JKSCI

한국컴퓨터정보학회논문지
Journal of The Korea Society of Computer and Information
Vol. 29 No. 5, pp. 1-10, May 2024
https://doi.org/10.9708/jksci.2024.29.05.001

# Cluster-Based Similarity Calculation of IT Assets: Method of Attacker's Next Targets Detection

Dongsung Kim*, Seon-Gyoung Shon**, Dan Dongseong Kim***, Huy-Kang Kim*

*Student, School of Cybersecurity, Korea University, Seoul, Korea
**Senior Researcher, Electronics and Telecommunications Research Institute, Daejeon, Korea
***Associate Professor, Cybersecurity, University of Queensland, St Lucia, Australia
*Professor, School of Cybersecurity, Korea University, Seoul, Korea

## [Abstract]

Attackers tend to use similar vulnerabilities when finding their next target IT assets. They also continuously search for new attack targets. Therefore, it is essential to find the potential targets of attackers in advance. Our method proposes a novel approach for efficient vulnerable asset management and zero-day response. In this paper, we propose the ability to detect the IT assets that are potentially infected by the recently discovered vulnerability based on clustering and similarity results. As the experiment results, 86% of all collected assets are clustered within the same clustering. In addition, as a result of conducting a similarity calculation experiment by randomly selecting vulnerable assets, assets using the same OS and service were listed.

▸ **Key words**: Common Platform Enumeration (CPE), Clustering, Similarity Measurement, Network Features, Network Scanner

## [요 약]

공격자들은 공격 대상인 IT 자산을 찾을 때 자신이 가지고 있는 유사한 취약점을 사용하는 경향이 있다. 따라서 IT 자산 중 표적이 될 수 있는 유사한 운영체제, 애플리케이션이 있을 때 이를 사전에 찾아내는 것이 중요하다. 본 논문은 효율적인 취약자산 관리 및 제로데이 대응을 위한 새로운 접근 방식을 제안한다. 해당 방법론은 클러스터링과 유사도 계산 결과를 기반으로 새로운 취약점이나 이미 발견된 취약점에 의해 감염될 가능성이 있는 IT 자산을 탐지하는 기능을 제공한다. 실험 결과, 수집된 전체 자산의 86%의 정확도로 클러스터의 목적에 맞게 분류되었으며, 무작위 자산을 선정하여 유사성 계산 실험을 한 결과 동일한 운영체제 및 서비스를 사용하는 자산이 나열됐다.

▸ **주제어**: Common Platform Enumeration (CPE), 클러스터링, 유사도 측정, 네트워크 특징, 네트워크 스캐너

• First Author: Dongsung Kim, Corresponding Author: Huy-Kang Kim
*Dongsung Kim (kimds4962@korea.ac.kr), School of Cybersecurity, Korea University
**Seon-Gyoung Shon (sgsohn@etri.re.kr), Electronics and Telecommunications Research Institute
***Dan Dongseong Kim (dan.kim@uq.edu.au), Cybersecurity, University of Queensland
*Huy-Kang Kim (cenda@korea.ac.kr), School of Cybersecurity, Korea University
• Received: 2024. 04. 29, Revised: 2024. 05. 16, Accepted: 2024. 05. 20.


http://www.ksci.re.kr pISSN:1598-849X | eISSN:2383-9945

# I. Introduction

2023년 사이버 공격 시도가 2022년에 비해 꾸준히 증가하고 있으며, 해킹 공격으로 인한 침해 사고도 지속적으로 증가하고 있다.

일반적으로 공격자는 목표를 달성하기 위해 다양한 취약점을 활용한다. 게다가 공격자가 하나의 IT 자산을 손상시킬 수 있는 취약점을 발견하면 공격 비용을 줄이기 위해 다른 장치에서 이를 악용하는 경향이 있다. 따라서 언더그라운드 마켓에서 익스플로잇 도구를 구입하면 하나의 취약점으로 공격 성공 확률이 높은 목표를 다수 탐색한다. 이로 인해 유사한 자산이 지속적으로 검색되어 공격에 발생하는 비용이 절감되므로 공격자의 목표가 된다[1].

Mirai Botnet은 널리 알려진 공격 사례 중 하나로, IoT 장비의 초기 비밀번호가 변경되지 않는다는 점을 악용하여 다양한 제조사에 심각한 피해를 입혔다. 이 멀웨어는 장치를 원격으로 제어할 수 있는 봇으로 감염시켜 공격자가 명령 및 제어 센터를 통해 장치를 조작할 수 있게 한다. 이와 같은 공격의 특징은 멀티 운영체제를 지원하는 소프트웨어 제품에서도 플랫폼이나 제품의 버전과 관계없이 유효할 수 있다.

본 논문에서는 공격자의 다음 목표를 찾기 위한 새로운 방법을 제안한다. 제안한 방법은 클러스터링 알고리즘으로 공격자의 목표가 되는 유사한 IT 자산을 도출한다. 그리고 유사성 측정을 이용해 취약한 자산과 동일한 운영체제 및 서비스를 사용하는 다른 자산을 찾아낸다. 또한, 운영체제나 서비스를 사용하는 네트워크 자산들을 그룹화하여 관리하는 방법을 제시한다. 해당 방법론으로 발견된 취약점의 확산을 막기 위해서 세운 보안 대책을 중요한 우선순위에 맞춰 처리하는 데 도움이 될 것으로 판단한다.

본 논문은 다음과 같은 구성으로 진행된다. 2장에서는 관련된 연구들을 소개한다. 3장에서는 제안된 방법론에 대해서 설명하고, 4장에서는 실제 수집한 네트워크 데이터를 바탕으로 실험한 결과를 설명한다. 5장에서는 4장까지의 결과를 토대로 고려해야 할 사항들에 대해 정리하였다. 마지막으로 6장에서는 본 논문의 결론을 설명한다.

# II. Related Works

취약한 장비를 관리하는 것은 관리자들의 가장 중요한 임무 중 하나다. 따라서 기업은 자산관리를 다양한 방식으로 수행하며 많은 자원과 시간을 사용한다. 그럼에도 불구하고 취약 자산을 악용한 해킹 사고는 해마다 증가하고 있으며, 각종 해킹 시도로 기업은 막대한 손실을 입고 있다. IBM 조사에 따르면 2023년 고객 데이터 유출 사고로 인한 해킹 사고 손실액은 445만 달러로 조사됐다[2]. 해킹 피해를 줄이기 위한 자산관리 연구는 다음과 같이 요약할 수 있다.

## 1. Asset Management

소비자에게 제품을 제공하는 다양한 분야의 기업들은 손실을 줄이기 위해 정기적으로 자산을 관리해야 한다. 따라서 많은 연구자는 자산을 효율적으로 관리하기 위해 다양한 실험을 수행하고 있다[3, 4].

최근 온라인 서비스를 제공하는 기업이 늘어나면서 IT 시스템에 대한 네트워크 공격도 점차 증가하고 있다. 그중, 일부 회사는 고객의 개인정보나 자산을 활용하여 서비스를 제공한다. 따라서 사고 후 처리가 이루어지면 자원 피해가 훨씬 더 클 수밖에 없다. 이를 방지하기 위해 네트워크 기능을 활용하여 취약 자산을 찾는 네트워크 서비스에 관한 연구가 활발히 진행되고 있다.

## 2. Scoring-Based Methods

Juan Fco. Gómez 등은 위험 기반 평가 방법으로 네트워크 유틸리티의 자산 중요도를 계산하여 유지 관리를 결정하는 방법론을 제안했다. 제안된 방법론을 통해 네트워크 토폴로지와 계층적 관점이 자산관리에 미치는 영향을 평가했다. 이를 통해 가치에 따른 자산의 중요도를 평가할 수 있었고, 전체 네트워크를 고려하여 유지 관리 작업 수준을 적절하게 조정하는 데 기여했다[5].

## 3. Clustering-Based Methods

Li과 Lin은 사용자 행동 데이터를 이용해 정상 행위와 이상 행위를 구별했다. 이때, 이상 행위가 발생한 자산을 관리할 수 있는 클러스터링 방법에서 일반적으로 존재하지 않는 자산의 Peer Group 데이터를 자동으로 생성하고 클러스터링하는 방법을 제안했다. 이 자산 분류는 자산 이상 탐지에서 Peer 분석의 중요한 방법으로 User Entity and Behavior Analytics (UEBA)를 통해 네트워크 자산을 클러스터링하여 효율성을 증명했다[6].

Everson, Cheng은 기업 자산을 관리하기 위해 공격자와 방어자의 관점에서 접근했으며 효율적인 자산 관리 방법으로 클러스터링 알고리즘을 제안했다. 공격 표면의 형태와 복잡성을 보다 정확하게 측정하기 위해 클러스터링 알고리즘을 사용했으며, 이를 통해 공격 표면을 구성하는

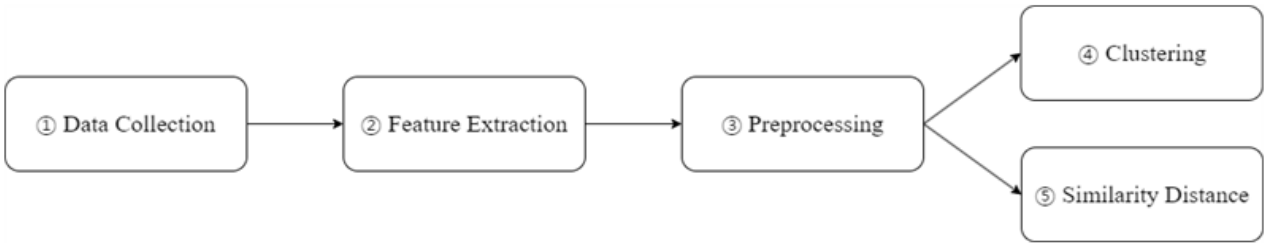


Fig. 1. Overview of Classification IT Assets

수많은 장치를 단순화함으로써 블루팀의 작업량을 줄였다. 이 방법론으로 방어 테스트를 수행할 때 적용 범위가 향상됐다[7].

Guillaume Dupont 등은 유사도 기반 네트워크 장치 클러스터링 방법을 제안했다. 먼저 자산의 특징으로 이름을 재정의한 후 비지도학습 클러스터링으로 자주 사용되는 Hierarchical Density-Based Spatial Clustering of Applications with Nosie (HDBSCAN)을 이용하여 분류하는 방법을 제시하였다. 그리고 수동으로 특징을 정의하는 머신러닝 기반 방식의 한계를 해결하기 위해 클러스터링을 제안했다. 또한, 화이트박스 접근 방식을 통해 자동 분류와 수동 검증을 결합하여 결과의 타당성과 신뢰성을 높였다[8].

### 4. System-Based Methods

Tovarňák 등은 취약한 자산을 식별하기 위해 그래프 기반 접근 방식을 사용했다. 그래프 모델에는 자산의 취약성에 대한 정보, 자산의 구성요소, Common Platform Enumeration (CPE) 및 Common Vulnerabilities and Exposure (CVE) 정보가 저장된다. 또한, 자산 구성요소의 변화에 따라 이 정보를 삽입하는 방법을 제안한다. 방법론은 자산의 CVE 취약점 및 CPE 구성을 포함하여 다양한 요소 조합에 대한 정보를 검색하는 방법을 제안한다. 이 방법으로 자산 구성이 지속적으로 변경되는 상황에서 취약점이 있는 자산의 CPE를 효율적으로 매핑하고 식별한다[9].

Diaz-Honrubia 등은 효과적인 네트워크 자산관리 및 비정상적인 패턴 감지 방법으로 Nmap을 개선한 자산 스캔 도구를 제안했다. 각 분산된 망에서 Nmap 기반의 스캔을 사용해 정보를 수집하고 통합하여 잠재적인 취약점과 위협을 식별했다. 또한, 방법론으로 포트, 서비스 정보에서 비정상적인 패턴을 보이는 자산을 분석하여 취약한 자산을 식별하는 연관 규칙 알고리즘을 만들어 제시했다. 알고리즘은 포트들의 트래픽 생성 시계열을 분석하여 비정상적인 트래픽 흐름이 감지되면 네트워크 관리자에게 경고한다[10].

## III. Proposed Method

본 논문은 Fig 1과 같이 1) Data Collection, 2) Feature Extraction, 3) Pre-processing, 4) Clustering, 5) Similarity Distance 5단계로 구성된다. 데이터 수집 단계에서는 네트워크 스캐너 도구를 사용하여 선택한 자산의 네트워크 특징을 수집한다. 특징 추출 단계에서는 수집된 네트워크 특징에서 자산을 식별하는데 사용되는 특징을 추출한다. 전처리 단계에서는 추출된 특징을 클러스터 및 유사도 계산을 위한 값으로 전처리 작업을 수행한다. 클러스터링 및 유사도 거리 단계에서는 클러스터링 및 유사도를 통해 계산된 값을 이용하여 유사한 특징을 가진 자산을 그룹화하고 분류한다.

### 1. Data Collection

데이터 수집 단계에서는 IT 자산 데이터를 네트워크 스캐닝 도구로 수집했다. 먼저 Criminal IP[11]로 제품의 사용 목적에 해당하는 label과 IP 정보를 수집한다. 수집한 제품의 IP 정보를 이용해 Nmap[12]으로 운영체제, 포트, 서비스, CPE 등 네트워크 정보를 수집한다.

### 2. Feature Extraction

특징 추출 단계에서는 데이터 수집 단계 이후에 진행된다. 앞서 수집된 데이터의 CPE, 운영체제, 서비스 등 중요한 특징들을 확인한다. 그리고 분류된 특징들은 각각 다른 개수로 구성되어 있으므로 주성분 분석을 사용하여 모두 같은 차원을 가지도록 차원 축소를 진행한다.

이때, 데이터로 수집되는 CPE는 자산을 효율적으로 관리하기 위해 NIST에서 제시하는 구조화된 표준 기술 시스템이다[13]. CPE의 주요 목적은 취약점의 영향을 받는 제품을 고유하게 식별하는 것이다. 이를 통해 조직의 관리자가 관련 제품 식별자를 알아내고 발생할 수 있는 문제를 식별할 수 있다. 따라서 네트워크 스캐너는 CPE 정보를 연결하여 자산을 관리하는 데 큰 역할을 할 수 있다.

CPE에는 자산에 대한 다양한 정보가 포함되어 있으며 애플리케이션(a), 운영체제(o), 하드웨어(h)인 3가지 부분

으로 나누어 분류된다. CPE 정보에는 Part, Vendor, Product가 포함되며, 제품을 고유한 값으로 설정하여 자산을 분류하는 방법을 사용한다.

CPE dictionary에는 2.3 버전에서 총 952,444개의 자산이 분류되어 있다. CPE 자산 정보에 대해서는 Table 1에서 볼 수 있듯이 Vendor에는 총 18,305개의 자산 정보가 포함되어 있고 Product에는 94,702개의 자산 정보가 포함되어 있다.

Table 1. CPE Dictionary v2.3 Asset Types

| Part | Vendor | Product |
|---|---|---|
| Application | 15,687 | 31,944 |
| OS | 1,323 | 24,856 |
| Hardware | 1,295 | 37,902 |
| Total | 18,305 | 94,702 |

### 3. Pre-processing

전처리 단계에서는 특징을 추출하고 클러스터링에서 사용하기 위한 값으로 변경하는 작업을 진행한다. 이때, 전처리를 통해 Nmap Frequency, OS Identifier, Service Identifier인 3가지 값을 계산한다. 첫 번째로 Nmap Frequency는 Nmap에서 수집한 데이터를 바탕으로 제품의 열린 포트 빈도를 제공한다. 따라서 공격자들의 목표로 자주 설정되는 포트들이 서로 그룹화되도록 설정할 수 있다. 두 번째로 OS Identifier는 전처리된 CPE 값에서 특정 값으로 인코딩된다. 특정 값은 Term frequency[14] 계산을 통해 조직 내에서 자주 사용하는 OS에 대해 높은 가중치를 설정한다. 마지막으로 Service Identifier는 사전 처리된 CPE를 사용하여 OS 식별자와 같은 특정 값을 인코딩한다. 마찬가지로 자주 사용하는 서비스에 가중치를 부여하기 위해 Term frequency를 사용했다.

### 4. Clustering

클러스터링 단계에서는 그룹화에 맞는 클러스터링을 선택한다. 따라서 K-Means, K-Means++, Hierarchical 알고리즘을 비교해 정확도가 높은 클러스터링을 선택한다. 각 클러스터링은 장점이 존재하는데, 먼저 K-Means 알고리즘은 데이터의 사전 정보가 필요하지 않다는 장점이 있다. 또한, 선형 시간 복잡도를 가지므로 대규모 데이터 세트를 처리하는 데 적합하다. 그리고 K-Means ++ 알고리즘은 K-Means 알고리즘에 중심점 무작위 선정 문제를 해결하여 최적의 그룹을 결정할 수 있다는 장점이 있다. 따라서 평균적으로 K-Means 알고리즘보다 빠른 속도로 계산해낼 수 있다. 마지막으로 Hierarchical 알고리즘은 사전에 중심점의 개수를 선정하지 않아도 되고 그룹 간 관계 파악이 가능하므로 적은 양의 데이터를 클러스터링하기에 좋다는 장점이 있다.

### 5. Similarity Distance

유사도 거리 단계에서는 자산의 유사도를 판단해 비슷한 제품을 순서대로 파악할 수 있도록 한다. 이때 거리 계산 알고리즘으로 Cosine similarity를 사용해 유사도 계산을 진행한다. 따라서 0에 가까울수록 유사한 제품으로 판단할 수 있으며, 앞서 전처리한 자산을 벡터화한 특성값으로 유사도 계산을 수행한다.

## IV. Experimental Result

### 1. Experiment Environment

본 실험에서는 데이터 수집 및 클러스터링 실험을 위해 다음과 같은 환경을 구성했다. 그리고 CPU Intel i7 3.6GHz 및 RAM 64GB를 사용하고 운영체제로 Windows 11을 사용하여 실제 IoT 제품의 네트워크 정보를 수집했다. 또한, 네트워크 스캐너로 Criminal IP를 통해 장비의 유형을 결정하는 label과 IP를 수집하고 Nmap 네트워크 스캐너를 사용하여 3일간 네트워크 특성 데이터를 수집했다.

### 2. Dataset

본 논문에서 실험을 위해 네트워크 스캐너를 사용하여 제품 유형을 추측하여 제시하는 데이터와 제품의 특징들을 수집했다. 먼저 Criminal IP를 통해 제품 목적에 맞는 label 정보와 IP 정보를 수집했다.

먼저 데이터의 실시간 변화에 장점이 있는 Criminal IP를 사용하여 IP와 사용 목적 label을 수집했다. 이때, 평가 기준을 설정하기 위해 제품이 하나의 클러스터 내에서 비중이 35% 이상에 도달하면 해당 클러스터에 제품의 label을 할당했다. 이것은 단일 클러스터가 여러 label을 포함할 수 있음을 의미한다.

우리는 CPE 정보를 제공하는 Nmap 스캐너를 이용해 네트워크 특징을 수집했다. Nmap 스캐너는 다른 스캐너보다 빠른 데이터 수집을 가능하게 하며 CPE 네트워크 자산을 제공하기 때문에 수집된 IP 주소에 대해 Nmap 스캐너를 이용해 자산 정보를 수집했다.

자산 사용량은 수시로 변경될 수 있으므로 데이터의 정확성을 위해 짧은 시간 내에 데이터를 수집했다. 이어 NAS, IP Camera, Switch, Printer 등 주로 공격자의 목표가 되는 IoT 제품들을 중심으로 실험을 진행했다.

Table 2. Number of Product Types

| Product | Number of |
|---|---|
| NAS | 117 |
| IP Camera | 123 |
| Switch | 118 |
| Printer | 117 |
| Total | 475 |

실험을 위해 데이터 세트는 네트워크 스캐너를 통해 직접 수집한 데이터로 진행한다. Table 2와 같이 수집된 데이터는 NAS 117개, IP Camera 123개, Switches 118개, Printer 117개로 총 475개의 데이터로 구성되어 있다. 그리고 각 자산의 네트워크 정보에는 포트, 운영체제, 서비스가 포함된다. 데이터 수집 중 제품 목적이 변경되면 label 값과 다른 정보 데이터가 수집될 수 있어 데이터 수집 기간을 짧고 빠르게 진행했다.

## 3. Pre-processing

데이터 전처리 단계에서는 클러스터링 및 유사성 계산이 용이하도록 자산 기능이 전처리 된다. 특히, Nmap에서 제공하는 Nmap Frequency 데이터를 활용하여 네트워크 기능 내의 포트 값에 대한 통찰력을 제공한다. 이 지표는 인터넷 검색 중에 포트가 열린 것으로 식별되는 빈도를 수량화한 수치다. 따라서 우리는 Nmap에서 제공하는 상위 5,000개 포트 중에서 Fig 2에 설명된 대로 가장 빈번하게 사용되는 10개 포트 번호를 실험에 사용한다. 결과적으로 Nmap Frequency를 통해 식별된 자주 사용되는 포트를 중심으로 효과적인 클러스터링을 기대할 수 있다.

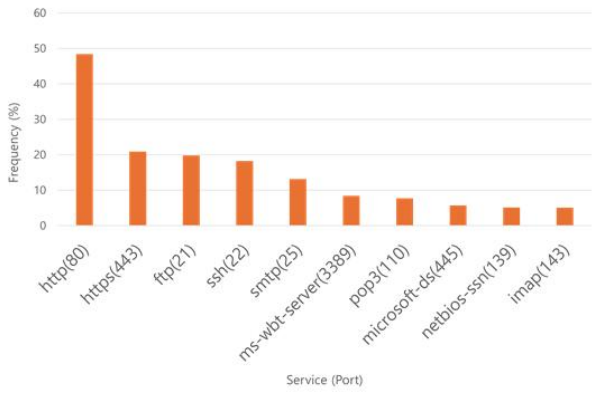


Fig. 2. The Most Used Port Numbers in Network

OS Identifier는 전체 자산 데이터 세트에서 운영체제 CPE를 추출한다. 그리고 추출된 값은 전처리된 CPE 값과 비교하여 특정 값으로 인코딩된다. 이때, 특정 값은 Term frequency 계산을 통해 결정된 조직 내에서 자주 사용되는 운영체제 버전에 높은 가중치를 할당한다. 이 방법으로 공격자가 다음 목표로 삼을 가능성이 큰 제품들을 그룹화할 수 있다.

Service Identifier는 전체 자산 데이터 세트에서 서비스 CPE를 추출하고 구성한다. 이 값도 마찬가지로 운영체제 식별자와 같은 접근 방식을 사용하여 전처리된 CPE를 기반으로 특정 값을 인코딩한다.

하지만 추출된 값에서 데이터들은 다양한 차원의 특성을 가지고 있다. 예를 들어, 서비스를 하나만 사용하는 자산과 두 개를 사용하는 자산을 비교하는 방법으로 같은 차원의 특성값을 만들 수 있는 차원 축소를 사용한다. 따라서 차원 축소를 통해 OS Identifier와 Service Identifier는 제품마다 동일한 차원을 가지게 된다.

이때, 차원 축소를 진행하며 적절한 차원 크기를 선택하는 것이 중요하다. 작은 차원을 사용하면 상당한 데이터 손실이 발생할 수 있으며 지나치게 큰 차원을 사용하면 비효율적일 수 있다. 따라서 우리는 최적의 크기를 식별하기 위해 데이터 손실이 적고 적당한 차원을 선택하여 사용했다.

차원의 수는 최적의 데이터 보존을 목표로 실험을 통해 결정한다. 실험 결과, Fig 3과 Fig 4에서 OS Identifier와 Service Identifier의 데이터 보존 비율을 나타낸다. 우리는 데이터 차원이 증가함에 따라 데이터 보존량이 0.9 이상인 값에서 거의 일정하게 유지되므로 해당 값을 임계값으로 선택하여 차원 축소를 진행했다.

이때, OS Identifier는 3차원, Service Identifier는 19차원을 사용할 때 데이터 보존량의 임계값인 0.9를 넘어선다. 결과적으로 Nmap Frequency, OS Identifier, Service Identifier로 이루어진 값은 클러스터링 및 유사성 측정을 계산하기 위한 23차원으로 구성된다.

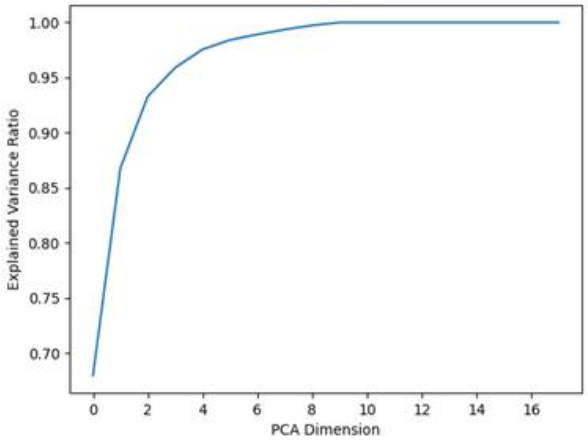


Fig. 3. OS Identifier PCA Explained Variance Ratio

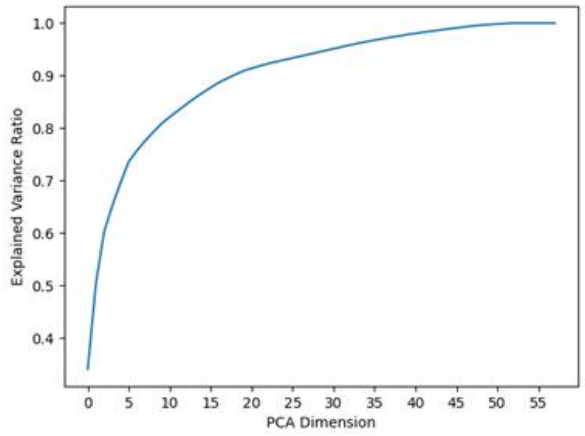


Fig. 4. Service Identifier PCA Explained Variance Ratio

## 4. Asset Clustering

각 클러스터는 이전에 수집된 4개의 IoT 장치로 구성된다. 실험 결과, 9개의 클러스터에 모든 유형의 장비가 포함되는 결과를 얻었으며, Fig 5를 통해 클러스터 분포를 확인할 수 있다. 그리고 우리는 실험을 통해 다음과 같은 클러스터 분포 특징들을 발견했다.

클러스터링 실험에서 같은 제품으로 분류된 경우에도 운영체제나 서비스가 다르면 동일한 클러스터에 속하지 않을 수 있다. 반대로, 서로 다른 제품으로 분류된 경우에 같은 운영체제나 서비스를 사용하면 동일한 클러스터에 속할 수 있다.

클러스터의 실험 결과에 대한 설명은 각 클러스터에서 다수의 비중에 속하는 제품 설명과 사용하는 운영체제 및 서비스의 내용을 포함하고 있으며 다음과 같이 나타난다.

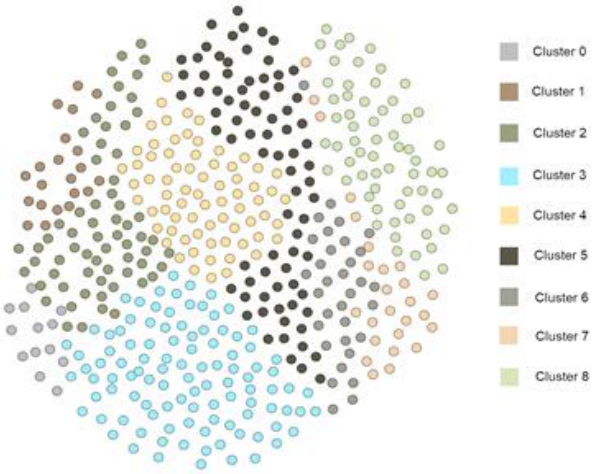


Fig. 5. Asset Distribution by Cluster

먼저 클러스터 0에서는 클러스터 내의 모든 스위치에서 Crestron OS와 Lighttpd 서비스를 사용하는 것으로 나타났다. 해당 클러스터는 수집한 데이터의 전체 제품 개수의 비율 중 가장 높은 22%를 차지한다.

클러스터 1은 주로 NAS와 Printer 구성되어 비슷한 비율의 클러스터를 형성했다. 해당 클러스터는 특정 서비스가 아닌 Crestron OS 및 HTTP에 연결된 80, 443, 8080과 같은 포트를 포함한다.

클러스터 2는 클러스터 내 제품 개수 비중의 97%를 차지하는 IP Camera로 이루어져 있다. 또한, 해당 클러스터 내의 모든 제품은 OpenSSH 및 Lighttpd를 사용하고 있다. 특히, IP Camera는 h.239 프로토콜을 서비스하는 포트 5001을 사용하는 것이 관찰됐다.

클러스터 3은 클러스터 내의 제품이 주로 Printer로 구성되어 있으며, 클러스터 내의 전체 제품 중 67%를 차지하고 있다. 클러스터의 모든 제품에는 포트 80이 열려 있고 Linux 운영체제를 사용한다. 따라서 해당 클러스터는 서비스가 아닌 포트와 운영체제를 기반으로 분류됐다.

클러스터 4는 클러스터 내의 제품이 주로 NAS로 구성되어 있으며 클러스터 내의 전체 제품 중 94%를 차지한다. 또한, 해당 클러스터 내의 모든 제품은 Apache HTTP 서버를 활용하고 있다. 특히 클러스터 내의 제품은 HTTP 포트에 바인딩된 특정 포트 번호로 구성되어 있다.

클러스터 5는 클러스터 내의 제품이 거의 Printer로 구성되며 클러스터 내의 전체 제품 중 79%를 차지한다. 또한, 해당 클러스터에서는 모든 제품이 OpenBSD를 운영체제로 사용하고 있다. 대부분 제품에서 Jetdirect 서비스를 제공하는 포트 9100을 활용한다.

클러스터 6은 서비스와 운영체제가 다른 제품들이 섞여 있다는 특징을 가지고 있었다. 결과적으로 해당 클러스터는 이상값 클러스터로 판단할 수 있다.

클러스터 7은 클러스터 내의 제품 중 대부분 IP Camera로 구성되어 있으며 해당 클러스터 제품의 76%를 구성하고 있었다. 또한, 해당 클러스터의 모든 제품은 OpenSSH 및 Crestron OS를 사용하며, 클러스터 내의 모든 IP Camera는 Lighttpd를 사용하고 다른 제품들은 Apache HTTP 서버를 사용한다.

마지막으로 클러스터 8은 클러스터 내의 모든 제품이 IP Camera로 구성되며 모두 Linux를 사용하고 있고 Lighttpd 서비스를 사용한다.

요약하면, Table 3에서 볼 수 있듯이, 클러스터별 제품 구성은 운영체제, 포트 및 서비스의 사용에 따라 결정된다. 또한, 운영체제가 동일할 경우 장치는 같은 클러스터로 그룹화되는 경향이 있다. 반대로 서비스나 포트가 일치하고 운영체제가 다르더라도 유사한 제품으로 같은 클러

Table 3. Clustering Results: Asset Cluster Description

| Cluster No. | Device | OS | Service |
|---|---|---|---|
| 0 | Switch | Crestron | Lighttpd |
| 1 | NAS, Printer | Crestron | HTTP |
| 2 | IP Camera | Linux | OpenSSH, Lighttpd, h.239 |
| 3 | Printer | Linux | jetdirect, HTTP |
| 4 | NAS | Linux | Apache HTTP, diskstation_manager |
| 5 | Printer | OpenBSD | Jetdirect |
| 6 | Outlier | - | - |
| 7 | IP Camera | Crestron | OpenSSH, Lighttpd, Apache HTTP |
| 8 | IP Camera | Linux | Apache HTTP |

스터 내에 속할 수 있다.

우리는 실험 결과를 통해 클러스터에 속한 장치를 그룹화하여 분류했다. 그룹화란 각 그룹 내 장치의 특성을 기준으로 장치를 분류하고 유사한 특성을 가진 그룹을 관리한다.

Table 4. Clustering Results: Groups and Cluster Numbers

| Group No. | Cluster No. | Description |
|---|---|---|
| 0 | 0 | Switch |
| 1 | 1, 4 | NAS |
| 2 | 2, 7, 8 | IP Camera |
| 3 | 3, 5 | Printer |
| 4 | 6 | Outlier |

클러스터의 그룹화는 Table 4와 같이 5개의 그룹으로 나눌 수 있다. 그룹 0에는 주로 Switch가 포함된다. 그룹 1에는 주로 NAS가 포함되어 있다. 특이하게도, NAS 사용자는 특정 포트 번호를 바인드해서 사용한다. 따라서 그룹 1에는 10,000이 넘는 포트가 많이 열려 있는 것을 볼 수 있다. 그리고 그룹 2에는 포트 9100이 열려 있는 대부분의 Printer와 다른 제품들이 포함되어 있다. 그룹 3에는 sIP를 통신 프로토콜로 사용하는 5060 포트가 열려 있는 대부분의 IP 카메라가 포함되어 있다. 마지막으로 그룹 4에는 나머지 그룹에 포함되지 않은 제품이 포함되어 있다. 이 그룹에는 다른 그룹에서 사용하지 않는 9개의 운영체제가 포함되어 있으며 주로 사용되지 않는 서비스를 포함하고 있다.

실험 결과 K-Means 클러스터링에서 전체 제품 label은 86%의 정확도를 얻을 수 있다. Printer의 정확도가 97%로 가장 높으며, IP Camera의 정확도가 78%로 가장 낮게 측정됐다.

## 5. Clustering Comparison

본 논문에서는 K-Means, K-Means++, Hierarchical의 세 가지 클러스터링 성능을 비교하는 실험을 진행했다. 정확도 비교 실험은 3가지 클러스터링 모두 앞선 실험의 정확도 측정과 동일하게 진행하였다. 이때, 클러스터링을 위한 하이퍼 파라미터로 K-means와 K-means++는 iteration = 1000, 중심 설정에서 random_state를 사용해 2개부터 20개의 중심점을 갖는 클러스터의 정확도를 계산했다. 그리고 Hierarchical 클러스터링에서는 하이퍼 파라미터로 Euclidean과 ward 방식을 사용해 2개부터 20개의 중심점을 갖는 클러스터링 정확도를 계산했다.

실험 결과 전체 클러스터 수가 8개를 초과할 때 정확도 평균은 85%였다. 결과적으로 K-Means 클러스터링은 중심의 무작위 초기화로 인해 매번 다른 결과를 생성한다는 단점이 있다. 따라서 K-means++ 클러스터링[15]은 결과가 상대적으로 일정하므로 이 문제에 대응하는 안정적인 알고리즘으로 평가받고 있다. 그리고 Hierarchical 클러스터링[16]은 같은 클러스터링 결과를 지속적으로 생성하는 특징이 있으므로 소규모 데이터 세트에 특히 유리하다.

Table 5. Cosine Similarity Distance to Target Asset

| Asset | OS | Service | Ports | Distance |
|---|---|---|---|---|
| Target | Linux | Apache HTTP server | 80, 443 | - |
| Asset A | Linux | Apache HTTP server | 80, 1720, 5000 | 5.30e-05 |
| Asset B | Linux | Apache HTTP server | 80, 1720, 5000, 7000 | 5.30e-05 |
| Asset C | Linux | Apache HTTP server | 80, 443, 1720, 5000, 7000, 8000 | 5.57e-04 |
| Asset D | Linux | Apache HTTP server | 80, 90, 554, 1720, 1723, 5000, 6000, 8000, 8100, 8200, 8300, 8400 | 9.51e-04 |
| Asset E | Linux | Apache HTTP server | 21, 80, 443, 1720, 5000, 5001 | 1.02e-02 |
| Asset F | Linux | Apache HTTP server | 21, 80, 1720, 5000 | 1.02e-02 |

이러한 클러스터링 방법의 정확성을 평가하기 위해 비교 분석을 수행하고, 그 결과는 Fig 6에서 볼 수 있다.

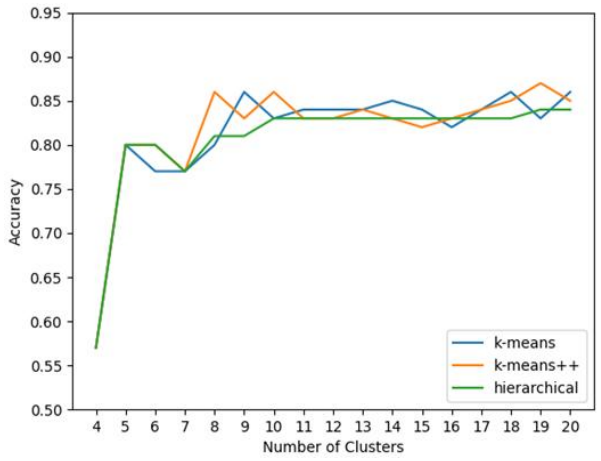


Fig. 6. Comparison of Clustering Accuracy

### 6. Asset Similarity

Cosine Similarity Distance는 일반적으로 고차원 공간에서 두 벡터 간의 유사성을 정량화하기 위해 사용하는 알고리즘이다. 주로 데이터 분석 및 정보 검색에서 사용되며, 알고리즘의 벡터 사이 각도의 cosine을 계산하여 0과 2 사이의 값을 얻는다.

본 논문에서 Cosine Similarity Distance는 특정 기능이나 속성을 기반으로 다양한 제품이나 클러스터를 나타내는 데이터 간의 유사성 또는 차이점을 평가하는 데 사용하고 있다. 코사인 유사도 값이 클수록 비슷하다는 것을 의미하고, 값이 낮을수록 비슷하지 않다는 것을 의미한다.

해당 실험은 취약점이 발견된 기기의 기능을 가상으로 설정하여 진행했다. 가상의 취약점은 CVE-2021-42013이며 Apache HTTP 서버에서 발생한다. 이 취약점은 특정 버전의 Apache HTTP 서버를 사용할 때 나타나며 해당 취약점을 악용하여 서버 정보를 얻어낼 수 있다. 실험을 위한 해당 제품에는 운영체제인 Linux와 포트 80 및 443에서 Apache HTTP 서비스를 수행하는 역할을 부여한다.

실험 결과 목표 자산과 유사하게 측정된 자산들의 특징은 Table 5에서 볼 수 있고, 목표로 지정된 가상의 자산과 유사한 순서로 결과를 나열됐다. 유사한 자산으로 나열된 제품은 HTTP 사용을 위해 포트 80 및 443이 열려 있고 운영체제인 Linux와 Apache HTTP 서버를 사용하는 것을 확인할 수 있다. 또한, 많은 NAS 제품이 HTTP 서비스를 포트 5000으로 리디렉션하므로 유사한 자산에 포트 5000이 열려 있는 것을 볼 수 있다.

### 7. Evaluation

본 논문에서는 실루엣 계수를 이용해 데이터의 클러스터링이 잘 이루어져 있는지 평가한다. 실루엣 계수는 데이터의 클러스터링 또는 그룹화 품질을 평가하기 위해 클러스터 분석 및 기계 학습에서 사용하는 통계 측정항목이다. 이를 통해, 데이터 내에서 서로 다른 클러스터가 잘 분리되어 있는지에 대한 정량적 측정값을 확인할 수 있다.

각 클러스터의 실루엣 계수는 가장 가까운 이웃 클러스터의 데이터에 대한 근접성과 동일한 클러스터 내의 다른 데이터에 대한 근접성을 기준으로 계산된다. 값의 범위는 -1부터 1까지로 계산되며, 값이 클수록 데이터가 잘 클러스터되어 있고 이웃 클러스터보다 해당 클러스터에 더 가깝다는 것을 의미한다. 반대로, 값이 낮을수록 데이터가 다른 클러스터에 더 적합할 수 있음을 의미한다.

따라서 각 클러스터의 실루엣 계수 평균을 계산하여 나온 결과로 값은 Fig 7에서 볼 수 있다. 전체적으로 계산된 클러스터의 실루엣 계수는 0.8로 수렴하여 적절한 분포를 보인다는 것을 알 수 있다.

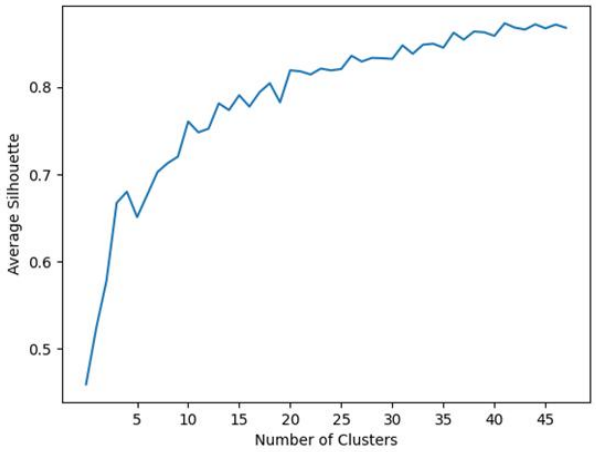


Fig. 7. Silhouette Coefficients for Number of Clustering

## V. Discussion

### 1. Limitations

우리는 실험 단계에서 각 제품의 클러스터링을 정의할 때 문제를 발견했다. 본 논문에서 수집한 데이터는 네 가지 IoT 제품 유형으로 구성되어 있다. 그러나 제품을 이용하는 일부 사용자 중에서 다른 목적으로 장치를 사용하는 경우가 존재했다. 예를 들어, NAS와 Switch가 HTTP를 활용하여 개인 서버로 사용하는 데이터를 찾을 수 있었다. 따라서 NAS와 Switch라는 label에도 불구하고 클러스터

링 실험에서는 유사한 다른 제품으로 분류되었다.

또한, 이전 연구들에서 강조된 것처럼 정확한 데이터를 수집하는 데 있어서 Nmap의 한계가 명확히 존재한다는 것을 알 수 있다[17]. 결과적으로 우리는 데이터 수집을 위해 더 정확한 네트워크 스캐너가 필요하다는 것을 알 수 있었다. 그리고 실시간으로 사용하는 자산은 순간적으로 용도 변경이 이루어질 수 있다. 정보를 수집하는 동안 제품의 전원이 꺼지거나 새로운 서비스가 설치될 수 있으므로 최신 label이 붙은 제품으로부터 데이터를 빠르게 수집하는 것이 필요하다.

### 2. Assumptions

공격자는 취약점 사용을 하나의 제품에만 사용하지 않는다. 오히려 여러 제품에서 순차적으로 이를 활용하는 경향이 있다. 또한, 자동화된 악성 코드를 사용하여 수많은 대상을 식별하고 자산을 공격한다. 이러한 위험을 완화하려면 조직이 소유한 자산을 다양화하는 것이 중요하다. 그렇지 않으면 조직이 소유하는 모든 자산이 동일한 운영체제나 서비스를 사용하는 경우 자산에 취약점이 발생하면 잠재적으로 모든 자산이 취약한 대상이 되어 심각한 피해를 입을 수 있다. 따라서 자산의 다양성을 선택하는 것은 생존 가능성에 대한 전략적 이점이 될 수 있다.

결론적으로 우리는 공격자의 다음 목표를 찾기 위한 접근 방식을 제안했다. 이 방법론은 자동화된 IoT Worm을 탐지하고 차단하는 데 효과적인 것으로 입증됐다. IoT Worm의 공격 특징인 비슷한 취약 자산을 인식하여 자동화된 코드 전파를 사용하는 것에 효과적으로 방지한다.

우리의 접근 방식은 액세스 가능한 모든 IoT 제품에서 단일 취약점을 악용하여 권한을 얻는 것을 중심으로 연구를 진행하기 때문에 동일한 취약점이 발생한 장치를 신속하게 식별하여 우리의 방법론을 이용해 대응할 수 있다. 하지만 이 접근 방식은 특정 가정을 전제로 수행되기 때문에 공격자가 임의의 대상을 설정하여 공격하는 경우에는 적합하지 않을 수 있다. 결과적으로 우리는 이러한 한계점을 해결하기 위해 이 방법론을 확장할 필요가 있다.

## VI. Conclusions

기존 연구에는 공격이 발생한 제품의 취약점이나 관리할 수 있는 플랫폼을 이용해 자산의 취약성을 관리했다. 우리가 제안한 방법론은 공격이 발생하기 전에 IT 인프라 내에서 취약한 자산을 식별하는 효율적인 수단을 제시한다. 또한, 악의적인 공격자의 잠재적인 표적을 신속하게 탐지하는 것에 통찰력을 제공한다. 이 방법론은 체계적인 자산 식별 및 데이터 수집 프로세스와 관련 자산 특징 추출의 순서로 진행된다. 추출된 자산의 특징을 전처리하고 활용하여 유사성을 식별하는 자산을 그룹화하고 목록화하기 위해 클러스터링을 수행한다. 실험 결과는 자산 클러스터링의 정확도가 86%인 것으로 나타났다. 또한, 유사성 계산을 통해 대상으로 지정된 자산을 비교한 결과, 유사한 서비스를 사용하는 자산은 가까운 거리의 자산과 연관되어 있음을 관찰하여 우리 접근 방식의 효율성을 확인했다. 향후 연구에서는 오탐을 완화하고 공격자가 무작위 자산을 찾고 표적으로 삼는 방법을 조사하기 위한 방법론을 개선한다.

## ACKNOWLEDGEMENT

This work was supported by the Korea Research Institute for Defense Technology Planning and Advancement(KRIT) - Grant funded by Defense Acquisition Program Administration (DAPA)(KRIT-CT-21-037)

## REFERENCES

[1] L. Allodi, “Economic factors of vulnerability trade and exploitation,” in Proceedings of the 2017 ACM SIGSAC conference on computer and communications security, pp. 1483-1499, October 2017. DOI: https://doi.org/10.1145/3133956.3133960

[2] “Cost of a data breach 2023 a million-dollar race to detect and respond.” https://www.ibm.com/reports/data-breach, 2022.

[3] M. Kozlovszky, L. Kovacs, M. Torocsik, G. Windisch, S. Acs, D. Prem, et al., "Cloud security monitoring and vulnerability management", Proc. IEEE 17th Int. Conf. Intell. Eng. Syst. (INES), pp. 265-269, Jun. 2013. DOI: https://doi.org/10.1007/978-3-319-28091-2_11

[4] M. Nyanchama, “Enterprise vulnerability management and its role in information security management.,” Inf. Secur. J. A Glob. Perspect., vol. 14, no. 3, pp. 29-56, July. 2005. DOI: 10.1201/1086.1065898X/45390.14.3.20050701/89149.6

[5] Juan Fco Gómez, Pablo Martínez-Galán, Antonio J. Guillén and Adolfo Crespo, "Risk-Based Criticality for Network Utilities Asset Management", IEEE Transactions on Network and Service

Management, March. 2019. DOI: 10.1109/TNSM.2019.2903985

[6] A. Li and D. Lin, “Generating interpretable network asset clusters for security analytics,” in 2018 IEEE International Conference on Big Data (Big Data), pp. 2972-2979, IEEE, December. 2018. DOI: 10.1109/BigData.2018.8622077

[7] D. Everson and L. Cheng, “Network attack surface simplification for red and blue teams,” in 2020 IEEE Secure Development (SecDev), pp. 74-80, IEEE, September. 2020. DOI: 10.1109/SecDev45635.2020.00027

[8] G. Dupont, C. Leite, D. R. dos Santos, E. Costante, J. den Hartog, and S. Etalle, “Similarity-based clustering for iot device classification,” in 2021 IEEE International Conference on Omni-Layer Intelligent Systems (COINS), pp. 1-7, IEEE, August. 2021. DOI: 10.1109/COINS51742.2021.9524201

[9] D. Tovarňák, L. Sadlek and P. Čeleda, "Graph-based cpe matching for identification of vulnerable asset configurations", 2021 IFIP/IEEE International Symposium on Integrated Network Management (IM), pp. 986-991, May. 2021.

[10] Diaz-Honrubia, A. J., Herranz, A. B., Santamaría, L. P., Ruiz, E. M., Rodríguez-González, A., Gonzalez-Granadillo, G., ... & Xenakis, C., “A trusted platform module-based, pre-emptive and dynamic asset discovery tool,” Journal of Information Security and Applications, vol. 71, p. 103350, December. 2022. DOI: https://doi.org/10.1016/j.jisa.2022.103350

[11] Criminal ip.” https://www.criminalip.io/en, 2022.

[12] G. F. Lyon, “Nmap.” https://nmap.org, 1997.

[13] E. a. Mary C. Parmelee, “Common platform enumeration: Name matching specification version 2.3,” RFC 7696, NIST, August 2011.

[14] W. Zhang, T. Yoshida, and X. Tang, “A comparative study of tf* idf, lsi and multi-words for text classification,” Expert systems with applications, vol. 38, no. 3, pp. 2758-2765, March. 2011. DOI: https://doi.org/10.1016/j.eswa.2010.08.066

[15] A. Kapoor and A. Singhal, “A comparative study of k-means, k-means++ and fuzzy c-means clustering algorithms,” in 2017 3rd international conference on computational intelligence & communication technology (CICT), pp. 1-6, IEEE, February. 2017. DOI: 10.1109/CIACT.2017.7977272

[16] F. Nielsen and F. Nielsen, “Hierarchical clustering,” Introduction to HPC with MPI for Data Science, pp. 195-211, February. 2016. DOI: https://doi.org/10.1007/978-3-319-21903-5_8

[17] D. W. Richardson, S. D. Gribble, and T. Kohno, “The limits of automatic os fingerprint generation,” in Proceedings of the 3rd ACM workshop on Artificial intelligence and security, pp. 24-34, October. 2010. DOI: https://doi.org/10.1145/1866423.1866430

## Authors

Dongsung Kim received his BS degrees in Computer Science and Information Security Convergence from Korea University, Republic of Korea, in 2014 and 2020. He is currently pursuing MS degree in Graduate School of Information Security from Korea University, Republic of Korea. His current research interests are in anomaly detection using data-driven.

Seon-Gyoung Shon received the B.S. and M.S. degrees in Computer Science from Chonnam National University, Korea, in 1999 and 2001, respectively. She is currently a principal researcher at Electronics and Telecommunications Research Institute, Korea. Her main research interests include cyber security, security threat response and cyber warfare.

Dan Dongseong Kim was a Senior Lecturer/Lecturer in cybersecurity at The University of Canterbury from August 2011 to December 2018. From June 2008 to July 2011, he was a Postdoctoral Researcher at Duke University. Dan Dongseong Kim is an Associate Professor in cybersecurity at The University of Queensland, Australia since January 2019. His research interests include automated cybersecurity modeling and analysis for the Internet of Things, cloud computing, and moving target defense.

Huy-Kang Kim received a B.S. degree in Industrial Management, M.S. degree in Industrial Engineering and Ph.D. degree in Industrial and System Engineering in Korea Advanced Institute of Science and Technology (KAIST), Republic of Korea. He is a serial entrepreneur; he founded A3 Security Consulting in 1999 and AI Spera, the data-driven cyber threat intelligence service company in 2017. Currently, he is a professor in the School of Cybersecurity, Korea University. His recent research is focused on anomaly detection in the intelligent transportation system, online gaming and internet banking by using data analytics and machine learning techniques.