Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.32, NO.3, Jun. 2022 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2022.32.3.527

ATT&CK 기반 공격체인 구성을 통한 APT 공격탐지 시스템 구현

조 성 영,[1†‡] 박 용 우,[2] 이 경 식[1]
[1,2]국방과학연구소 (선임연구원, 현역연구원)

Implementation of an APT Attack Detection System through ATT&CK-Based Attack Chain Reconstruction

Sungyoung Cho,[1†‡] Yongwoo Park,[2] Kyeongsik Lee[1]
[1,2]Agency for Defense Development (Senior Researcher, Researcher)

요 약

본 연구에서는 조직화된 공격 주체가 수행하는 APT 공격을 효과적으로 탐지하기 위하여, 공격체인을 구성하여 공격을 탐지하는 시스템을 구축하였다. 공격체인 기반 APT 공격 탐지 시스템은 다양한 호스트 및 네트워크 모니터링 도구에서 생성하는 이벤트를 수집하고 저장하는 '이벤트 수집 및 저장부', 이벤트로부터 MITRE ATT&CK®에 정의된 공격기술 수준의 단위공격을 탐지하는 '단위공격 탐지부', 단위공격으로 생성된 이벤트로부터 Provenance Graph 기반의 인과관계 분석을 수행하여 공격체인을 구성하는 '공격체인 구성부'로 구성하였다. 시스템을 검증하기 위하여 테스트베드를 구축하고 MITRE ATT&CK Evaluation 프로그램에서 제공하는 모의공격 시나리오를 수행하였다. 실험 결과 모의공격 시나리오에 대해 공격체인이 효과적으로 구성되는 것을 확인하였다. 본 연구에서 구현한 시스템을 이용하면, 공격을 단편적인 부분으로 이해하기보다 공격의 진행 흐름 관점에서 이해하고 대응할 수 있을 것이다.

ABSTRACT

In order to effectively detect APT attacks performed by well-organized adversaries, we implemented a system to detect attacks by reconstructing attack chains of APT attacks. Our attack chain-based APT attack detection system consists of 'events collection and indexing' part which collects various events generated from hosts and network monitoring tools, 'unit attack detection' part which detects unit-level attacks defined in MITRE ATT&CK® techniques, and 'attack chain reconstruction' part which reconstructs attack chains by performing causality analysis based on provenance graphs. To evaluate our system, we implemented a test-bed and conducted several simulated attack scenarios provided by MITRE ATT&CK Evaluation program. As a result of the experiment, we were able to confirm that our system effectively reconstructed the attack chains for the simulated attack scenarios. Using the system implemented in this study, rather than to understand attacks as fragmentary parts, it will be possible to understand and respond to attacks from the perspective of progress of attacks.

Keywords: APT, MITRE ATT&CK, TTPs, Attack Chain, Reconstruction

Received(04. 15. 2022), Accepted(05. 23. 2022)
† 주저자, sycho@add.re.kr
‡ 교신저자, sycho@add.re.kr(Corresponding author)

I. 서 론

최근 조직화된 공격 주체가 국가 또는 조직 수준의 지원을 받아 수행하는 것으로 판단되는 사이버 공격이 지속적으로 증가하고 있다. 특히 표적 공격(targeted attack) 또는 APT(advanced persistent threat) 공격과 같은 사이버 공격은 기밀 탈취, 시스템 파괴와 같은 궁극적인 공격 목표를 달성하기 위해 다양한 공격 방법들을 조합하여 여러 공격 단계들로 구성된 공격 캠페인(campaign)[1] 수준에서 수개월 또는 수년의 시간 동안 수행되는 것으로 보고되고 있다. 지난 2020년에 발생한 솔라윈즈(SolarWinds) 해킹 사건은 전형적인 APT 공격 사례로, 미국 정부 부처, 민간 기업 및 주요 인프라 조직을 표적으로 진행되었다. 국내에서도 통일부 등 대북 관련 업무 종사자를 대상으로 한 스피어피싱 공격 시도와 원자력연구원 해킹 사건 등을 포함한 전방위적인 사이버 공세가 이어지고 있다.

다양한 사이버 공격 중에서도 특히 APT 공격을 탐지하고 대응하기 위한 국내외 많은 연구가 진행되고 있으며, 다양한 정보보호 제품도 출시되어 여러 기관과 조직에 적용되고 있다. 미국에서는 이미 APT 공격을 탐지하기 위한 기술을 연구, 개발하고 있었으며, 대표적으로 미국 DARPA에서는 5년여 동안 Transparent Computing[1] 프로젝트에서 여러 대학, 기업들이 CADETS[2], ClearScope[3], MARPLE[4], ADAPT[5], TRACE[6] 등과 같은 다양한 연구가 진행되었다. Transparent Computing 프로젝트는 2장에서 후술할 Provenance Graph를 이용하여 시스템 내부의 가시성을 확보하고 시스템 내부에 대한 활동을 표현함으로써 APT 공격을 탐지하고 추적하고자 하였다.

이들을 포함한 최근의 연구 동향은 APT 공격을 개별적인 공격 단계 수준에서 바라보는 것이 아니라 공격이 수행되는 전반적인 맥락을 이해하려는 방향으로 진행되고 있다. 미국 해군에서는 지난 2020년 인공지능과 머신러닝을 이용한 공격 캠페인의 네트워크 탐지 챌린지[8]를 수행하였다. 이는 앞으로는 공격의 단편적인 부분의 탐지보다는 공격 캠페인 수준에서의 APT 공격 탐지 및 대응에 관심을 가진 것이라는 점을 시사한다. ATT&CK 프레임워크를 관리하는 MITRE 社의 산하 연구조직인 Center for Threat Informed Defense(CTID)에서 최근 발표한 Attack Flow[9]는 공격 행위를 개별적으로 바라볼 것이 아니라, 공격자가 공격 목표(goal)를 달성하기 위해 여러 공격기술을 순차적으로 수행하는 것에 초점을 두고 APT 공격 사례를 공격 흐름 형태로 분석하여 제시함으로써 보다 체계적으로 대응해야 한다고 제안하고 있다.

이러한 흐름에 따라, 본 연구에서는 사이버 공격(특히 APT 공격)을 구성하는 단위공격만을 탐지하는 단편적인 관점에서 벗어나, 단위공격을 하나의 흐름으로 연결하여 공격을 탐지하는 시스템을 제안하고 이를 구축하고자 하였다. 본 연구의 접근 방법은 다음과 같다.

(1) ATT&CK의 공격 TTPs(tactics, techniques, and procedures)를 이용하여 공격체인을 구성하는 공격기법을 표현하며, ATT&CK에 정의된 데이터 소스(data source) 및 데이터 컴포넌트(data component)를 이용하여 APT 공격을 구성하는 단위공격을 탐지하기 위해 수집해야 하는 이벤트를 식별하였다.
(2) 공격체인을 구성하기 위하여 Provenance Graph 기반의 이벤트 간 인과관계 분석을 통한 공격 탐지의 연구 방법론을 채택하였다.

2장에서는 본 연구에서 활용하는 ATT&CK 프레임워크, 경보 상관분석, Provenance Graph 기반의 APT 공격 탐지에 관한 기존 연구 내용을 살펴본다. 3장에서는 본 연구에서 제안하는 공격체인 기반 APT 탐지 시스템을 설명하고, 4장에서는 3장에서 설명한 시스템을 구현한 내용을 설명한다. 5장에서는 제안 시스템을 검증하기 위한 테스트베드 및 모의공격 시나리오를 설명하고, 6장에서는 실험 결과를 제시한다. 7장에서는 결론과 현재 시스템의 개선 방안을 포함한 향후 연구 방향을 제시한다.

II. 관련 연구

2.1 MITRE ATT&CK

현재 대부분의 정보보호 제품들은 공격자가 수행하는 SQL 인젝션, 무차별 대입 공격(brute-force

1) STIX에 따르면 공격 캠페인은 특정 공격대상에 대해 일정 기간 수행된 악성 행위 또는 공격의 집합을 설명하는 공격자 행위의 그룹으로 정의된다[7].

attack) 등과 같이 특정 시점의 단위공격을 탐지한다. 그러나 최근 10여 년 동안 공격자들은 정부, 국가기관 또는 대규모 단체의 지원을 받아 기업과 같은 조직의 형태를 구성하여 정보 탈취, 데이터 및 시스템의 가용성 또는 무결성 침해로 작전 또는 업무 방해와 같은 특정 목표를 달성하기 위하여 APT 공격을 진행하고 있다. APT 공격을 효과적으로 탐지하고 대응하기 위한 여러 활동 중 MITRE 社의 ATT&CK[10] 프레임워크는 APT 공격을 수행하는 여러 공격 그룹들 및 그들이 수행하는 공격기법들을 TTPs 형태로 체계적으로 정리하여 공개하고 있다. ATT&CK은 공격자의 단기적 목표에 해당하는 14가지의 전술(tactics)이 정의되어 있고, 각각의 전술적 목적을 달성하기 위해 공격자가 수행할 수 있는 공격기술(technique) 및 세부 공격기술(sub-technique)들이 정의되어 있다. 각 (세부) 공격기술은 하나 이상의 전술적 목적을 달성할 수 있다.

ATT&CK 프레임워크가 적용된 일부 사례로는 악성코드의 정적 분석을 위한 도구인 PEStudio[11], KISA에서 발간하는 침해사고 조사분석 보고서[12] 등이 있다. ATT&CK 프레임워크는 사이버 위협과 관련한 공격자의 공격 행위 및 기법을 설명하기 위한 분류체계의 사실상 표준으로 활용되고 있다.

2.2 경보 상관분석

경보 상관분석(alert correlation)은 정보보호 제품 또는 시스템에서 발생하는 저수준(low-level) 이벤트 또는 로그(이하 이벤트) 간의 상관관계를 분석하여 고차원(high-level)의 공격 정보를 유추하는 기술을 통칭한다.

저차원(low-level) 이벤트 데이터가 다량으로 발생하여 보안 분석가가 적시에 사이버 공격 상황을 인지하는 데 어려움을 겪었다면, 경보 상관분석은 고수준 관점의 공격 정보를 유추하여 적시에 공격 상황을 종합적으로 인식하고 대응하여 공격자의 목표 달성을 좌절시킬 수 있다. 또는 이미 발생한 침해사고에 대해 공격을 추적하여 공격의 원인을 분석하고 공격이 확산되지 않도록 대응할 수 있다. 또한 오탐(false positive) 이벤트가 과다하게 발생하였을 때도 공격과 오탐을 구분할 수 있으며, 미탐(false negative)으로 인하여 공격을 실시간으로 탐지하지 못한 경우에도 전후 맥락(context)을 이용하여 공격을 인지하고, 미탐이 발생하지 않도록 탐지 룰을 강화하는 등 공격에 대비할 수 있다.

경보 상관분석은 약 20여 년 동안 연구되어 오고 있으며, 크게 유사성(similarity) 기반[13-15], 사전조건-결과(precondition-consequence) 기반[16-18], 그리고 시나리오 기반[19-20] 경보 상관분석 등이 있다. 여기에 더해 머신러닝 기반[21-22]의 상관분석 방법들이 지속해서 연구되고 있다.

2.3 Provenance Graph

Provenance Graph는 Fig. 1.과 같이 호스트 내의 개체(entity) 간 정보 흐름(information flow) 또는 제어 흐름(control flow)을 표현한 그래프이다. 그래프의 노드는 개체를 표현하며, 프로세스는 주체(subject) 개체이며, 프로세스를 제외한 파일, 파이프, 네트워크 소켓 등은 객체(object) 개체이다. 그래프의 엣지는 주체 개체와 객체 개체 간의 흐름 또는 의존성(dependency)을 나타내기 때문에 Provenance Graph를 의존성 그래프(Dependency Graph)라고도 한다.

만약 특정 프로세스가 매우 오랫동안 실행되고 있으면, 이와 관련하여 무수히 많은 입력(input)과 출력(output)이 관여하여 그래프의 크기가 매우 커지는 의존성 폭발(Dependency Explosion) 문제가 발생한다. 예를 들어 공격과 연관된 악성 프로세스를 식별하였을 때 이와 직접 연관된 입출력(예를 들어 파일 또는 네트워크 소켓)만을 식별하는 것이 아니라 무수히 많은 입출력을 악성으로 분류하여 그래프가 매우 크게 팽창한 모습으로 나타난다.

Provenance Graph를 이용하여 공격을 추적할 수 있는데 이 중 역방향 분석(backward analysis)은 공격을 탐지한 이벤트와 관련한 개체 노드로부터 역으로 탐색하여 공격의 진입 지점(entry point)을

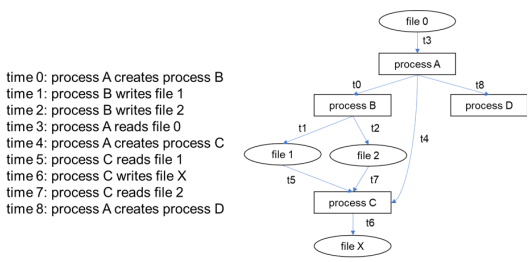


Fig. 1. An example of provenance graph (redraw of [24]). Left: example events, Right: provenance graph for entire events

식별하는 과정이며, 정방향 분석(forward analysis)은 역방향 분석을 통해 식별한 공격의 진입 지점으로부터 Provenance Graph를 추적하여 공격의 모든 영향 및 종료 지점(exit point)을 식별하는 과정이다.

Provenance Graph를 이용하여 이벤트 간의 인과관계를 분석하여 공격을 탐지하기 위한 다양한 연구가 진행되었다[23-32]. 관련 연구를 종합하면 Provenance Graph를 이용하여 APT 공격을 실시간으로 탐지하거나[25-27], Graph를 이용한 이벤트 간 인과관계 분석을 통하여 공격 사후 침해사고를 분석하거나 추적하였다[23-24, 28-32]

III. 공격체인 기반 APT 탐지 시스템

본 연구에서 제안하는 공격체인 기반 APT 탐지 시스템은 Fig. 2.와 같이 '이벤트 수집/저장부', '단위공격 탐지부', 그리고 '공격체인 구성부'로 구성된다.

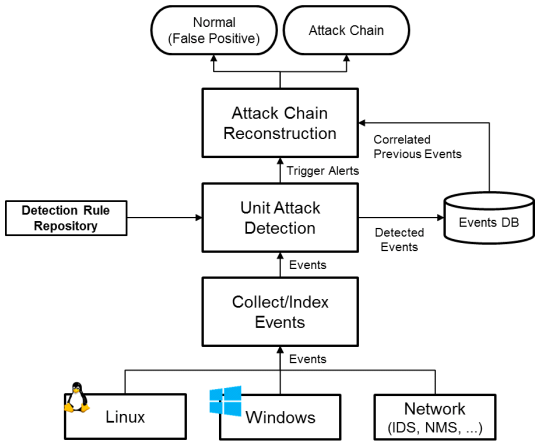


Fig. 2. System Overview

3.1 이벤트 수집/저장

엔터프라이즈 환경에서 네트워크에 연결된 모든 Windows 및 Linux 호스트, 네트워크 공격을 탐지하는 침입탐지시스템(NIDS, network-based intrusion detection system)과 네트워크 모니터링 시스템(NMS, network monitoring system)에서 정상 행위 및 잠재적 공격 행위로 인하여 발생하는 이벤트를 수집, 저장한다.

ATT&CK에서 정의한 각 (세부) 공격기술을 탐지하는 데 필요한 데이터 소스 및 데이터 컴포넌트[33]와, 이와 매핑되는 구체적인 이벤트 목록[34]을 이용하여 수집해야 할 이벤트를 구체적으로 식별하였다.

예를 들어, ATT&CK에 정의된 (세부) 공격기술 중 하나인 "Command and Scripting Interpreter: PowerShell"(T1059.001)을 탐지하기 위해 사용할 수 있는 데이터 소스 및 데이터 컴포넌트는 "Command Execution", "Module Load", "Process Creation", "Script Execution"이 있으며, 이와 연관된 이벤트는 Sysmon 이벤트 ID 1 또는 Windows 이벤트 ID 4688이 있다. 이들 이벤트를 이용하여 T1059.001과 관련된 공격에 대한 이벤트를 식별할 수 있다. 그러나 ATT&CK에서 정의한 데이터 소스 및 데이터 컴포넌트는 연관된 구체적 이벤트 종류를 명시하지 않고 있으며, [34]에서는 Windows 호스트에서 수집할 수 있는 이벤트만을 목록에서 제시하고 있다는 한계가 있다.

3.2 단위공격 탐지

ATT&CK에 정의된 (세부) 공격기술과 매핑되는 단위공격 탐지 규칙을 정의하고 단위공격 수준의 행위로 인하여 발생되어 수집/저장된 이벤트를 탐지한다. IDS와 같은 정보보호 제품에서의 탐지 규칙은 패킷에서 나타나는 시그니처를 이용하여 공격을 탐지하는 반면에 행위 기반의 규칙은 공격이 이루어졌을 때 발생하는 이벤트의 특징을 이용하여 공격을 탐지하는 점에서 차이가 있다.

앞에서 예로 든 "Command and Scripting Interpreter: PowerShell"(T1059.001)의 경우 PowerShell이 실행된 이벤트(powershell.exe 프로세스 생성 이벤트)가 발생하였을 때 이를 탐지하는 규칙을 설정하면, PowerShell이 실행될 때마다 규칙에서 해당 이벤트를 탐지하고 경보가 발생한다.

3.3 공격체인 구성

단위공격 탐지부에서 단위공격을 탐지하여 트리거된 이벤트는 연관된 다른 과거 이벤트와 인과관계 분석(causality analysis)을 수행하여 공격체인을 구성한다. 이벤트 간 인과관계 분석을 위하여 앞서 설명한 Provenance Graph 개념을 적용하였으며, 본 연구에서의 이벤트 간 인과관계는 Table 1.과 같이 정의할 수 있다.

Table 1. Types of causal relationship between events

Category	Type	
Process	Same Process	
	Parent-Child Process	
File	File loaded	
	File executed	
	File transferred	
	File used (etc)	
Network	Same	Source-Destination IP/Port
	Reverse	
Persistence	Service Installation	
	Service Execution	
Lateral Movement	Same user, different host	
	Remote logon	
	Remtoe service (SSH, RDP)	

3.3.1 프로세스 기반 인과관계

프로세스 기반 인과관계는 동일한 호스트에서 발생한 두 이벤트가 다음의 조건 중 하나를 만족할 때 서로 인과관계가 있다고 판단한다. 프로세스 기반 인과관계를 분석하기 위해서는 각 이벤트에는 Table 2.의 프로세스에 대한 속성을 이용한다.

(1) Same Process: 두 이벤트가 같은 프로세스에 의한 활동일 경우. Fig. 3.에서 ①번 이벤트와 ②번 이벤트, ①번 이벤트와 ③번 이벤트는 같은 프로세스 B에 의한 활동이므로 ①→② 및 ①→③의 인과관계가 있다.

(2) Parent-Child Process: 선행하는 이벤트와 관련된 프로세스가 후행하는 이벤트에 관련한 프로세스의 부모 프로세스일 경우. Fig. 3.에서 ③번 이벤트와 ④번 이벤트의 경우 각각 프로세스 B와 C에 의한 활동이며 프로세스 B는 프로세스 C의 부모 프로세스이므로 ③→④의 인과관계가 있다.

Table 2. Properties of process in an event (field names defined in ECS)

Field name	Description
process.pid	Process PID
process.name	Process Name
process.executable	Process Execution Path

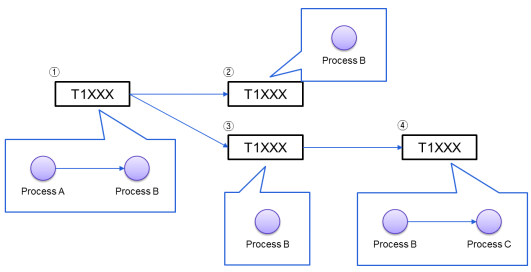


Fig. 3. Process-based causal relationship

3.3.2 파일 기반 인과관계

파일 기반 인과관계는 두 이벤트가 다음의 조건 중 하나를 만족할 때 서로 인과관계가 있다고 판단한다. 파일 기반 인과관계를 분석하기 위해서는 파일과 관련한 이벤트에는 Table 3.의 파일에 대한 속성을 이용한다.

(1) File loaded: 선행하는 이벤트(Fig. 4.의 ①번 이벤트)에 명시된 파일 경로(file.path)가 후행하는 이벤트(Fig. 4.의 ②번 이벤트)의 프로세스의 실행 인자(process.argument)에 명시된 경우(①→②)

(2) File executed: 선행하는 이벤트(Fig. 4.의 ①번 이벤트)에 명시된 파일 경로가 후행하는 이벤트(Fig. 4.의 ②번 이벤트)에 명시된 프로세스의 실행 경로(process.executable)와 일치하는 경우(①→②)

(3) File transferred: 선행하는 이벤트(Fig. 4.의 ①번 이벤트)에 명시된 파일 경로 또는 파일 이름(file.name)이 후행하는 이벤트(Fig. 4.의 ③번 이벤트)에 명시된 네트워크에서 전송되는 파일 경

Table 3. Properties of file in an event (field names defined in ECS)

Field name	Description
file.name	File Name
file.path	File Path

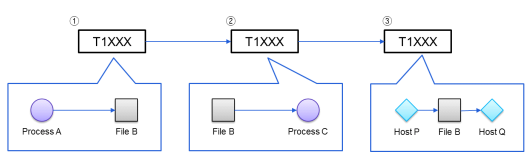


Fig. 4. File-based causal relationship

로 또는 파일 이름과 일치하는 경우(①→③)

(4) File used: 선행하는 이벤트에 명시된 파일 경로가 후행하는 이벤트에 명시된 파일 경로와 일치하는 경우. 이는 앞의 3가지 상황에 해당하지 않는다.

3.3.3 네트워크 기반 인과관계

네트워크 기반 인과관계는 네트워크 활동과 관련한 두 이벤트(Fig. 5.의 ①번 이벤트와 ②번 이벤트)가 다음의 조건 중 하나를 만족할 때 서로 인과관계가 있다고 판단한다. 네트워크 기반 인과관계를 분석하기 위해서는 네트워크와 관련한 이벤트에는 Table 4.의 네트워크 호스트에 대한 속성을 이용한다. 두 경우 모두 IP와 포트가 모두 일치하는 것이 바람직하나, 모두 일치하지 않더라도 어느 한쪽의 IP와 포트가 같고 다른 쪽의 IP가 일치하면 두 이벤트는 인과관계가 있다고 볼 수 있다.

(1) Same Source-Destination: 선행하는 이벤트에 명시된 출발지 및 목적지가 후행하는 이벤트에 명시된 출발지 및 목적지와 일치하는 경우(Fig. 5.의 ①→②)

(2) Reverse Source-Destination: 선행하는 이벤트(A)에 명시된 출발지 및 목적지가 후행하는 이벤트(B)의 목적지 및 출발지와 일치하는 경우(A.source=B.destination 및 B.destination=A.source)

Table 4. Properties of source and destination in an event (field names defined in ECS)

Field name	Description
source.ip	Source IP address
source.port	Source Port number
destination.ip	Destination IP address
destination port	Destination Port number

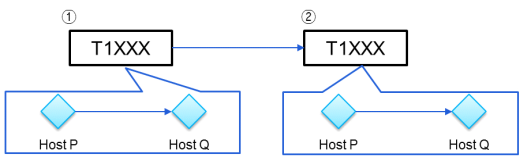


Fig. 5. Network-based causal relationship

3.3.4 지속성(persistence) 기반 인과관계

지속성 기반 인과관계는 두 이벤트가 공격의 지속성을 유지하기 위한 과정에서 다음의 조건 중 하나를 만족할 때 서로 인과관계가 있다고 판단한다.

(1) Service Installation: 후행하는 이벤트(Fig. 6.의 ②번 이벤트)가 공격의 지속 유지를 위해 서비스(service)를 설치하는 이벤트일 때 서비스의 경로(service.path)가 선행하는 이벤트(Fig. 6.의 ①번 이벤트)의 파일 경로와 일치하는 경우(①→②)

(2) Service Execution: 선행하는 이벤트(Fig. 6.의 ②번 이벤트)가 공격의 지속 유지를 위해 서비스를 설치하는 이벤트일 때 서비스의 경로가 후행하는 이벤트(Fig. 6.의 ③번 이벤트)의 프로세스 실행경로와 일치하는 경우(②→③)

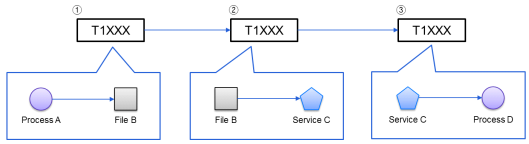


Fig. 6. Persistence-based causal relationship

3.3.5 내부 확산(lateral movement) 기반 인과관계

내부 확산 기반 인과관계는 공격자가 특정 호스트에서 다른 호스트로 내부 확산을 하는 과정에서 발생하는 두 이벤트가 다음의 조건 중 하나를 만족할 때 서로 인과관계가 있다고 판단한다.

(1) 동일한 사용자, 다른 호스트(Fig. 7. (a)): 선행하는 이벤트(①)에서 활동하는 사용자(user.name)가 후행하는 이벤트(②)와 같은 사용자이면서, 두 이벤트에 명시된 호스트(host.ip)가 서로 다른 경우, 선행하는 이벤트가 발생한 호스트에서 후행하는 이벤트가 발생한 호스트로 내부 확산을 하였다고 볼 수 있다. (①→②)

(2) 원격 로그인(Fig. 7. (b)): 선행하는 이벤트(③)와 후행하는 이벤트(④)가 원격 로그인 관련 이벤트를 의미하고, 선행하는 이벤트의 목적지 호스트와 후행하는 이벤트의 접속 호스트가 일치하는 경우, 선행하는 이벤트가 발생한 호스트에

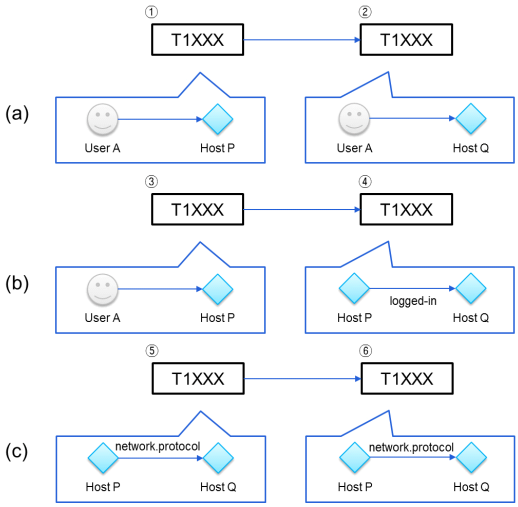


Fig. 7. Lateral movement-based causal relationship. (a) Same user, different hosts, (b) Remote logon, (c) Remote service

서 후행하는 이벤트가 발생한 호스트로 내부 확산을 하였다고 볼 수 있다. (③→④)

(3) 원격 서비스(Fig. 7. (c)): 선행하는 이벤트(⑤)와 후행하는 이벤트(⑥)가 원격 서비스 접속과 관련한 이벤트를 의미하고, 선행하는 이벤트의 목적지 호스트와 후행하는 이벤트의 접속 호스트가 일치하는 경우, 선행하는 이벤트가 발생한 호스트에서 후행하는 이벤트가 발생한 호스트로 내부 확산을 하였다고 볼 수 있다. (⑤→⑥) 이 경우 현재 연구에서는 가장 많이 사용하는 원격 서비스인 SSH와 원격 데스크톱 프로토콜(RDP, remote desktop protocol)의 경우를 명시하였다.

3.3.6 인과관계 기반 공격체인 구성

단위공격 탐지부에서 정의된 룰셋에 의해 특정 이벤트가 트리거되어 경보가 발생하면, 그 시점 이전에 트리거되어 Fig. 2.의 이벤트 DB에 저장된 과거 이벤트들과 인과관계 분석을 수행한다. 그 결과 현재 트리거된 이벤트와 인과관계가 있다고 판단되는 연관 이벤트가 발견되면 해당 이벤트들을 연결한다. 이때 현재 트리거되어 저장된 이벤트에 Fig. 8.과 같은 구조에 연관된 과거 이벤트의 정보(ID)를 저장한다.

인과관계 분석은 단위공격 탐지부에서 단위공격에 대한 이벤트가 트리거될 때마다 수행되어, 현재 트리거된 이벤트 관점에서는 인과관계가 있다고 판단되는 이벤트로부터 인과관계가 있다고 판단되는 최초 이벤트까지 체인 형태로 연결된 모습을 관찰할 수 있다.

```
{
  "_id": "XXXX",
  ...,
  "prev": {
    "process": {"same":[], "parent_child":[]},
    "file": {"loaded":[], "executed":[], "transferred":[],
             "used":[]},
    "network": {"same":[], "reverse":[]},
    "persistence": {"service_installation":[],
                    "service_execution":[]},
    "lm": {"same_user":[], "remote_logon":[],
           "remote_service":[]}
}
```

Fig. 8. JSON-formatted data schema to save causally related previous events for a event

IV. 시스템 구축

4.1 이벤트 수집/저장

4.1.1 Windows 호스트 이벤트 수집

MITRE ATT&CK에 정의된 데이터 소스 및 데이터 컴포넌트, 그리고 [34]의 이벤트 목록에 대한 연관분석을 수행하여, Table 5.에 나열한 이벤트 채널로부터 Windows 호스트에서 발생하는 정상 및 공격 관련 이벤트를 수집하였다.

Sysmon[35]은 Windows에서 수행되는 행위를 효과적으로 모니터링하고 이를 이벤트에 로깅할 수 있는 도구이다. Sysmon을 호스트에 설치할 때 로깅 항목을 설정한 XML 형식의 설정 파일을 이용하면 다량으로 발생하는 정상 행위와 관련한 이벤트의 수를 감소시킬 수 있다. 본 연구에서는 Sysmon 설정 파일 중 가장 많이 사용되는 SwiftOnSecurity[36]에서 제공하는 설정 파일을 기반으로 하되, 기본으로 생성하는 이벤트를 배제하지 않도록 수정하였다. 설정 파

Table 5. Events to collect from Windows Host

Application
Security (Microsoft-Windows-Security-Auditing)
System
Microsoft-Windows-Sysmon
Microsoft-Windows-PowerShell
Microsoft-Windows-WMI-Activity

일 내부에 〈include〉 항목에 대한 규칙을 설정하면 설정한 항목에 대한 이벤트만 생성되어 설정한 규칙에 해당하지 않는 공격 행위와 관련된 이벤트를 수집할 수 없게 되므로, 설정 파일 중 〈include〉 항목에 나열된 규칙을 모두 제거하였다.

Security 이벤트에서 ATT&CK의 (세부) 공격기술과 관련한 행위 이벤트를 수집하기 위하여, Fig. 9.의 Windows의 로컬 그룹 정책 편집기의 "고급 감사 정책"에서 Table 6.에서 나열한 각각의 항목에 대해 "성공", "실패" 항목에 대한 감사 이벤트를 구성하도록 설정하였다.

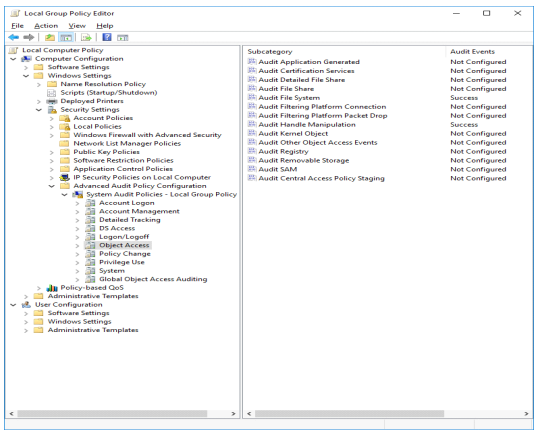


Fig. 9. Local Group Policy Editor(gpedit.msc) for Windows Security Event Auditing

Table 6. Items to audit in Local Group Policy Editor(gpedit.msc) for Windows Security Event Auditing

Category	Audit Items
Account Logon	Audit User Account Management
Detailed Tracking	Audit Process Creation
	Audit Process Termination
DS Access	Audit Directory Service Access
	Audit Directory Service Changes
Logon/ Logoff	Audit Account Lockout
	Audit Logoff
	Audit Logon
	Audit Other Logon/Logoff Events
	Audit Special Logon
Object Access	Audit File Share
	Audit File System
	Audit Filtering Platform Connection
	Audit Kernel Object
	Audit Other Object Access Events
	Audit Registry
	Audit SAM
Policy Change	Audit Authentication Policy Change
System	Audit Security System Extension

PowerShell에서의 활동을 로깅하기 위해 마찬가지로 Windows의 로컬 그룹 정책 편집기에서 "'컴퓨터 구성'→ '관리 템플릿'→'Windows 구성 요소'→ 'Windows PowerShell'"의 다음 항목에 대해 감사 이벤트를 구성하도록 설정하였다.

- Turn on Module Logging(모듈 로깅 사용)
- PowerShell 스크립트 블록 로깅 켜기

4.1.2 Linux 호스트 이벤트 수집

Linux의 경우 운영체제에서 기록하는 이벤트 로그(/var/log/sysmon 또는 /var/log/messages)와 auditd 데몬에서 생성하는 감사 로그를 기반으로 호스트에서 발생하는 정상 및 공격 관련 이벤트를 탐지한다. 쉘(shell) 명령어를 통한 행위(예, 프로세스 생성)를 탐지하기 위하여 auditd 설정 파일[37]을 적용하고 감사 로그를 수집하였다.

4.1.3 네트워크 이벤트 수집

네트워크 활동과 관련한 이벤트는 IDS(Suricata[38])에서 생성하는 경보 및 NMS(Zeek[39])에서 생성하는 이벤트 로그를 수집하였다.

4.1.4 이벤트 데이터 스키마

ATT&CK에 정의된 (세부) 공격기술 수준의 단위 공격을 탐지하고 이를 공격체인으로 구성하기 위해서는 수집한 이벤트를 일관된 형태로 변환하는 전처리 과정이 필요하다. 본 연구에서는 Elastic Stack에서 정의한 데이터 스키마인 Elastic Common Schema(ECS)[40]를 이용하였다. ECS는 이벤트를 구성하는 요소(필드)에 대한 명명 규칙에 일관성을 제공하며, 이벤트의 종류(예. 프로세스 생성, 파일 수정)에 대한 분류 라벨링을 제공하여 이벤트를 보다 직관적으로 분석할 수 있도록 돕는다.

4.1.5 이벤트 저장

본 연구에서는 이벤트를 수집하고 저장하기 위해 가장 많이 활용되는 제품인 Elastic Stack(Beat, Elasticsearch, Kibana)[41]를 이용하였다. 각 호

스트 및 IDS, NMS 등에는 에이전트인 비트(beat)를 설치하고, 비트를 통하여 지정된 이벤트를 Elasticsearch로 전송 및 저장하도록 구성한다. 시스템의 안정적인 운영을 위하여 7.15.2 버전의 Elastic Stack이 설치된 3개의 서버로 구성된 클러스터를 구축하였다.

4.2 단위공격 탐지

ATT&CK에 정의된 (세부) 공격기술 수준의 단위 공격을 탐지하기 위한 행위 기반의 탐지 규칙을 구축하였다. 이를 위하여 MITRE CAR(Cyber Analytics Repository)[42], Sigma(The Generic SIEM Rule)[43], Elastic Detection Rules[44]의 공개된 룰셋 리포지토리를 참고하였다. 본 연구에서 구축한 규칙은 총 420개로, 이에 대한 ATT&CK 전술별 분포는 Fig. 10.과 같다. 정찰(reconnaissance) 및 자원 개발(resource development) 전술은 방어 관점에서 시야 밖의 영역이므로 이에 대한 규칙은 공개된 룰셋 리포지토리에도 전무한 실정이다.

단위공격 탐지 규칙은 앞서 구축한 Elastic Stack의 Alert 기능을 이용하여 구축하였으며, 탐지 규칙에 따른 경보가 발생하였을 때 경보를 트리거한 이벤트를 MongoDB 및 Neo4j 기반의 이벤트 데이터베이스에 저장하고 공격체인 구성부에 REST API를 이용하여 전달하도록 구성하였다.

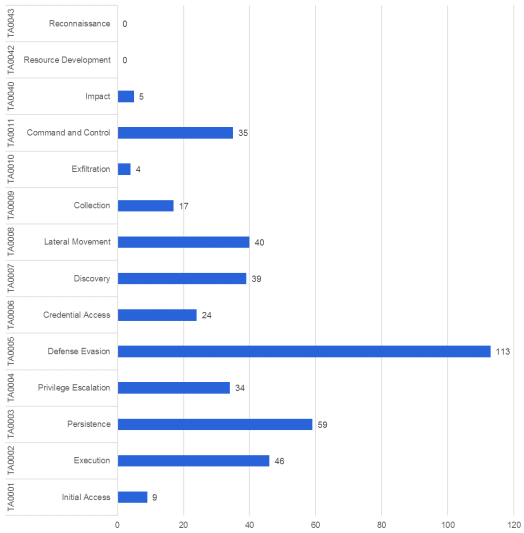


Fig. 10. Number of implemented rulesets per ATT&CK tactic

4.3 공격체인 구성

단위공격 탐지부에서 트리거된 이벤트에 대해 공격체인을 구성하는 공격체인 구성부는 Table 7.과 같이 시스템을 구축하였고, Python을 이용하여 프로그램을 구현하였다. 구현된 프로그램은 트리거된 이벤트에 대해 3.3절에서 기술한 방법에 따라 인과관계 분석을 수행하여 이벤트 데이터베이스에 저장된 과거 이벤트 중 연관된 이벤트들을 체인 형태로 구성하고 APT 공격 여부를 판단한다.

Table 7. Attack Chain Reconstruction System Specification

Hardware	
CPU	Intel Xeon Gold 2.5GHz 10 Core, 20 Threads
Memory	128GB
Storage	3.84TB
Software	
OS	Ubuntu 20.04.3 LTS
Python	3.8.10
Database	MongoDB (5.0.4), Neo4j (4.1.0)

V. 시스템 검증을 위한 실험

5.1 테스트베드 구성

5.1.1 네트워크 구성

본 연구에서 구축한 시스템을 검증하기 위하여, Fig. 11.과 같이 공격자가 위치한 네트워크와 피해 호스트들이 위치한 네트워크로 구성된 테스트베드를 구축하였다. 테스트베드는 VMware ESXi 6.7이 설치된 컴퓨터에서 가상 스위치, 가상 포트 그룹으로 네트워크를 구성하고 각각의 네트워크에는 가상머신들을 제작하여 배치하였으며, 각각의 네트워크는 pfSense[45] 라우터를 이용하여 구성하였다.

공격자 네트워크에는 공격자가 공격을 수행할 수 있는 가상머신 호스트들로 구성하였으며, 피해자 네트워크에는 공격을 탐지하고 이벤트를 발생할 수 있는 네트워크 IDS(Suricata) 및 NMS(Zeek)을 구축하고, Windows 가상머신 호스트(클라이언트의 경우

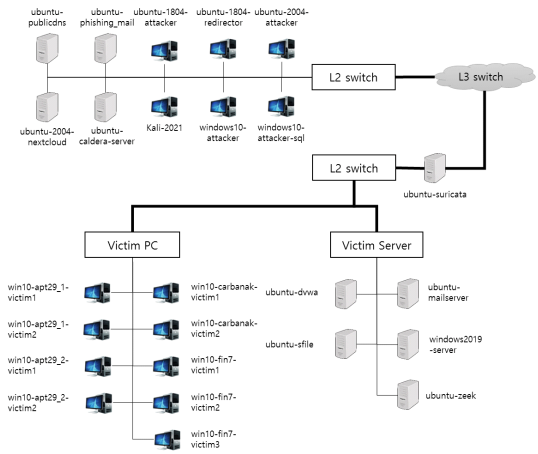


Fig. 11. Test-bed topology

Windows 10 Pro 20H2, 서버의 경우 Windows Server 2019 Standard), Linux 가상머신 호스트(Ubuntu 20.04 LTS)들을 구축하였다.

5.1.2 호스트 구성

공격자 호스트는 Table 8.과 같은 모의공격용 소프트웨어들로 구성하였다. 구체적으로는 다음 절에서 설명할 [50]의 APT29 공격 그룹을 모사한 두 가지의 공격 시나리오와, FIN7 공격 그룹을 모사한 각각의 공격 시나리오를 수행할 수 있도록 각각의 시나리오에서 요구하는 공격자 준비사항을 참고하였다.

피해자 호스트는 모의공격의 대상이 되도록 호스트 유형별로 Table 9.와 같이 구성하였다. 이때 모의공격 시나리오를 정상적으로 수행하기 위하여 Active Directory를 구성하였다.

Table 8. Software in attacker hosts

APT29	FIN7
Ubuntu 18.04 LTS	Kali 2021.2
Pupy[46], PoshC2[47] Metasploit [48]	Metasploit (for FIN7, Windows 10 Pro + MSSQL Server 2019)

5.2 모의공격 수행

앞서 구축한 테스트베드에서 이루어지는 APT 공격을 탐지하기 위하여, APT 공격을 수행할 수 있는 모의공격 시나리오가 필요하였다. 이를 위하여 MITRE 社에서 매년 수행하여 발표하는 MITRE ATT&CK Evaluations Program[49]의 모의공격 시나리오들을 활용하였다. 이 모의공격 시나리오들은 원래 엔드포인트에서 발생하는 공격을 탐지하고 대응하기 위한 보안제품인 EDR(Endpoint Detection and Response)의 성능을 평가하기 위해 개발되었다. MITRE 社에서는 알려진 공격 그룹들이 수행한 것으로 추정되는 APT 공격 사례를 분석하고 이를 모사할 수 있도록 매년 다른 시나리오로 재구성하고 이를 누구나 수행할 수 있도록 수행 절차 및 이와 관련된 악성코드 또는 공격 스크립트를 제공하고 있다[50]. 이 중 APT29 공격 그룹을 모사한 공격 시나리오 중 첫 번째 시나리오와 FIN7 공격 그룹을 모사한 공격 시나리오의 2가지 공격 시나리오를 수행하였다. 각 공격 시나리오의 수행 절차는 Table 10.에 요약하였다.

VI. 실험 결과

APT29 시나리오와 FIN7 시나리오를 수행한 기간 및 그동안 발생한 전체 이벤트 수, 단위공격 탐지부에서 트리거된 이벤트 수는 Table 11.과 같다. 표

Table 9. Victim hosts configurations

Items	Windows Endpoint	Windows Server	Linux Server
OS	• Windows 10 Pro 20H2	• Windows Server 2019 Standard	• Ubuntu 20.04.3 Server LTS
	• Active Directory (AD) joined		
SW	• Microsoft Office 2016 (Word, PowerPoint, Excel, Outlook)		
	• Browser: Google Chrome, Microsoft Edge		
For event collection	• Sysmon 13.33 [35] • Auditbeat/Winlogbeat 7.15.2		• Auditbeat/Filebeat 7.15.2
Disable security features	• Disable PowerShell execution policy • Disable Windows Defender • Disable OLE related security setting		

Table 10. Phases of penetration tests

Phase	APT 29 Scenario 1	FIN7 Scenario
1	Initial Breath	Initial Breath
2	Collection and Exfiltration	Delayed Malware Execution
3	Deploy Stealth Toolkit	Target Assessment
4	Defense Evasion and Discovery	Staging Interactive Toolkit
5	Persistence	Privilege Escalation
6	Credential Access	Expand Access
7	Collection and Exfiltration	Setup User Monitoring
8	Lateral Movement	User Monitoring
9	Collection	Persistence (Shim)
10	Persistence Execution	Exfiltration

Table 11. Scenario summary

	APT29	FIN7
Time duration	01:00	01:17
Total number of events	600,736	944,280
Number of events triggered for attack chain reconstruction	359	315
Events ratio	0.0598%	0.0334%

에서 알 수 있듯이 각 공격 시나리오를 수행하면서 발생한 이벤트 중 실제 공격과 관련된 이벤트는 1% 도 채 되지 않음을 알 수 있다.

각 공격 시나리오를 수행하면서 발생한 단계별 누적된 이벤트의 수는 Fig. 12.와 같다.

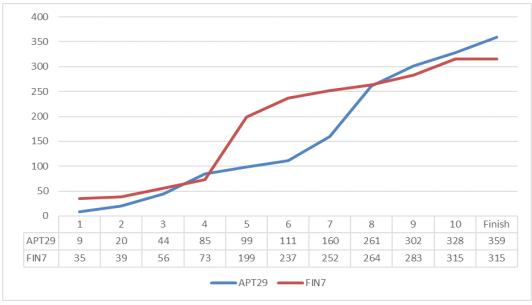


Fig. 12. Number of triggered events per phase for each scenario

6.1 Provenance Graph

두 가지 시나리오를 수행하여 얻은 Provenance Graph는 각각 Fig. 13.과 14.와 같다. 각 그림에서 프로세스는 파란색 상자, 파일은 회색 상자, 네트워크 연결의 경우 연결 대상 호스트는 주황색 상자로 나타내었다. 각 그림에서 프로세스를 나타내는 파란색 상자 중 굵게 강조한 프로세스 이름은 공격자의 C2 서버에 연결되어 활동하는 프로세스를 나타낸다.

6.1.1 APT29 시나리오

APT 29 시나리오(Fig. 13.)에서는 5단계에서 공격의 지속성 유지를 위해 ① 'javamtsup'이라는 이름의 서비스를 등록하여 시스템이 재시작할 때마다 서비스가 실행되도록 하고, ② 시작 프로그램 폴더에 'hostui.lnk'라는 이름의 바로가기 파일을 생성하여 시스템이 재시작할 때마다 'hostui.exe'라는 파일이 실행되도록 한다. 그러나 Provenance Graph에서는 5단계에서 수행한 두 지속 유지 행위와 10단계에서 시스템이 재시작할 때 실행된 두 지속 유지 행위가 연결되지 않은 것을 확인할 수 있었다. 이는 각각의 지속성 유지 방법(Fig. 13.에서의 ①, ②)을 직접 연결할 수 있는 이벤트가 없었기 때문이다.

6.1.2 FIN7 시나리오

FIN 시나리오(Fig. 14.)에서는 첫 번째 피해자 호스트(Victim 1)에서 두 번째 피해자 호스트(Victim 2)로의 내부 확산 공격을 수행하여 두 호스트와 관련된 Provenance Graph가 연결되는 모습을 보여 주고 있다. 그러나 세 번째 피해자 호스트(Victim 3)에서는 다른 두 피해자 호스트와의 연결 관계를 확인할 수 없었다. 이는 8단계의 사용자 모니터링 단계에서 두 번째 피해자 호스트에서 키로깅(keylogging)을 통하여 세 번째 피해자 호스트로의 접속 정보(IP 및 자격증명 정보)를 획득하고, 9단계에서 공격자 호스트에서 로컬로 포트 포워딩(port forwarding)을 설정하여 세 번째 피해자 호스트로 원격 데스크톱 연결(RDP)을 수행하여 내부 확산을 시도했기 때문에 8단계와 9단계를 연결할 수 있는 이벤트가 없기 때문이다.

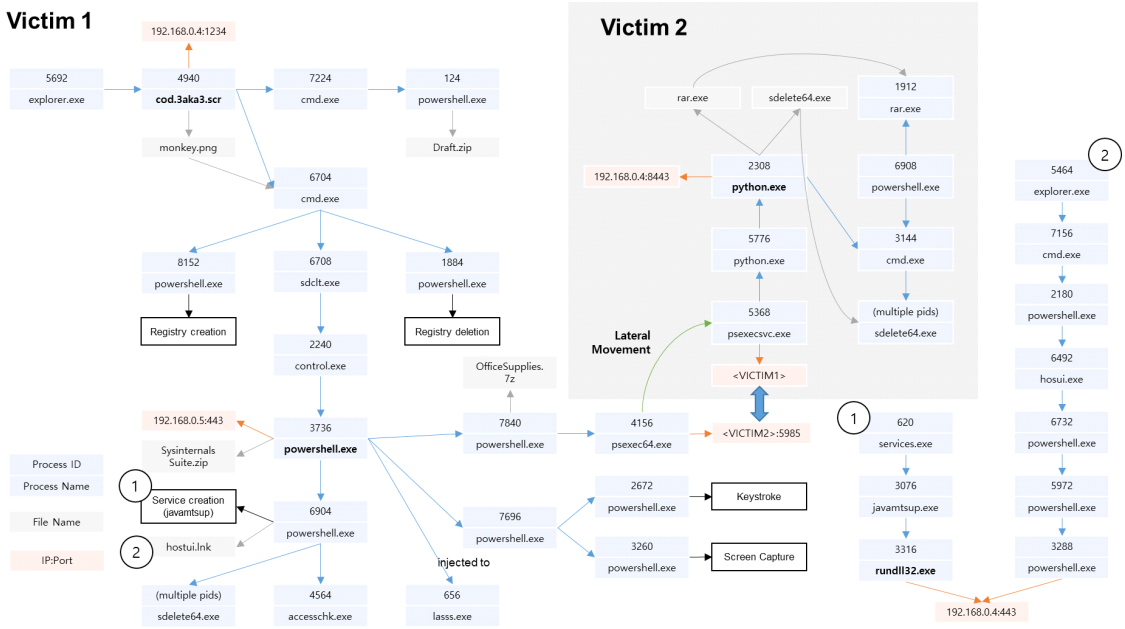


Fig. 13. Provenance Graph of APT29 Scenario

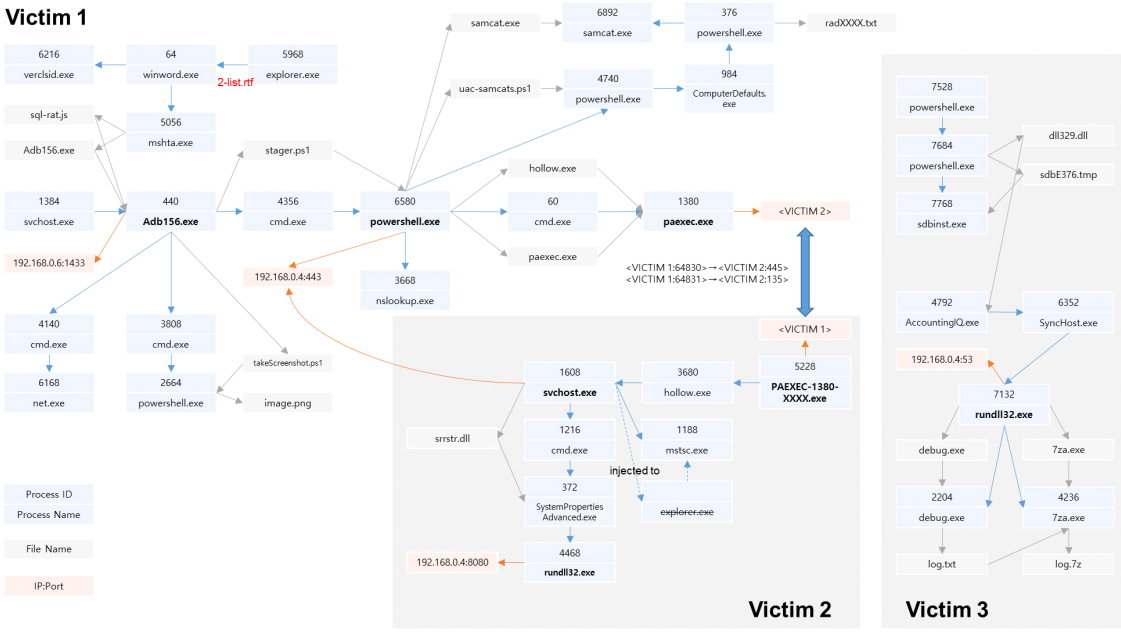


Fig. 14. Provenance Graph of FIN7 Scenario

6.2 공격체인 구성

두 모의공격 시나리오를 수행하여 얻은 공격체인 그래프는 각각 Fig. 15.와 16.과 같다. 그림에서 각각의 인과관계 종류는 선 색(프로세스는 검정색, 파일은 주황색, 네트워크는 초록색, 지속 유지는 파란색, 내부 확산은 노란색)으로 나타내었다.

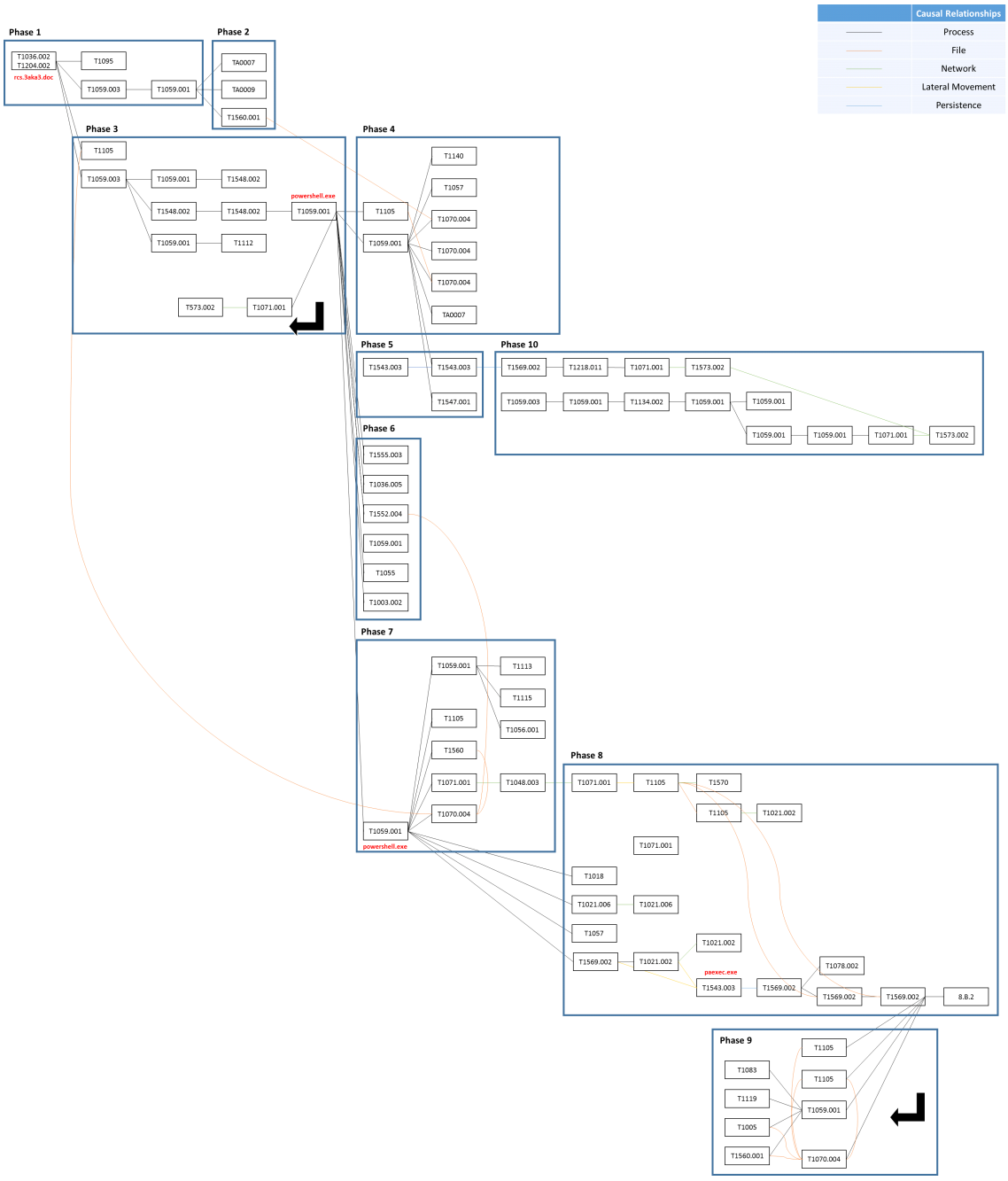


Fig. 15. Attack Chain of APT29 Scenario

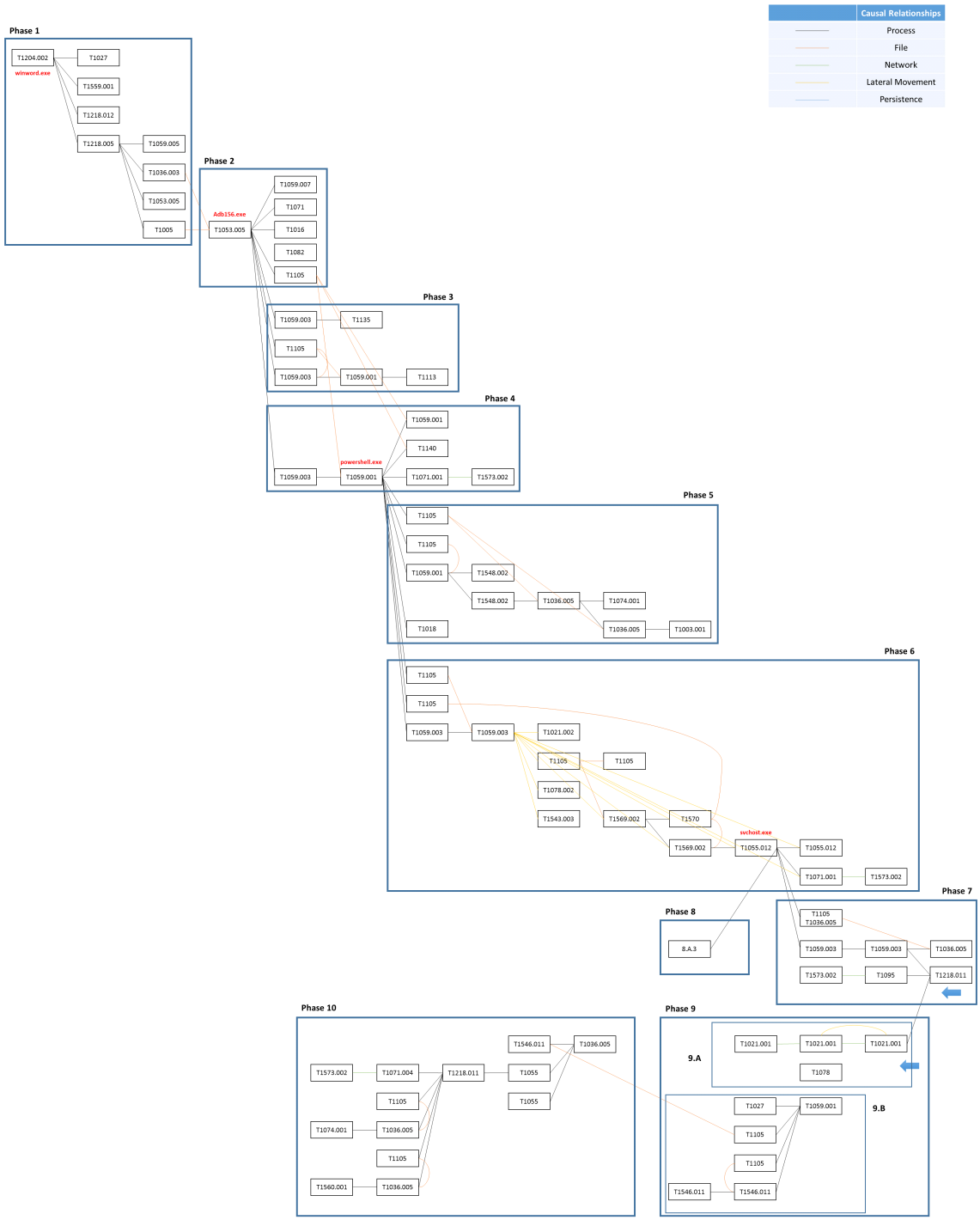


Fig. 16. Attack Chain of FIN7 Scenario

6.2.1 APT29 시나리오

APT29 시나리오(Fig. 15.)에서는 1단계를 시작으로 1→2, 1→3→4→5→10, 1→3→6, 1→3→7→8→9단계로 이어지는 크게 네 종류의 체인이 하나의 큰 체인으로 연결되는 모습을 관찰할 수 있었다. 시나리오를 진행하면서 핵심이 되는 단계는 1단계에서 초기에 악성코드(rcs.3aka3.doc 파일)가 초기에 시작한 단계, 3단계에서 사용자 계정 컨트롤(UAC, user account control) 우회(bypass)를 통하여 높은 수준의 권한으로 PowerShell이 실행된 단계, 3단계에서 얻은 PowerShell로부터 7단계에서 PowerShell을 실행한 단계이다.

APT29 시나리오의 Provenance Graph에서는 5단계의 지속 유지 수행과 시스템 재시작 후 수행된 10단계의 지속성 유지 방법 실행 단계가 연결되지 않았으나 인과관계 중 지속 유지 관계에 따라 5단계의 공격기술들과 10단계의 공격기술들이 체인으로 연결된 것을 확인할 수 있었다.

6.2.2 FIN7 시나리오

FIN7 시나리오(Fig. 16.)에서는 1단계를 시작으로 1→2→3, 1→2→4→5, 1→2→4→6→7→9, 1→2→4→6→8, 9→10단계로 이어지는 크게 네 종류의 체인이 두 개의 큰 체인으로 연결되는 모습을 관찰할 수 있었다. 시나리오를 진행하면서 핵심이 되는 단계는 1단계에서 초기에 마이크로소프트 워드 문서 형태의 악성코드(2-link.rtf 파일)가 시작한 단계, 2단계에서 문서형 악성코드로부터 페이로드를 조합하여 Adb156.exe 프로세스가 실행하는 단계, 2단계에서 전달된 PowerShell 스크립트(stager.ps1 파일)가 실행되어 공격자의 C2 서버에 연결되는 단계, 6단계에서 두 번째 피해 호스트로 내부 확산을 수행하면서 svchost.exe 프로세스가 실행되어 또 다른 공격자의 C2 서버에 연결되는 단계이다.

FIN7 시나리오의 Provenance Graph에서는 9단계 이후 수행되는 공격은 이전의 Provenance Graph와 연결되지 않았고, 공격체인에서도 9단계 내 세부 공격 단계에서 공격체인이 연결되지 않는 모습을 보였다. 이는 2번째 피해 호스트에서 3번째 피해 호스트로 원격 데스크톱으로 접속하는 과정(9.A)까지는 이전 공격 단계와 공격체인으로 연결되었으나, 9.B에서는 공격자가 정상 사용자처럼 PowerShell을 실행하는 경우이므로 이벤트를 이용해서는 이전 공격 행동들과 공격체인으로 연결할 수 없어 9.A와 9.B는 서로 연결되지 않는 것을 확인할 수 있었다.

6.3 공격체인 구성 시 소모 자원

두 가지 시나리오를 수행하면서 시스템이 동작하였을 때, 특히 공격체인 구성부에서 공격체인을 구성하면서 소모한 자원(CPU 및 메모리)의 점유 수준을 측정하였다. 측정 결과는 Table 12.와 같으며, 매우 미미한 수준의 자원이 소모됨을 확인할 수 있었다.

Table 12. Resource consumption for attack chain reconstruction

	APT29	FIN7
CPU Utilization Ratio (%)	0.0	0.0
Memory Usage (MB)	138.97	132.69

VII. 결론 및 향후 연구

APT 공격을 탐지하기 위해, 공격의 단편적인 부분만을 탐지하는 기존의 접근 방법 대신 진행 흐름 관점에서 공격을 체인 형태로 구성하는 시스템을 구축하였다. MITRE ATT&CK Evaluation 프로그램에서 제공하는 모의공격 시나리오들을 수행하여, 시스템이 공격체인을 구성하여 효과적으로 APT 공격을 탐지할 수 있음을 확인하였다.

현재 구축한 시스템이 개선되어야 할 방향은 다음과 같다. 첫째, 공격체인 구성부에서 실시간으로 공격체인을 구성할 수 있도록 설계하였으나, 모의공격 시나리오를 모두 완료한 후 그 결과로 생성된 이벤트에 대해 오프라인으로 공격체인을 구성하도록 구현되었다. 향후에는 현재 공격 시점에서 발생한 이벤트에 대해 실시간으로 공격체인을 구성하도록 개선함으로써, 공격자가 궁극적인 공격 목표를 달성하기 전에 대응할 수 있는 시스템과 연계하도록 할 계획이다.

둘째, MITRE ATT&CK Evaluation 프로그램에서 제공하는 모의공격 시나리오들을 수행하면서 탐지할 수 없는 세부 공격 단계들이 식별되었다. 이는 ATT&CK에 정의된 특정 공격기술을 탐지할 수 있는 데이터 소스에 대한 이벤트를 수집하고 있다고 하더라도 수집되는 이벤트가 포함하는 정보가 제한적이어서 탐지하기 어렵거나, 해당 데이터 소스가 개념적

수준으로 정의되어 있어 이에 해당하는 또 다른 이벤트 종류를 식별하여 수집하지 못한 것이 원인이다. 따라서 향후 연구를 통하여 더 다양한 이벤트를 수집함으로써 보완하여야 한다.

셋째, 단위공격 탐지 규칙에 대한 한계는 시그니처 기반 IDS가 가지는 한계와 많은 점을 공유한다. 즉 규칙에 존재하지 않는 공격이 발생하였을 때 이에 대한 이벤트가 발생하더라도 단위공격 탐지부에서 해당 이벤트를 트리거할 수 없다. 다만 Sigma를 포함한 개방된 규칙 리포지토리에서 여러 보안 전문가들의 노력으로 다양한 규칙을 개발하여 공유하고 있으므로 이를 활용하여 개선할 수 있고, 더불어 ATT&CK에 정의된 공격기술에 대한 심층 분석을 통하여 더 넓은 탐지 범위를 가지는 규칙들을 구축할 수 있을 것이다.

마지막으로, 공격체인을 구성하기 위한 이벤트 간 인과관계 분석을 위해서, 지속성 기반 및 내부 확산 기반 인과관계는 ATT&CK 전술 중 각각 지속 유지(persistence) 및 내부 확산(lateral movement) 전술에 속한 여러 (세부) 공격기술 중 일부만을 포함하고 있으므로, 향후 연구에서 더 많은 인과관계를 분석하여 공격체인을 구성하여야 한다.

향후 연구에서는 이와 같은 개선 사항을 반영하여, APT 공격을 탐지하였을 때 위협 인텔리전스를 이용하여 해당 공격 진행 상황에 대해 유사한 공격 그룹 또는 공격 캠페인을 분류함으로써 공격자가 궁극적인 공격 목표(즉, 정보 탈취나 시스템 또는 데이터의 기밀성/무결성 훼손 등)를 달성하지 못하도록 방어하는 시스템과 연계하고자 한다.

References

[1] Defense Advanced Research Projects Agency(DARPA), Transparent Computing (Archived), https://www.darpa.mil/program/transparent-computing, accessed on Mar. 2022

[2] Amanda Strnad, Quy Messiter, Robert Watson, Lucian Carata, Jonathan Anderson and Brian Kidney, "Casual, adaptive, distributed, and efficient tracing system (CADETS)," AFRL-RY-WP-TR-2019-0115, BAE Systems, Sep. 2019

[3] Michaell Gordon, Jordan Eikenberry, Anthony Eden, Jeffrey Perkins, Malavika Samak, Henny Sipma and Martin Rinard, "ClearScope: Full stack provenance graph generation for transparent computing on mobile devices," AFRL-RY-WP-TR-2020-0013, Massachusetts Institute of Technology, Jul. 2020

[4] Josyula Rao, Yan Chen, R. Sekar, Venkat Venkatakrishnan, "Mitigating advanced and persistent threat (APT) damage by reasoning with provenance in large enterprise network (MARPLE) Program," AFRL-RY-WP-TR-2019-0285, International Business Machines Corporation, Jan. 2020

[5] Ryan Wright, Alan Fern, Anthony Williams, James Cheney, Ghita Berrada and Sid Ahmed Benabderrahmane, "A diagnostics approach for persistence threat detection (ADAPT)," AFRL-RY-WP-TR-2019-0140, Galois, Inc., Nov. 2019

[6] Gabriela Ciocarlie, "Tracking and analysis of causality at enterprise-level (TRACE)," ARFL-RY-WP-TR-2019-0337, SRI International, Mar. 2022

[7] OASIS, STIX Version 2.1 Specification, https://docs.oasis-open.org/cti/stix/v2.1/os/stix-v2.1-os.html, accessed Mar. 2022

[8] FireEye, "Naval Information Warfare Systems Command (NAVWAR) Awards FireEye First Place in Network Threat Detection Challenge," https://www.fireeye.com/company/press-releases/2021/naval-information-warfare-systems-command-navwar-awards-fireeye-first-place.html, accessed on Mar. 2022

[9] Center for Threat Informed Defense, Attack Flow, https://ctid.mitre-engenuity.org/our-work/attack-flow/", accessed on Mar. 2022

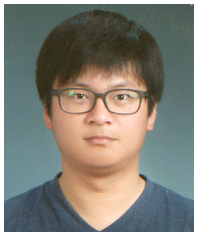
[10] MITRE, ATT&CK, https://attack.mitre

.org/, accessed on Mar. 2022
〔11〕 PEStudio, https://winitor.com, accessed on Mar. 2022
〔12〕 KISA, "TTPs #6 Target Watering Hole Attack Strategy Analysis," Sep. 2021, https://www.krcert.or.kr/filedownload.do?attack_file_seq=3277&attach_file_id=EpF3277.pdf, accessed on Mar. 2022
〔13〕 Alfonso Valdes and Keith Skinner, "Probabilistic alert correlation," International Workshop on Recent Advances in Intrusion Detection (RAID), pp. 54-68, Oct. 2001
〔14〕 Frédéric Cuppens, "Managing alerts in a multi-intrusion detection environment," Proceedings of the 17th Annual Computer Security Applications Conference, pp. 22-31, Dec. 2001
〔15〕 Hervé Debar and Andreas Wespi, "Aggregation and correlation of intrusion-detection alerts," International Workshop on Recent Advances in Intrusion Detection (RAID), pp. 85-103, Oct. 2001
〔16〕 Peng Ning, Yun Cui and Douglas S. Reeves, "Analyzing intensive intrusion alerts via correlation," International Workshop on Recent Advances in Intrusion Detection (RAID), pp. 74-94, Oct. 2002
〔17〕 Frédéric Cuppens and Alexandre Miege, "Alert correlation in a cooperative intrusion detection framework," Proceedings 2002 IEEE Symposium on Security and Privacy, pp. 202-215, May 2002
〔18〕 Faeiz Alserhani, Monis Akhlaq, Irfan U. Awan, Andrea J. Cullen and Pravin Mirchandani, "MARS: multi-stage attack recognition system," 2010 24th IEEE International Conference on Advanced Information Networking and Applications (AINA), pp. 753-759, Apr. 2010
〔19〕 Benjamin Morin, Ludovic Mé, Hervé Debar and Mireille Ducassé, "M2D2: A formal data model for IDS alert correlation," International Workshop on Recent Advances in Intrusion Detection (RAID), pp. 115-137, Oct. 2002
〔20〕 Steven T. Eckmann, Giovanni Vigna and Richard A. Kemmerer, "STATL: An attack language for state-based intrusion detection," Journal of computer security, vol. 10, no. 1-2, pp. 71-103, 2002
〔21〕 Bin Zhu and Ali A. Ghorbani, "Alert correlation for extracting attack strategies," International Journal on Network Security, vol. 3, no. 3, pp.244-258, Nov. 2006
〔22〕 Hanli Ren, Natalia Stakhanova and Ali A. Ghorbani, "An online adaptive approach to alert correlation," International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, pp.153-172, Jul. 2010
〔23〕 Samuel T. King, Z. Morley Mao, Dominic G. Lucchetti and Peter M. Chen, "Enriching intrusion alerts through multi-host causality," Proceedings of Network and Distributed System Security Symposium (NDSS), Feb. 2005
〔24〕 Samuel T. King and Peter M. Chen, "Backtracking intrusions," Proceedings of the 2003 Symposium on Operating Systems Principles, pp. 223 - 236, Oct. 2003
〔25〕 Md Nahid Hossain et al., "SLEUTH: Real-time attack scenario reconstruction from COTS audit data," 26th USENIX Security Symposium (USENIX Security 17), pp. 487-504, Aug. 2017
〔26〕 Sadegh M. Milajerdi et al., "HOLMES:

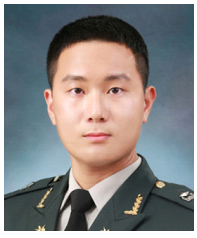
Real-time APT detection through correlation of suspicious information flows," 2019 IEEE Symposium on Security and Privacy, pp. 1137-1152, May 2019
[27] Chunlin Xiong et al., "CONAN: A practical real-time APT detection system With high accuracy and efficiency," IEEE Transactions on Dependable and Secure Computing, vol. 19, no. 1, pp. 551-565, Feb. 2020
[28] Kexin Pei, et al., "HERCULE: Attack story reconstruction via community discovery on correlated log graph," Proceedings of the 32nd Annual Conference on Computer Security Applications, pp. 583-595, Dec. 2016
[29] Jun Zeng, et al., "WATSON: Abstracting behaviors from audit logs via aggregation of contextual semantics," Proceedings of the 28th Annual Network and Distributed System Security Symposium (NDSS), pp. 1-18, Feb. 2021
[30] Cesar Ghali, Gene Tsudik and Ersin Uzun, "Needle in a haystack: Mitigating content poisoning in named-data networking," Proceedings of NDSS workshop on security of emerging networking technologies (SENT), Feb. 2014
[31] Yang Ji, et al., "Enabling refinable cross-host attack investigation with efficient data flow tagging and tracking," 27th USENIX Security Symposium (USENIX Security '18), pp. 1705-1722, Aug. 2018
[32] Shiqing Ma et al., "Kernel-supported cost-effective audit logging for causality tracking," 2018 USENIX Annual Technical Conference (USENIX ATC 18), pp.241-254, Jul. 2018
[33] MITRE ATT&CK, Data Source, https://attack.mitre.org/datasources, accessed on Oct, 2021
[34] OTRF, OSSEM Detection Model (DM), https://github.com/OTRF/OSSEM-DM, accessed on Oct, 2021
[35] Microsoft Sysinternals Sysmon, https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon, accessed on Jan. 2022
[36] SwiftOnSecurity, Sysmon Config, https://github.com/SwiftOnSecurity/sytsmon-config, accessed on Jan. 2022
[37] Neo23x0, auditd, https://github.com/Neo23x0/auditd, accessed on Jan. 2022
[38] Suricata, https://suricata.io, accessed on Jan. 2022
[39] Zeek, https://zeek.org, accessed on Jan. 2022
[40] Elastic, Elastic Common Schema, https://elastic.co/guide/en/ecs/1.12/index.html, accessed on Jan. 2022
[41] Elastic, Elastic Stack, https://elastic.co/elastic-stack, accessed on Mar. 2022
[42] MITRE, CAR (Cyber Analytics Repostory), https://github.com/mitre-attack/car, accessed on Mar. 2022
[43] SigmaHQ, Sigma, https://github.com/SigmaHQ/sigma, accessed on Mar. 2022
[44] Elastic, Elastic Detection Ruels, https://github.com/elastic/detection-rules, accessed on Mar. 2022
[45] pfSense, https://www.pfsense.org, accessed on Jan. 2022
[46] Pupy, https://github.com/n1nj4sec/ pupy, accessed on Jul. 2021
[47] PoshC2, https://github.com/netitude/PoshC2, accessed on Jul. 2021
[48] Metasploit, https://github.com/ rapid7/metasploit-framework, accessed on Jul. 2021
[49] MITRE Engenuity, ATT&CK Evaluations, https://attackevals.mitre-engenuity.org. accessed on Mar. 2022
[50] MITRE Center for Threat Informed

Defense, Adversary Emulation Library, https://github.com/center-for-threat-informed-defense/adversary_emulation_library, accessed on Jul. 2021

〈저자소개〉



조 성 영 (Sungyoung Cho) 정회원
2009년 8월: 한국과학기술원 정보통신공학과 학사
2013년 2월: 한국과학기술원 정보보호대학원 석사
2013년 9월~현재: 국방과학연구소 사이버/네트워크 기술센터 선임연구원
〈관심분야〉 정보보호, 사이버 상황인식, 사이버 보안 시각화, 사이버전



박 용 우 (Yongwoo Park) 정회원
2018년 2월: 고려대학교 사이버국방학과 학사
2018년 8월~현재: 국방과학연구소 사이버/네트워크 기술센터 현역연구원
〈관심분야〉 정보보호, 사이버 보안 시각화, 사이버전

이 경 식 (Kyeongsik Lee) 정회원
2009년 2월: 세종대학교 컴퓨터공학과 학사
2011년 2월: 고려대학교 정보경영공학과 석사
2011년 1월~현재: 국방과학연구소 사이버/네트워크 기술센터 선임연구원
〈관심분야〉 정보보호, 디지털 포렌식, 침해사고대응