 데이터 보안을 확인하는 방법에 대한 문제를 해결하기 위해 그리드 데이터의 프로비넌스를 수집, 인증 및 질의(query)하기 위한 분산 서비스로 2008년 SPADEv1을 개발하였다. 그리고 2010년 두 번째 버전 SPADEv2는 데이터 프로비넌스를 수집(생산), 관리(저장) 그리고 분석(사용) 기능을 분리하는 커널을 개발하여, 프로비넌스 생산자와 소비자 사이의 중간자 역할을 하게 하였다. 이 커널은 다양한 프로비넌스 소스(Windows, Linux, Mac OS X)로부터 입력되는 메타데이터를 버퍼링과 필터링하는 기능을 이용하여 프로비넌스 그래프 요소 스트림을 처리하고, 이벤트 데이터 요소들을 집계하고, 융합하며, 합성한다. 그리고 H2/MySQL, Neo4J 데이터베이스에 프로비넌스 데이터를 저장하며, DB 질의는 멀티프렉싱 기능을 이용하여 원격(remote)과 지역(local)에서 병렬적으로 처리할 수 있도록 지원하고 있다[7].

SPADE는 2016년에 빅 프로비넌스 그래프를 처리하기 위한 커널로 발전하였다. 이 커널은 대규모 프로비넌스 데이터를 다루고, 질의에 대한 응답을 동적으로 재작성하는 변환기(Transformers) 기능을 추가하였다[8]. 이 변환기기는 수신한 질의를 지역 클라이언트 저장소에 전달하고, 해당 지역 클라이언트들로부터 질의 결과를 받아 정보를 취합 후 그래프를 생성한다.

2021년 최신 SPADE 커널 아키텍처는 Fig. 2와 같다. 이 SPADE는 리포터 모듈, 필터 모듈, 저장 모듈, 분석 모듈로 나누어지며, 적용사례로 컴퓨터 운영체제 이벤트 프로비넌스 그래프를 이용하여 악성코드 감염을 추적하는 사례와 비트코인 거래 프로비넌스 그래프로 계정(account) 간 코인 송수신 흐름을 추적하는 사례를 제시하였다. 이 연구에서는 다양한 질의 구문을 정의하여 데이터베이스 종류와 독립적으로 그래프 버틱스(vertices)와 에지(edges), 경로 그리고 기원을 찾는데 편의성을 제공하였다[9].

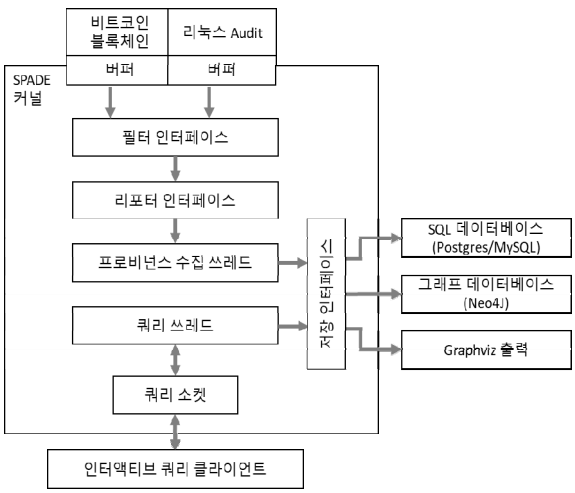

Fig. 2. SPADE kernel architecture

최도현의 “그래프 데이터베이스 기반 악성코드 행위 탐지 기법” 연구에 의하면 그래프 이론으로 공격 경로를 추적하는 연관관계 분석에 의미 있는 결과를 보여주었다. 이 연구에 의하면 악성 행위와 변종 공격 행위 탐지에 성능이 향상되었고, 관계형 DB 대비 9.84배 이상의 성능이 향상되었다[10]. 이 연구는 SPADE처럼 프로비넌스 그래프를 그래프 데이터베이스로 저장하여 추적경로를 분석에 활용한 추가 사례이다.

Xhang Xu의 연구에 의하면 한 호스트에서 발생하는 이벤트 데이터의 양은 1년에 0.5 ~ 1 GB이다. 실제 한 은행은 20만대의 호스트를 가지고 있으며 APT 공격은 평균 188일 간 내부에 잠복하므로 이를 분석하기 위해서는 6개월에서 1년의 이벤트 빅데이터를

수집하고 저장해야 한다[11].

사이버공격 추적시스템은 관련 연구 SPADE에서 제시한 프로비넌스 그래프 데이터 처리 기능처럼 호스트에서 이벤트 데이터를 수집하여, 프로비넌스 그래프로 통합 및 DB저장 후 질의를 통해 추적경로를 분석을 할 필요가 있으며, 공격 의심행위를 분석하기 위해서는 MITRE ATT&CK CAR 정보와 그래프 이론을 활용할 필요가 있다. 그리고 조직 구성이 다단계로 구성된 국방도 Xhang Xu[11]가 조사한 엔터프라이즈처럼 대량의 호스트로 구성되어 있으므로 다중 네트워크의 호스트에서 발생하는 이벤트를 처리하기 위해 확장성을 고려한 대용량 분산처리 시스템으로 운용아키텍처를 정립할 필요가 있다.

3. 추적시스템 운용아키텍처

사이버공격 추적시스템은 호스트에서 발생하는 이벤트 로그 데이터로 추적정보 분석에 활용하는 데이터 처리 시스템이다. 본 장에서는 추적에 필요한 이벤트 데이터를 호스트 시스템 계층에서 수집하고, 프로비넌스 그래프를 시간적 공간적으로 확대하여 생성하며, 이 프로비넌스 그래프로부터 침입한 악성코드의 위치를 찾아, 그 악성코드의 전파경로와 추가적인 의심행위 분석을 자동화하는 데이터 처리 절차와 각 단계별 설계 고려사항을 제시한다. 그리고 이 절차에 의해 다중 네트워크로 구성된 엔터프라이즈 환경 특성을 반영하고 사용자인 침해사고 조사분석가 관점의 운용개념을 제시한다.

3.1 프로비넌스(Provenance) 데이터 처리 절차

본 절에서는 추적에 필요한 이벤트 데이터를 호스트 시스템 계층에서 수집하고, 프로비넌스 그래프를 시간적 공간적으로 확대하여 생성하며, 이 프로비넌스 그래프로부터 침입한 악성코드의 위치를 찾아, 그 악성코드의 전파경로와 추가적인 의심행위 분석을 자동화하는 데이터 처리 절차와 각 단계별 설계 고려사항을 제시한다.

3.1.1 이벤트(Event) 수집

이벤트 수집은 호스트의 시스템 수준에서 발생하는 로그를 수집하는 기능이다. 수집한 이벤트는 경로추적 분석에 필요한 의미를 포함하면서 동시에 최소한의 양으로 수집해야 한다. 이벤트 수집 시 중요 고려사항은 다음과 같다.

첫째 전파경로 경로추적은 악성코드가 실행하며 발생한 프로세스가 파일, 네트워크, 레지스트리 그리고 외부장치에 행하는 일련의 이벤트 관계이다. 추적을 위해 호스트 계층에서 수집할 수 있는 이벤트 데이터 요소는 Table 1과 같다. 이 표는 에이전트(이벤트 행위주체), 엔티티(이벤트 행위객체), 액티비티(주체와 객체 간의 행위관계) 인스턴스를 명기하였다. 각 에이전트와 엔티티 요소는 속성으로 IP와 호스트명이 포함되어야 하며, 액티비티는 발생시간(TimeStamp)을 공통 속성으로 수집한다. 파일 엔티티의 속성은 해시값과 저장 디렉토리이다. 이외 필요한 데이터 요소는 ATT&CK 프레임워크에서 제공하는 데이터요소를 참고하여 추가할 수 있다.

Table 1. Primary event data elements

에이전트 (행위주체)	액티비티 (관계)	엔티티 (행위객체)
사용자	생성, 종료	프로세스
실행파일	생성, 종료	프로세스
프로세스	생성, 종료, 호출	프로세스
	생성, 읽기, 쓰기, 삭제, 이름변경	파일
	생성, 쓰기, 삭제	레지스트리
	접속, 송신, 수신, 접속종료	네트워크
	접속, 접속종료	외부장치

두 번째로, OS 커널 내에서 이벤트 데이터를 직접 수집한다. ETW(Event Trace for Windows) 또는 Syslog 같이 시스템 운영체제에서 제공하는 Audit 기능을 이용하여 거칠게(coarse-grained) 수집하는 방법보다 OS 커널 내에서 이벤트를 직접 수집하면 처리 시간과 스토리지 용량 측면에서 더 나은 성능을 기대할 수 있다[12]. 그리고 프로세스 이벤트 정보를 수집 후 이와 관련된 이벤트를 추가 수집할 때 관련 프로세스가 이미 제거되어 추가 이벤트를 수집할 수 없는 경우가 발생하는데 OS 커널 내에서 정교한(fine-grained) 수집을 할 경우는 순간적으로 생성 후 삭제된 프로세스

이벤트도 수집이 가능하며 조사분석 단계에서 공격행위를 분석하기 위해서는 이러한 이벤트를 수집하고 저장할 필요가 있다.

세 번째는 수집한 이벤트가 MITRE ATT&CK TTPs CAR(Cyber Analytic Repository) 기준의 규칙(rule)과 일치 여부를 확인 후 수집 단계에서 공격 의심행위를 식별하여 이벤트 데이터 요소에 포함시킨다. 예를 들어 공격자는 공격코드 파일을 엔터프라이즈 내부 다른 호스트로 복사할 것이며 이를 위해 SMB(Server Messaged Block)나 RDP(Remote Desktop Protocol)를 이용하여 유효한 계정으로 원격접속을 시도할 것이다. 이런 의심행위를 식별하는 방법으로 powershell 사용 여부를 확인할 수 있다. 악의적 파워쉘 사용 여부 확인은 네트워크 전파행위 이벤트 발생 후 profile.ps1 파일을 *$PsHome이나 *$PsHome\ MyDocuments\PowerShell 같이 시스템에서 사용하는 특정 디렉토리 외에 다른 위치에 저장하는 이벤트를 의심행위로 식별할 수 있다. 이외 실행, 취약점 스캔, 자격증명, 로그삭제 등의 의심행위를 MITRE ATT&CK CAR에서 제공하는 규칙을 기반으로 식별한다.

이벤트 수집 단계에서 설계 고려사항 요약은 다음과 같다.

- 사이버공격의 주요 행위인 프로세스, 파일, 레지스트리, 네트워크 및 외부장치 관련 행위 이벤트를 수집
- 호스트 성능 부담 감소를 위해 OS 커널 드라이브 기반으로 정교하게 이벤트 데이터 수집
- 추적시스템 부하 분산 차원에서 호스트에서 이벤트 수집 단계에서 규칙(rule) 기반 의심행위 식별

3.1.2 프로비넌스 그래프(Provenance Graph) 생성

프로비넌스 그래프는 수집한 이벤트를 이용하여 행위 종속관계를 표현한 그래프로 사이버 위협 존재 여부 및 유입 전파경로를 식별하는 데 활용한다. 프로비넌스 그래프는 버틱스와 방향성을 가진 에지로 구성한다. 이벤트에서 행위 주체인 에이전트와 행위 객체인 엔티티를 그래프의 버틱스로 정의하고, 두 버틱스 관계인 엑티비티를 에지로 정의한다. 본 논문에서는 다중네트워크로 구성된 엔터프라이즈 환경 특성을 고려하여 프로비넌스 그래프를 '호스트 프로비넌스 그래프'와 '시공간 확장 프로비넌스 그래프'로 구분한다.

'호스트 프로비넌스 그래프'는 Fig. 3처럼 한 호스트가 전원을 한 번 켜서 운용하고 전원을 끌 때까지 발생한 이벤트 간 관계를 기록한 데이터이다. '시공간 확장 프로비넌스 그래프'는 Fig. 4처럼 하나의 호스트 프로비넌스 그래프가 과거에 생성한 자신의 과거 호스트 프로비넌스 그래프와의 관계를 연결하여 시간적으로 확장하고, 호스트 간 통신행위 종속관계를 연결하여 공간적으로도 확장한 프로비넌스 그래프이다. 프로비넌스 그래프 생성시 고려사항은 다음과 같다.

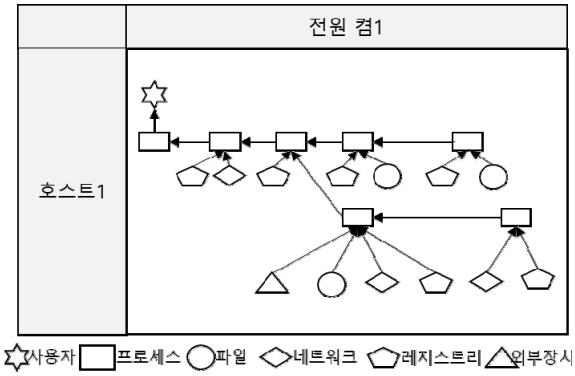

Fig. 3. Example of a host provenance graph

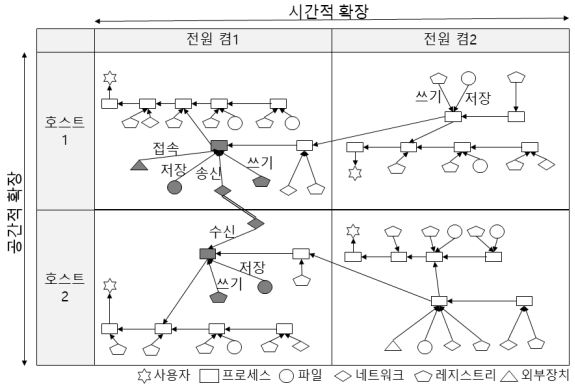

Fig. 4. Example of a spatio-temporal expansion provenance graph

첫째, '호스트 프로비넌스 그래프'는 프로세스가 다른 프로세스를 생성, 호출, 종료하는 관계를 시간 순서로 연결한다. 그리고 각 프로세스가 레지스트리, 파일, 네트워크, 외부장치에 행한 이벤트를 링크로 연결한다.

두 번째, 프로비넌스 그래프를 시간적으로 확장하기 위한 고려사항이다. 호스트를 재부팅한 경우, 과거 프로세스와 새로 부팅 후 프로세스가 동일한 실행파일

에서 생성되면 이 두 프로세스를 동일한 프로세스로 연결시킨다. 그러나 시스템에서 자동 부여하는 프로세스ID는 동일 실행파일임에도 시스템 부팅시 다른 프로세스ID로 부여하기 때문에 동일 프로세스로 식별하지 못한다. 따라서 동일 실행파일에 의한 동일 프로세스로 식별할 수 있는 별도의 ID를 부여하고 유지하여, 별도 부여된 프로세스 ID가 동일한 경우 동일 프로세스로 연결하여 프로비넌스 그래프를 시간적으로 확장한다.

세 번째는 프로비넌스 그래프를 공간적으로 확장하기 위한 고려사항이다. 악성코드를 다른 호스트로 전파한 행위를 식별하기 위해 네트워크 이벤트를 이용한다. 예를 들어 호스트1 네트워크 이벤트의 송신 ip, 송신 port, 송신파일 속성 값과 호스트2 네트워크 이벤트 수신 ip, 수신 port, 송신파일 속성 값이 동일할 때, 양쪽 호스트 네트워크 이벤트를 연결하여 프로비넌스 그래프를 공간적으로 확장한다.

네 번째, 프로비넌스 그래프 저장을 JSON 형태의 문서DB와 그래프DB로 저장한다. RDB의 경우 빅데이터 처리 및 확장성에 제한이 있고, 요구사항 변경시 스키마를 변경해야 하는 어려움이 있다. 그러나 JSON 형태 문서DB는 스키마 없이 <키, 값> 만을 저장하므로 수집 및 분석 요구사항 변경에도 스키마를 변경하지 않아도 되며, 데이터 양이 빅데이터로 증가하여도 클러스터링을 통한 확장성(scale out)이 용이하며 텍스트 검색에 유리하다. 그러나 이벤트간 관계가 복잡하게 연결될 경우 문서형 DB는 노드와 링크 간 다양한 관계성 분석 알고리즘 개발에 많은 노력을 소모해야 한다. 반면 그래프DB은 기본 제공 기능으로 버틱스와 에지 간의 관계성을 상대적으로 용이하게 검색하여 분석할 수 있다. 따라서 그래프DB는 추적경로 생성을 위한 버틱스와 에지 간 관계성 분석에 활용을 하며, 세부적인 텍스트 검색은 문서형DB를 사용하여 상호 보완적으로 활용할 경우 전파경로와 의심행위 분석성능 향상을 기대할 수 있다.

프로비넌스 그래프 생성 단계에서 설계 고려사항 요약은 다음과 같다.

- 호스트 프로비넌스 그래프는 프로세스 간 액티비티 관계를 시간 순으로 연결하고 이외 엔티티는 연관된 프로세스를 중심으로 액티비티 관계를 연결
- 프로비넌스 그래프의 시간적 확장은 프로세스에 별도의 ID를 부여하여, 동일한 실행파일에서 발생한 과거 프로세스와 컴퓨터 재부팅 후 현재 프로세스를 연결
- 프로비넌스 그래프의 공간적 확장은 네트워크 송신 호스트와 수신 호스트의 IP, port, 파일이 동일한 경우 두 호스트 간 네트워크 이벤트를 연결
- 추적경로 생성 기능 구현 부담 경감 및 및 분석 정보 검색 효율성 제고를 위해 프로비넌스 그래프 데이터는 JSON 파일 형태의 문서형 DB와 그래프DB에 저장

3.1.3 추적경로 생성

추적경로 생성은 '시공간 확장 프로비넌스 그래프'를 분석하여 악성코드의 존재 여부를 확인하고 그 유입 전파경로를 식별하는 기능이다. Fig. 4 예시는 호스트1에서 외부장치 접속 후 파일이 저장되고 이를 호스트2로 전송 후 레지스트리에 쓰기를 하는 행위 이벤트가 존재함을 보여준다. 호스트2는 호스트1에서 수신한 파일을 저장하고 레지스트리에 쓰기를 하는 과정도 프로비넌스 그래프를 통해서 분석할 수 있다. 이때 송수신한 파일이 악성코드로 확인된 경우 이를 전파경로의 일부로 생성한다. 추적경로 생성 설계 고려사항은 다음과 같다.

첫째, 엔터프라이즈 내부에 악성코드의 존재 여부는 이 코드의 해쉬값이 프로비넌스 그래프에 기록된 파일의 해시값을 비교하여 확인한다.

둘째, 존재가 확인된 위협은 프로비넌스 그래프에서 해당 파일을 생성한 프로세스 노드를 중심으로 연관된 프로세스 간 에지를 역방향과 순방향으로 이동하며 유입원점부터 가장 최근 행위까지 수행한 버틱스를 식별하여 전체 호스트 간 전파경로를 식별한다.

세 번째, 그래프 패턴 유사도 검사와 그래프 이론에 의한 알고리즘과 MITRE ATT&CK CAR 규칙으로 의심행위를 식별할 수 있다.

추적경로 생성 단계에서 설계 고려사항 요약은 다음과 같다.

- 엔터프라이즈 내 악성코드 침투 여부 확인은 파일 해시값으로 검색
- 전파경로 생성은 발견한 악성코드 파일를 생성한 프로세스 버틱스를 중심으로 연결된 에지의 역방향과 순방향으로 해당 파일을 송신 또는 수신한 일련의 호스트와 해당 버틱스와 에지를 검색
- 그래프 이론과 룰(rule) 기반으로 의심행위 식별

3.2 운용아키텍처(Operational Architecture)

시간적 공간적으로 확대된 대용량의 엔터프라이즈 프로비넌스 그래프를 하나의 서버에서 생성하기에는 HW 자원 성능에 큰 부담이 된다. 따라서 다중 네트워크로 구성된 엔터프라이즈에 포함된 서브네트워크 중계서버와 호스트 자원을 최대한 활용하여 분산처리하여 엔터프라이즈 분석서버 부하를 최소한으로 한다. 운용아키텍처는 국방아키텍처 프레임워크 v1.5 산출물 템플릿인 OV-1 운용개념도와 OV-2 운용노드연결기술서로 표현한다[13].

Fig. 5는 운용노드연결기술서(OV-2)로 운용노드를 '호스트', '서브네트워크 전처리서버', '엔터프라이즈 분석서버' 3가지로 분류한다.

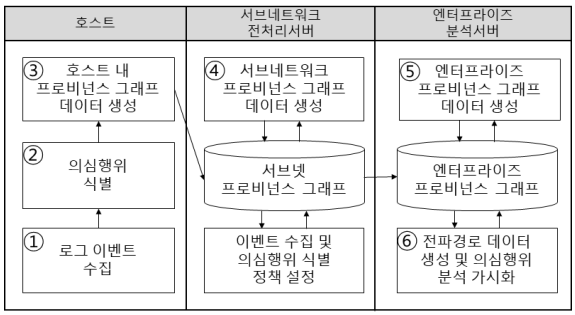

Fig. 5. Cyberattack tracing system operational resource flow description(OV-2)

첫 번째 운용노드인 '호스트'에서는 OS 시스템 계층에서 발생하는 이벤트를 커널 드라이브 기반으로 정교하게 수집하고, MITRE ATT&CK CAR에서 정의한 식별 규칙을 기준으로 중요 의심행위를 선별하여 이벤트 데이터에 추가한다. 그리고 컴퓨터 부팅 후 전원을 끄기까지 발행한 이벤트 간 관계를 연결하여 '호스트 프로비넌스 그래프' 정보를 생성 후 '서브네트워크 전처리서버'로 전송한다.

두 번째 운용노드 '서브네트워크 전처리서버'에서는 각 호스트에서 생성한 '호스트 프로비넌스 그래프'로 동일 호스트 내 과거 프로비넌스 그래프를 별도로 부여된 프로세스 ID를 기준으로 통합하고, 동일 IP와 Port로 송수신한 이벤트를 기준으로 서브네트워크 내 호스트 간 네트워크 이벤트를 연결하여 시간적으로 공간적으로 확대된 '서브네트워크 프로비넌스 그래프'를 생성한다. 생성한 데이터는 '엔터프라이즈 분석서버'로 전송한다. 또한 부가적으로 엔터프라이즈 분석서버로부터 호스트 수집 및 의심행위 식별 정책을 중계하는 운용활동을 포함한다.

세 번째 운용노드 '엔터프라이즈 분석서버'는 각 '서브네트워크 프로비넌스 그래프'의 과거 그래프와 연결을 하고, 서브네트워크 간 네트워크 이벤트를 발생한 호스트 내 버틱스를 연결하여 '엔터프라이즈 프로비넌스 그래프'로 통합한다.

Fig. 6 사이버공격 추적시스템 운용개념도는 여러 개의 서브네트워크로 이루어진 다중네트워크와 각 서브네트워크를 구성하는 다수의 호스트들로 구성된 엔터프라이즈 네트워크 환경을 보여주고 있다. 그리고 운용노드인 호스트와 서브네트워크 전처리서버 그리고 엔터프라이즈 분석서버의 설치 위치와 각 서버의 데이터를 처리하는 기능을 보여주고 있다. 또한 침해사고 조사분석가가 엔터프라이즈 분석서버에서 추적경로 정보와 공격의심행위를 검색할 수 있음을 보여주고 있다.

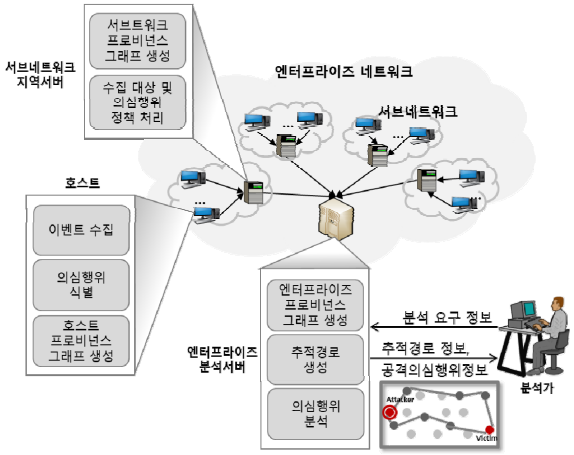

Fig. 6. Cyberattack tracing system high-level operational concept graphic(OV-1)

4. 운용아키텍처에 대한 논의

엔터프라이즈 프로비넌스 그래프는 DAG(Directed acyclic graph)이다. DAG 생성에 소모되는 시간은 버틱스와 에지의 숫자에 비례하여 기하급수적으로 증가한다. 따라서 이벤트의 숫자가 적은 각 지역 호스트와 서브네트워크 전처리서버에서 프로비넌스 그래프를 생성 후 중앙에서 통합하는 운용아키텍처가 높은 효

율을 기대할 수 있다.

그리고 호스트에 최대한 많은 기능을 할당할 수 있는 이유는 다음과 같다. OS에서 기본으로 제공하는 감사(Audit) 기능은 보안 목적으로 다양한 이벤트를 제공한다. 이를 이용하면 이벤트를 수집하는 개발 노력을 줄일 수 있는 장점이 있으나 이벤트 생성 자체에 대한 개입이 불가능하기 때문에 데이터에 대한 선별, 개입 및 성능 최적화가 불가능하다. 이에 비해 커널 드라이브 기반 데이터 수집은 개발 난이도가 상대적으로 높고 OS 및 커널 버전 변경시 지속적 유지보수가 필요하지만, 필요한 이벤트만 선별적으로 최소로 수집할 수 있어서 호스트 성능 부담을 줄일 수 있으며, 이벤트 수집과정에서 이벤트 간 연관관계를 분석하거나 파일의 내용, 네트워크 패킷 내용 분석 등을 더 상세하고 최적화하여 수집할 수 있다. 따라서 다단계 네트워크에 연결된 다수의 호스트에서 최대한 많은 처리를 하도록 분산처리를 하면서 호스트 자체의 성능 부담을 줄일 수 있는 커널 드라이브 기반 이벤트 수집이 적합한 것으로 설계 방향을 제안한다.

또한 사용자의 호스트 성능 저하감을 최소화하기 위해 화면보호기 작동시간 같은 사용자의 호스트 미사용 시간에 '호스트 프로비넌스 그래프'를 생성하고 전송할 수 있다.

사이버공격 추적시스템의 추적경로 생성 및 의심행위 분석 정보는 추적 기능 외에 사이버 보안의 다른 주제 영역으로 확장 가능하다. Table 2에서처럼 추적경로 정보 중 일부는 아티팩트 정보로 디지털 포렌식 증거로 활용할 수 있다. 또한 엔드포인트 APT 실시간 탐지 및 대응(EDR)에서는 보안사고 탐지, 엔드포인트에 보안사고 가두기(contain), 사고조사, 치료조치 가이드 등에도 사용할 수 있다. 그리고 제로트러스트 보안 관점에서 장기간 수집, 정제, 저장한 이벤트 데이터는 외부 위협뿐만 아니라 내부자 위협 분석으로도 확장하여 활용할 수 있다[14].

Table 2. Available cybersecurity areas of cyberattack tracing system

보안 기능	추적시스템 활용 가능 분야
디지털 포렌식	아티팩트 식별
EDR	사이버위협탐지, 침해사고조사
제로트러스트	내부자 위협 탐지

5. 결 론

체계적인 시스템엔지니어링은 시스템 설계 전에 운용아키텍처를 개발하여 사용자 요구사항 정의, 설계방향 정립, 이해당사자 간 의사소통 수단으로 활용한다. 본 논문은 엔터프라이즈에 침입한 악성코드와 그 행위를 자동으로 식별하기 위한 프로비넌스 그래프 데이터 처리절차와 설계 고려사항을 제시하였다. 그리고 이 데이터 처리 절차와 엔터프라이즈 환경 특성을 반영하여 사이버공격 추적시스템 운용아키텍처를 제안하였다. 제안한 운용아키텍처는 시스템 아키텍처와 상세설계로 발전시킬 예정이다.

후 기

이 논문은 2023년 정부(방위사업청)의 재원으로 국방과학연구소에서 수행한 연구결과임(912410301).

References

[1] https://attack.mitre.org ATT&CK, 2022.4.

[2] https://gartner.com, "Endpoint Detection and Response (EDR) Solutions Reviews and Ratings," 2022.

[3] https://car.mitre.org, MITRE Cyber Analytic Repository, 2020. 4.

[4] Wajih Ul Hassan, Adam Bates and Daniel Marino, "Tactical Provenance Analysis for Endpoint Detection and Response Systems," IEEE Symposium on Security and Privacy, 2020.

[5] Yuanzhao Gao, XingYuan Chen and Xuehui Du, "A Big Data Provenance Model for Data Security Supervision Based on PROV-DM Model," IEEE Access, Vol. 8, pp. 38742-38752, 2020.

[6] w3c.org/TR/2013/NOTE-prov-primer-20130430/#intuitive-overview-of-prov, "PROV Model Primer," W3C Working Group Note 30 April 2013.

[7] Ashish Gehani and Dawood Tariq, "SPADE: Support for Provenance Auditing in Distributed Environments," 13th ACM/IFIP/USENIX International Conference on Middleware, 2012.

[8] Ashish Gehani, Hasanat Kazmi, and Hassaan Irshad,

"Scaling SPADE to "Big Provenance," 8th USENIX Workshop on the Theory and Practice of Provenance (TaPP), 2016.

[9] Ashish Gehani, Raza Ahmad, Hassaan Irshad, Jianqiao Zhu and Jignesh Patel, "Digging Into 'Big Provenance'(With SPADE)," ACM Queue, Vol. 19(3), 2021.

[10] Do-Hyeon Choi and Jung-Oh Park, "Graph Database based Malware Behavior Detections Techniques," Journal of Convergence fro Information Technology, Vol. 11, No. 4, pp. 55-63, 2021.

[11] Xhang Xu, Zhenyu Wu and Zhichun Li, "High Fidelity Data Reduction for Big Data Security Dependency Analyses," In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 504-16, 2016 October 24-28.

[12] Zhenyuan Li, Ai Alfred Chen, Runqing Yang, Yan Chen and Wei Ruan, "Threat Detection and Investigation with System-Level Provenance Graphs: A Survey," Computers & Security, Vol. 106, July 2021, 102282.

[13] Republic of Korea Ministry of Defense, "Ministry of Defense Architecture Framework Version 1.5," 2019.

[14] Deirdre Doherty and Brian McKenney, "Implementing A Zero Trust Atchitecture: Are we there yet?," The MITRE, 2021.