논문 21-46-06-07 The Journal of Korean Institute of Communications and Information Sciences '21-06 Vol.46 No.06
https://doi.org/10.7840/kics.2021.46.6.983

소프트웨어 정의망에서 공격 그래프 기반의 침입감내 기술

김 혜 진◆, 윤 승 현*, 김 성 환*, 임 혁°

Attack Graph Based Intrusion Tolerance Method in Software-Defined Networks

Hyejin Kim◆, Seunghyun Yoon*, Sunghwan Kim*, Hyuk Lim°

요 약

사이버 보안 침해가 발생한 네트워크 자원은 네트워크 정보와 취약점 특성이 악용되어 주변 자원에 이차적인 피해를 줄 수 있다. 침입감내 기술은 이와 같은 침해 상황에도 제공 중인 서비스를 지속적으로 유지시킬 수 있다. 본 논문에서 제안하는 침입감내 기술은 베이지안 공격 그래프 (Bayesian Attack Graph, BAG)를 활용하여 차후 악용될 수 있는 자원들을 추정한 후, 소프트웨어 정의망 (Software-Defined Networks, SDN) 제어기로 추정된 자원들의 IP 주소, MAC 주소 등을 변경시켜 공격의 확산을 막는다. 또한, 소프트웨어 정의망과 가상화 기술을 활용하여 공격 피해가 예상되는 자원들만을 복구시킴으로써, 제공 중인 서비스의 성능 저하를 최소화시킨다. 제안하는 침입감내 기술의 효율성을 검증하기 위해, 베이지안 공격 그래프에서 공격 경로를 무작위로 생성한 다음, 침입감내 알고리즘을 적용하여 공격 경로를 예측할 수 있는지 비교 분석하였다. 실험 결과를 통해, 제안하는 침입감내 기술이 공격 피해 가능성이 높은 노드를 성공적으로 추정하고 해당 자원의 네트워크 주소 변경 및 서비스 복구를 선별적으로 수행하여, 적은 침입감내 수행 오버헤드로 높은 서비스 지속성을 제공할 수 있음을 보였다.

Key Words : Intrusion tolerance, attack graph, software-defined networking, network isolation technology, network address reassignment

ABSTRACT

The intrusion tolerance method can provide sustainable services in case of cybersecurity breaches. We propose a network intrusion tolerance method that identifies compromised nodes by utilizing Bayesian attack graphs (BAGs) and changes the IP/MAC addresses of the nodes to prevent the spread of attacks. Furthermore, we leverage software-defined networks (SDN) and virtualization technologies to recover only the resources that are expected to be compromised in order to minimize the performance degradation of the services. To validate the efficiency of the proposed intrusion tolerance technique, we generate a set of random attack paths and investigate how accurately the attack paths can be predicted by the proposed method in comparison with other methods. Experimental results show that the proposed intrusion tolerance method can successfully estimate attack-prone nodes and selectively perform the network address change and the service recovery of the nodes, resulting in high service persistence with low intrusion-tolerance overhead.

※ 본 연구는 국방과학연구소의 연구비 지원으로 수행되었습니다 (UD200007ED).
◆ First Author : Gwangju Institute of Science and Technology (GIST), AI Graduate School, hyejinkim@gist.ac.kr, 학생회원
° Corresponding Author : Gwangju Institute of Science and Technology (GIST), AI Graduate School, hlim@gist.ac.kr, 정회원
* Gwangju Institute of Science and Technology (GIST), School of Electrical Engineering and Computer Science, seunghyunyoon@gm.gist.ac.kr, 학생회원; sunghwankim@gist.ac.kr, 학생회원
논문번호 : 202011-293-B-RU, Received November 25, 2020; Revised January 8, 2021; Accepted March 1, 2021

Ⅰ. 서 론

침입감내 기술은 사이버 공격자가 네트워크에 침입하여 중요 자원을 위협하여도 이를 감내하고 정상적으로 서비스를 제공하기 위해 개발되었다. 네트워크에 침입한 공격자는 네트워크 자원을 침해하여 해당 자원의 네트워크 및 취약성 정보를 확보하고, 이를 악용하여 주변 자원에 이차적인 위협을 가할 수 있다. 이러한 침입으로 발생되는 피해를 최소화하기 위해서는 공격 대상이 될 수 있는 시스템을 예측하여, 우선순위를 두고 관리해 주는 과정이 필요하다.

시스템 내부의 공격 가능한 취약점은 공격 트리나 공격 그래프 기법을 활용하여 정량적으로 분석할 수 있다[1]. 특히, 조건부확률 계산에 기반하여 취약점들의 연관성을 분석하는 베이지안 공격 그래프 (Bayesian Attack Graph, BAG)는 그래프 추론 과정을 통해 침해 결과로부터 침해의 원인이 되는 노드를 추정할 수 있게 한다[2]. 이와 같은 모델링 기법들에 사용되는 취약점 정보는 취약점 식별번호 및 세부 정보를 제공하는 CVE (Common Vulnerabilities and Exposures), 취약점을 분석하여 표준화된 점수를 제공하는 CVSS (Common Vulnerability Scoring System) 등으로부터 제공받을 수 있다[3,4].

한편, 침입자가 이미 확보한 네트워크 정보는 추가적인 공격에 활용될 위험성이 있기 때문에 효과적인 침입 감내를 위해서는 해당 네트워크의 설정을 변경함으로써 침입자가 가지고 있는 네트워크 정보를 무효화시키는 것이 바람직하다. 기존의 네트워크에서는 네트워크의 재설정이 매우 제한적이고 까다로운 반면 소프트웨어 정의망 (Software-Defined Networks, SDN)에서는 SDN 제어기를 통해 즉각적으로 네트워크 재설정을 수행할 수 있다[5]. 특히, 소프트웨어 정의망의 이러한 특성을 활용한 사이버 침해 방어 기술로 MTD (Moving Target Defense) 기술을 예로 들 수 있다[6].

MTD는 침해 이전의 공격자가 시스템 파악을 어렵게 하기 위해 시스템의 주요 속성인 IP 주소, MAC 주소, 라우팅 경로 등을 동적으로 변화시키는 기술이다. MTD 기술을 활용하면, 공격 대상 노드의 네트워크 설정이 능동적으로 변하므로 공격자에게 혼란을 주어 시스템을 보호할 수 있게 한다. 하지만 빈번한 MTD의 적용은 시스템에 부하를 주어 성능 저하를 일으키므로 적용 시간을 고려하는 것이 중요하다.

본 논문은 사이버 침해 상황에도 서비스 지속성을 보장할 수 있는 공격 그래프 기반의 침입감내 기술을 제안한다. 제안하는 침입감내 기술은 침입이 탐지된 노드 정보를 기반으로 차후 악용될 가능성이 있는 노드를 찾고, SDN 기술을 활용하여 해당 노드들에 무작위 네트워크 주소 재설정을 수행한다. 이때, 주소 변경 대상 노드는 공격 예측 경로 안에 있는 노드로 제한하였으므로, 과도한 네트워크 설정의 변경으로 인하여 시스템 서비스의 성능이 저하되는 것을 방지할 수 있다.

Ⅱ. 관련 연구

2.1 침입감내 기술

네트워크 침입감내 기술은 사이버 침해 상황에서 정보시스템을 정상 작동시키기 위해 개발되었으며, 이중화 시스템 (replication), 자원 고립 (isolation) 등의 구현 방법이 존재한다. 이중화 시스템은 가상화 기술을 활용하여 침해 상황에서 손상된 서비스를 정상 서비스로 대체시켜 서비스 지속성을 보장한다. 이와 다르게, 자원 고립 방법은 침해가 발생한 시스템의 네트워크 연결을 끊어, 침해 피해를 최소화한다. 이외에도 다양한 구현 방법을 사용한 침입감내 기술이 국내외 활발히 발표되고 있다.

Reynolds *et al.*은 소프트웨어 취약점을 사용하는 공격에 감내하기 위하여 HACQIT (Hierarchical Adaptive Control of Quality of service for Intrusion Tolerance)를 제안하였다[7]. HACQIT는 상용 방화벽과 침입 차단 소프트웨어를 실행하는 게이트웨이 컴퓨터, 중요한 응용 프로그램이 실행되는 Primary 서버와 Backup 서버, 그리고 전체 시스템의 모니터링을 수행하는 제어 컴퓨터로 구성된다. Primary 서버와 Backup 서버는 동일한 서비스들로 구성되어 있으며, 이 두 개의 서버는 LAN (Local Area Network)이 아닌 OOB (Out-Of-Band) 링크를 통해 제어 컴퓨터로 연결되어 있어, 외부 망으로부터 웜이나 이메일 바이러스의 전파가 불가능하다. 비교적 구현이 간단하나, 침입 탐지 기능이 미약하다는 한계가 있다. Verissimoet *et al.*은 MAFTIA (Malicious and Accidental-Fault Tolerance for Internet Applications)를 제안하였다[8]. MAFTIA는 결함 감내, 분산 컴퓨팅, 암호 해독법, 정형화 증명, 그룹 통신 프로토콜, 접근 제어 등의 여러 분야 지식을 사용한 포괄적인 접근 방법의 침입감내 시스템이다. MAFTIA는 공격 피해에 대한 복원력 (resilience)을 갖추어, 대규모 네트워크 인프라 서비스를 신뢰할 수 있도록 하는 것을 목표로 한다. 다양한 기법들을 활용하기 때문에 침입감내 기

능이 뛰어나나, 구현 비용이 높다는 단점이 있다.

침입감내 시스템은 네트워크 자원의 종류와 제공되는 서비스에 따라 보안 위험도가 달라질 수 있다. Heo *et al.*은 기존 침입감내 시스템이 SPOF (Single Point Of Failure) 문제를 해결하기 위해 동일한 서비스를 redundant 요소로 배치하는 것은 공통된 취약점 (common vulnerability)을 노출시킨다고 지적하였다. 이러한 환경은 공격하기에 더 적합하므로 redundant 요소 추가 시 보안 위험도를 최소화하기 위한 소프트웨어를 찾는 방식을 제시하였다[9]. 이와 유사하게, Gorbenko *et al.*은 운영체제별로 취약점의 분포를 분석하고 이에 근거하여 침입감내 시스템 내 자원들의 운영체제 조합을 제안하였다[10].

한편, Yoon과 Heo는 서비스 지속성을 위한 정상상태 모델링 기반의 침입감내 기술을 제안하였다[11]. 제안된 기술은 시스템의 정상상태 특징을 사전에 파악하고, 이의 범주에 벗어난 시스템 상태가 인지되면 이에 대한 대응을 수행한다. 가상화 기술을 활용하여 서비스와 관련된 프로세스와 라이브러리 간 상호작용을 모니터링하고 침입을 탐지한다. 비정상 상태의 종류에 따라 적절한 대응을 위한 다양한 함수를 제안하였다.

상기 침입감내 기술은 단순히 침입을 탐지하고 차단하는 것에 집중하는 반면, 본 논문에서 제안하는 침입감내 기술은 추가적으로 발생할 수 있는 우회 공격에 대응한다. 기존의 HACQIT는 게이트웨이 컴퓨터에서 방화벽과 침입 차단 시스템을 실행함으로써 침입에 대응하나, 이와 같은 방법은 상용 소프트웨어에 대한 지식을 가지고 있는 공격자의 우회 공격으로부터 시스템을 보호하기 어렵다. MAFTIA 시스템 역시, 공격자가 이미 확보한 정보를 통해 추가적인 공격을 시도하는 경우는 설명하지 않는다.

2.2 취약점 탐지 기술

정보시스템 내 취약점은 사용 중인 운영체제와 소프트웨어의 버전이 취약한 경우 발견되며, 일반적으로 취약점 스캔 도구를 사용하여 탐지된다. 취약점 스캔 도구는 네트워크를 스캔하여 취약점 정보와 보안 대응법 등을 제공할 수 있으나, 여러 개의 취약점을 조합하는 공격의 대응 방법은 설명하지 못한다. 이러한 종류의 공격을 분석하기 위해 공격 시나리오를 예측 및 분석할 수 있는 공격 표현 모델이 제안되었다.

공격 표현 모델은 네트워크상의 목표 노드에 공격자가 접근할 수 있는 공격 경로를 나타낼 수 있고, 현재 보안성과 취약성에 대해 파악할 수 있다. 이는 가능한 모든 공격 경로를 사전에 파악하고 최적의 공격 경로를 찾을 수 있도록 하며, 공격 표현 모델을 통해 최종 목표 노드에 도달하기 위한 취약점 공격 순서를 확인할 수 있다. 특히, 공격 시나리오를 표현하고 평가하기 위하여 공격 트리와 공격 그래프가 보편적으로 사용되고 있다.

공격 트리는 트리 데이터 구조 기반의 공격 표현 모델로, 가능한 취약점 공격 방식의 조합을 나타낼 수 있어 취약성을 분석하는데 유용한 도구로 제안되고 있다. 공격 그래프는 그래프 데이터 구조 기반의 공격 표현 모델로 공격 경로를 표현하기에는 유용하나 확장성 측면에서 상태 공간 폭발 (state-space explosion)의 문제를 가지고 있다. 취약점이 소프트웨어 또는 운영체제의 버전에 따라 변하기 때문에 공격 트리와 공격 그래프는 규모가 큰 네트워크에 적용하기 어려운 문제가 있다.

이러한 문제를 해결하기 위하여 다양하고 확장성 있는 공격 표현 모델에 관한 연구가 활발히 진행되고 있다. 베이지안 공격 그래프는 단순히 그래프의 형태로 공격 순서를 나타내는 것에서, 공격 대상 시스템 내 취약점을 악용하기 위한 확률값을 그래프에 추가한 모델이다. 이때, 확률값은 취약점의 상대적인 난이도를 나타낸다.

2.3 소프트웨어 정의망 기반의 Fault Tolerance 기술

소프트웨어 정의망 기술은 네트워크상의 데이터 트래픽 전달을 중앙 집중화된 방식으로 제어하는 네트워킹 기술이다. 소프트웨어 기반의 프로토콜을 사용하므로 전체 네트워크를 유연하고 신속하게 관리 및 제어할 수 있다. 이러한 이점으로 인하여 소프트웨어 정의망 기술을 활용한 fault tolerance 시스템이 제안되었다.

Yoon *et al.*은 네트워크 노드의 고장 및 오류를 대응하기 위한 방법으로 SDN 기반의 노드 스위칭 기술을 제안하였다[12]. 현재 사용 중인 노드와 IP 주소 설정은 같고 MAC 주소는 다른 백업 노드를 네트워크에 항상 위치시켜두어 시스템 장애가 발생한 경우, SDN 제어기를 활용하여 고장난 노드에서 백업 노드로 연결을 즉각적으로 변경시킨다. 장애가 발생한 노드와 백업 노드가 같은 IP 주소를 사용하기 때문에, 시스템 상의 다른 노드들이 네트워크 재설정 및 경로 재탐색의 시간지연 없이 서비스를 받을 수 있다.

Sahri *et al.*은 네트워크의 링크 단절 및 연결 오류에 대비한 소프트웨어 정의망 기반의 path switching 기법을 제안하였다[13]. 제안하는 기법은 사전에 백업

경로를 계산해 둔 다음, 링크에 오류가 발생하면 SDN 제어기를 사용하여 백업 경로로 빠르게 변경시켜준다. 소프트웨어 정의망을 활용하여 현재 링크와 토폴로지 상태에 기반한 최적의 복구 경로를 설정하므로, 시스템을 오류로부터 신속하게 회복시킬 수 있다.

Ⅲ. 시스템 모델

3.1 대상 네트워크 시스템

본 논문은 Figure 1과 같은 소프트웨어 정의망 기술을 기반으로 한 데이터센터 네트워크 시스템을 고려한다. 해당 네트워크 시스템에서 중앙의 SDN 제어기는 OpenFlow 프로토콜을 통해 SDN 스위치들에 플로우 테이블 업데이트 (flow table update) 및 패킷 포워딩 (packet forwarding)과 같은 네트워크 제어 작업을 수행하며, 노드들에 가상 주소를 할당하여 가상 주소와 실제 주소의 연결성을 제공할 수 있다. 네트워크 패킷의 전송을 담당하는 SDN 스위치 역시 OpenFlow를 프로토콜을 지원하며, 이를 통해 새로운 플로우들에 대한 처리를 SDN 제어기로 요청할 수 있다. SDN 제어기는 새로운 플로우에 대한 처리 명령을 스위치에 전달하게 되고, 스위치는 해당 명령을 스위치 내에 있는 플로우 테이블에 업데이트하여 포워딩을 처리한다. 서버와 호스트의 네트워크 포트 (port)는 현재 제공되는 서비스에 한하여 개방되어있으며, 사이버 공격자는 실행 중인 서비스의 취약점을 악용하여 특정 노드로 침입할 수 있다.

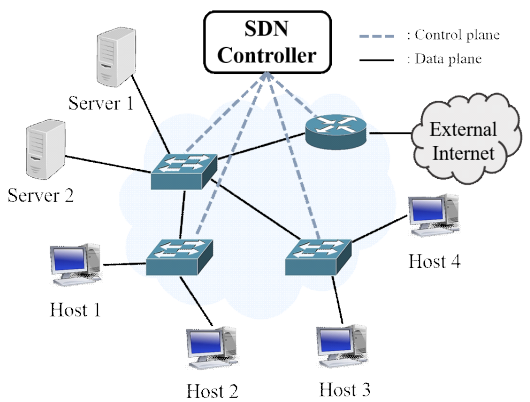


그림 1. 대상 네트워크 시스템
Fig. 1. Target network system

3.2 베이지안 공격 그래프

베이지안 공격 그래프는 네트워크 구조와 취약점 특성을 분석하여 취약점 exploit 공격 경로를 추정할 수 있는 보안 분석 방법이다. 일반적으로, 공격 그래프상의 노드는 취약점을 나타내며, 엣지 (Edge)는 취약점 간 연관성을 나타낸다. Edge에 부여된 가중치 (p_i)는 Edge가 가리키는 방향의 취약점을 공격하여 성공할 확률을 나타낸다.

취약점을 공격하여 성공할 확률은 공격 그래프 생성 시 정한 규칙에 따라 다양하게 정의될 수 있으나, 기본적으로 그 취약점의 CVSS 점수를 기반으로 정의된다. 예를 들어, Poolsappasit *et al.*는 이 확률값을 CVSS의 평가 기준에 있는 Exploitability 벡터값을 모두 곱하여 정의하였다[14]. Zhang *et al.*는 CVSS의 Exploitability 점수를 이용하여, 공격 그래프 Edge의 가중치(p_i)를 p_i = 1 - exp(-(Exploitability Score of X_i))로 정의하였다[2]. 이와 유사하게, 본 논문은 공격 그래프의 Edge 가중치(p_i)를 각 취약점의 Exploitability 점수를 CVSS의 최댓값인 10으로 정규화하여 정의하였으며, 이는 해당 취약점의 공격성공확률을 의미한다. Table 1은 Host 1의 개방된 포트와 취약점의 CVE 명칭, 이와 관련된 CVSS 점수, 그리고 공격 그래프의 Edge 가중치를 나타낸다.

Figure 2는 Figure 1의 호스트들이 가지고 있는 취약점 간의 연결성을 나타낸 베이지안 공격 그래프이다. Figure 2에 보여지는 그래프상의 노드 X_i는 Figure 1의 호스트들이 가지고 있는 취약점 중 하나를 의미한다. 본 논문은 취약점 X_i를 베르누이 확률 변수 (Bernoulli random variable)로 모델링하였으며, X_i = 1은 취약점이 공격자에 의해 성공적으로 exploit된 상태를, X_i = 0은 exploit되지 않은 상태를 나타낸다. 취약점 X_i로 향하는 Edge에 부여된 공격성공확률 p_i는 X_i의 부모 노드 중 하나에서 X_i를 공격하여 성공할 확률을 의미한다. 즉, X_i의 부모 노드 중 임의의 한 노드가 X_j인 경우, p_i는 p_i = Pr(X_i = 1|X_j = 1)로 표현할

표 1. Host 1의 취약점 및 그래프 Edge 가중치
Table 1. Vulnerabilities of host 1 and edge weight in a BAG

Port	CVE	CVSS	Edge weight in a BAG (p_i)
53	CVE-2012-1667	8.5	0.85
80	CVE-2017-7679	7.5	0.75
2121	CVE-2009-3639	5.8	0.58
2121	CVE-2011-4130	9.0	0.90

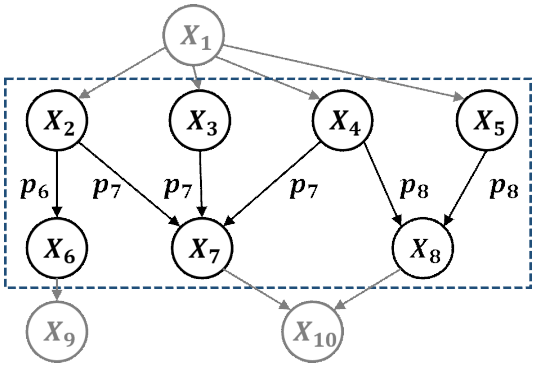


그림 2. 베이지안 공격 그래프의 예시
Fig. 2. A BAG of target network system

수 있다.

그래프의 구조와 모든 노드에 대한 공격성공확률 p_i이 주어지면, 공격자의 최초 침입으로부터 각 노드에 도달할 확률 $\Pr(X_i)$을 구할 수 있다. 도달성공확률 $\Pr(X_i)$는 베이즈 규칙 (Bayes rule)을 사용하여 아래와 같이 계산될 수 있으며, 가장 공격에 노출된 취약점을 파악하기 위해 사용된다. 아래 식에서, 베이지안 공격 그래프상의 모든 노드 집합은 $\boldsymbol{X}$로, 노드 X_i의 부모 노드 집합은 $\boldsymbol{pa}_i$로 표기하였다.

$$\Pr(X_i) = \sum_{\boldsymbol{X}-X_i} \Pr(\boldsymbol{X}) = \sum_{\boldsymbol{X}-X_i} \prod_{j=1}^{n} \Pr(X_j \mid \boldsymbol{pa}_j) \quad (1)$$

본 논문은 취약점 X_i를 성공적으로 공격하기 위해서는 $\boldsymbol{pa}_i$에 포함된 노드 중 하나라도 exploit된 상태이어야 한다고 가정한다. 즉, 식 (1)의 조건부확률 $\Pr(X_j|\boldsymbol{pa}_j)$는 다음과 같은 식으로 표현할 수 있다. $\boldsymbol{pa}_j$에 포함된 임의의 노드 X_k에 대하여, 모든 X_k의 상태가 0인 경우는 $\Pr(X_j=1|\boldsymbol{pa}_j)=0$이고, 하나라도 X_k의 상태가 1이면, $\Pr(X_j=1|\boldsymbol{pa}_j)=1-\Pi_{k:Xk}(1-p_k)$이다.

한편, 침해 발생 시 베이지안 공격 그래프를 활용하면 침입자가 어떤 공격 경로를 통해 해당 침해에 이르렀는지를 추정할 수 있다. 이러한 추정은 베이지안 그래프에 추론 알고리즘 (inference algorithm)을 적용하여 구할 수 있으며, Variable Elimination와 Junction Tree 등과 같은 그래프 알고리즘 등이 사용된다. 식 (2)는 공격 그래프상의 특정 노드의 침해를 감지하였을 때, 이 노드에서 발생한 사건을 증거 노드 (evidence) $\boldsymbol{E}$로 두고 추론하여 공격원인확률 $\Pr(X_i|\boldsymbol{E})$를 구하는 과정을 나타낸다. 공격원인확률 $\Pr(X_i|\boldsymbol{E})$를 모든 취약점에 대해서 계산하게 되면, 침해 사건 $\boldsymbol{E}$에 대한 원인이 될 가능성이 높은 취약점을 파악할 수 있다.

$$\Pr(X_i \mid \boldsymbol{E}) = \frac{\Pr(\boldsymbol{E} \mid X_i)\Pr(X_i)}{\Pr(\boldsymbol{E})} \quad (2)$$

Ⅳ. 소프트웨어 정의망을 활용한 침입감내 기술

본 논문이 제안하는 침입감내 기술은 네트워크 자원 침해가 발생하여 이를 인지한 경우, 해당 침해로부터 추가 발생할 수 있는 피해를 즉각적으로 제한하고, 기존의 서비스 가용성을 최대한으로 보장하는 것을 목표로 하고 있다. 즉, 네트워크 자원 침해 발생 시, 베이지안 공격 그래프로 공격 경로를 추정한 다음, 소프트웨어 정의망 기술을 활용하여 네트워크 자원을 즉각적으로 고립시키는 동시에 경로 내 자원들의 네트워크 설정을 변경시킨다. 또한, 자원 가상화 기술을 통하여 손상된 서비스를 정상 서비스로 복구하여, 침해 상황에도 기존 서비스의 성능 저하를 최소화한다. 이러한 일련의 절차는 사이버 공격의 확산을 막아 공격에 대한 높은 복원력과 서비스의 지속성을 유지할 수 있게 한다.

Figure 3은 제안하는 공격 그래프 기반의 침입감내 기술의 적용 예를 보여준다. Figure 3은 총 16개의 취약점으로 이뤄진 공격 그래프이다. X_1과 X_2는 외부망에서 접근이 가능한 취약점으로서 침입자는 초기에 X_1과 X_2를 통하여 침입을 시도하게 된다. 일반적으로, X_{15}와 X_{16}는 침입자가 최종 목표로 하는 네트워크 자원의 취약점이다. 침입자는 X_1과 X_2를 통하여 네트워크 망에 침입하여 X_{15}와 X_{16}에 도달할 때까지 네트워크의 토폴로지 및 주소 등의 정보를 사용한다.

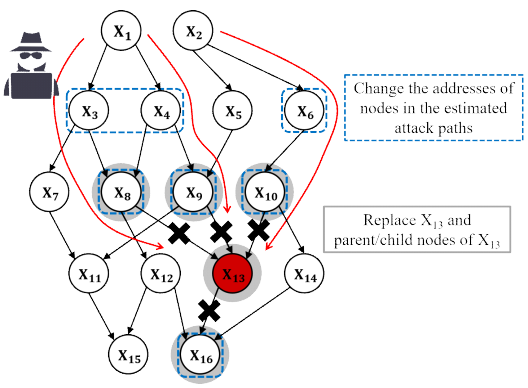


그림 3. 공격 그래프 기반 침입 감내 기술
Fig. 3. BAG based attack tolerance method

4.1 공격 그래프 기반의 취약점 분석

소프트웨어 정의망을 활용한 침입감내 기술을 위해, 가장 먼저 대상 네트워크를 취약점 기반의 공격 그래프로 표현한다. 공격 그래프는 침해 자원이 탐지된 경우, III장에서 언급한 추론 알고리즘을 적용하여 공격 경로를 추정하기 위해 사용된다. 추정된 공격 경로는 침해 노드로 도달하기 위해 이전에 공격되었을 노드들을 의미하며, 이를 활용하면 보다 정확하고 신속하게 공격에 대응할 수 있다. 침입감내를 위한 구체적인 방법은 다음 절에서 제시하는 침해 노드 고립화, 네트워크 설정 변경, 그리고 서비스 복구가 있으며, 이와 같은 절차를 통해 공격자의 영향력은 최소화하고 서비스 가용성은 유지하게 한다.

4.2 침해 노드의 고립화

공격 그래프 상에서 침해 노드와 인접한 노드들은 공격 우회 대상이 될 수 있다. 따라서, 공격의 확산을 막기 위해, 침해된 자원과 연결된 네트워크 링크는 모두 차단시켜야 한다. 이와 같은 이유로, 본 논문은 소프트웨어 정의망 기술을 활용하여 침해된 노드와 인접한 노드들의 네트워크 연결을 일시적으로 끊어주어 침해 노드를 고립화하는 것을 제안한다. Figure 3은 침해 노드 X_{13}와 주변 노드 X_8, X_9, X_{10}, X_{16} 사이의 네트워크 연결이 차단되어, X_{13}가 고립화된 것을 보여준다.

4.3 침해발생 공격경로상 노드의 네트워크 설정 변경

네트워크에 침입한 공격자는 침해 노드를 고립화하여도 네트워크의 정적인 특성을 사용하여 새로운 공격을 시도할 수 있다. 따라서, 공격 경로상 노드들의 네트워크 정보를 주기적으로 변경시켜 공격자가 설계한 공격 경로를 무효화시키는 것이 필요하다. 이때, 모든 노드의 네트워크 설정을 변경하면, 시스템 오버헤드가 커지므로, 공격 그래프를 활용하여 공격자의 영향력이 미칠 가능성이 높은 노드들만을 선별적으로 변경시킨다. 이를 위해, 공격 그래프에서 침해 노드를 시작으로 추론 알고리즘을 적용하여, 부모 노드의 공격원인확률 $\Pr(X_i|\boldsymbol{E})$을 구하고, 부모 노드 개수의 p% 비율로 공격원인확률이 큰 부모 노드들의 네트워크 설정을 변경한다. 이와 같은 과정은 공격 그래프의 초기 취약점에 도달할 때까지 역방향으로 반복하여 진행한다. Figure 3은 X_{13}을 증거 노드로 두어 공격원인확률이 높은 X_8, X_9, X_{10}을 찾고, X_8가 공격된 상태로 증거 노드에 두고 추론하여 공격원인확률이 가장 높은 부모 노드 X_3, X_4를 찾은 것을 나타내었다. 추론을 반복하여, 공격 경로를 추정하고, 공격 경로 내 노드의 네트워크 정보를 변경하면 공격자가 획득한 정보를 효율적으로 무효화할 수 있다.

4.4 침해발생 인접 노드의 서비스 복구

침해 노드를 일시적으로 고립화하고 공격 경로상의 노드들의 네트워크 설정을 변경한 다음, 손상된 서비스를 정상 서비스로 대체하여 복구하는 과정이 필요하다. 이를 위하여, 네트워크 자원의 모든 서비스를 가상화하여 제공하고, 침해 상황에 대비하여 네트워크 시스템에 구동 중인 서비스와 동일한 정상 상태의 백업 서비스를 미리 준비해둔다. 침해가 탐지되면, 공격 그래프에서 침해 발생 노드와 해당 노드의 부모 노드, 자식 노드를 파악하고, 이와 같은 인접노드의 백업 서비스들을 해당 노드들로 migration하여, 침해 이전상태로 복구한다. 침해 인접 노드에 서비스 복구화를 우선적으로 적용하는 이유는 침해 노드에 도달하기 위해 부모 노드 중 하나가 성공적으로 공격 되어야 하며, 공격자가 이미 자식 노드를 공격하고 있을 수 있기 때문이다. Figure 3은 침해가 탐지된 노드 X_{13}와 이 노드의 부모 노드, 자식 노드를 정상 서비스로 대체하여 복구한 것을 보여준다.

Ⅴ. 실험 결과

본 논문은 기존 침입감내 기술과 다르게 공격 경로를 예측하여 우회 공격을 사전에 막음으로써 추가적인 공격에 대한 가능성을 최소화시킨다. 침입 추정 성능을 평가하기 위해, MATLAB toolbox인 bnet로 베이지안 공격 그래프와 공격 경로를 생성한 다음, 제안하는 알고리즘으로 찾은 추정 경로와 비교 분석하였다[15]. 침해 대응을 위한 서비스 단절 시간은 Mininet emulator로 구축한 가상 네트워크 환경에서 측정하였다[16]. 예측된 공격 경로상 노드의 서비스 단절 시간을 측정하여 서비스 지속성에 대해 평가하였다.

5.1 침입 추정 성능 평가

본 논문이 제안하는 침입감내 기술의 성능 평가를 위해, MATLAB 환경에서 bnet을 활용하여 베이지안 공격 그래프를 생성하였다. MATLAB toolbox인 bnet은 베이지안 그래프를 생성하고, 그래프 노드의 다양한 확률 분포를 설정할 수 있으며, 그래프 분석에 필요한 추론 알고리즘, 파라미터 학습, 정규화 등을 제공한다. 이를 활용하여, 침입 추정 성능을 평가하기 위해 다음과 같은 베이지안 그래프를 생성하였다.

공격 그래프의 구조는 Layer마다 노드가 있고 노드에서 나가는 Edge는 바로 아래 Layer의 노드들로만 향하게 설정하였다. 여기에서, 노드 간의 연결과 각 노드들의 취약성은 CVE 데이터베이스에서 Windows 10을 키워드로 검색한 결과를 참조하였으며, 본 논문에서 사용한 Windows 10 관련 취약점들의 CVSS 점수는 Figure 4에 히스토그램으로 표현하였다. Edge 가중치(p_i)는 Edge가 향하는 노드의 CVSS 값을 정규화하여 할당하였다. 침입 추정의 정확도를 검증하기 위해, 그래프의 depth, 각 노드의 out degree, 그리고 각 Layer에 포함된 노드 수를 변경하여 다양한 토폴로지의 공격 그래프에서 시뮬레이션을 수행하였다.

제안하는 침입감내 기술의 성능 검증은 침입감내 기술을 적용하였을 때의 오버헤드와 침입 추론 성공률을 측정함으로써 보였다. 침입감내 오버헤드는 제안하는 방법과 다음의 두 가지 방법을 비교하여 평가하였다. 첫 번째 방법은 침입감내를 위하여 모든 노드의 네트워크 설정을 변경하고, 두 번째 방법은 침해 노드를 추정하여 역방향으로 가능한 모든 공격 경로상의 노드의 네트워크 설정을 변경한다.

침입 경로상의 침입 추론 성공률은 제안하는 침입감내 기술을 서로 다른 토폴로지를 갖는 베이지안 공격 그래프에 적용함으로써 그 결과를 보였다. 본 논문에서 생성한 2가지 토폴로지는 구체적으로 다음과 같다. 토폴로지 1은 depth가 5, 첫 번째 Layer의 노드 개수가 1개, 나머지 Layer의 노드 개수는 각각 10개, 각 노드의 out degree는 3인 전반적으로 가로로 긴 형태이다. 토폴로지 2는 depth가 10이고, 첫 번째 Layer를 제외한 모든 Layer들의 노드 개수가 각각 5개, 각 노드의 out degree가 3인 세로로 긴 형태의 구조이다.

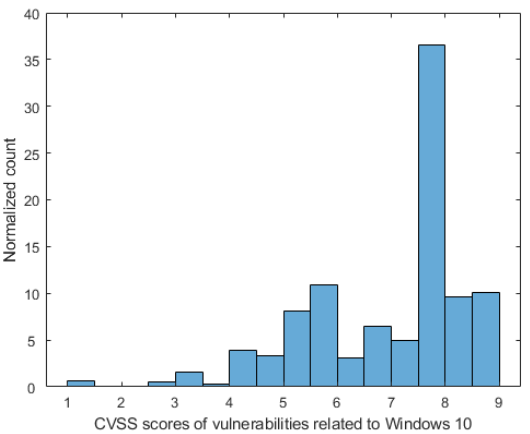


그림 4. Windows 10 취약점의 히스토그램
Fig. 4. Histogram of Windows 10 vulnerabilities

제안하는 기술은 침해가 발생하였을 것으로 예상되는 노드를 추론하고 네트워크 설정을 변경함으로써 침입자가 획득한 정보를 무효화 하는 방법이다. 따라서 침입 추론 성공률이 높을수록 침입에 의한 영향을 최소화할 수 있어서 침입감내 효과가 높다고 할 수 있다.

실험 결과는 다음과 같은 metric를 사용하여 정량적으로 표현하였다. 네트워크상의 모든 노드 집합은 $\boldsymbol{X}=\{X_1, ..., X_n\}$로 정의한 후, 총 노드 개수를 $N = |\boldsymbol{X}|$로 표기하였다. 또한, 침해가 예측되는 노드들을 추론하여 역방향으로 찾은 모든 공격 경로상의 노드들은 P_{attack} 으로, p%를 정한 뒤 제안하는 침입감내 알고리즘으로 찾은 노드들은 $P_{Reconfiguration,\ p}$로 정의하였다. 침입감내 기술의 오버헤드는 네트워크 설정이 변경되는 노드 개수와 네트워크상의 모든 노드의 개수를 비교하여 도출한다. 본 논문에서 사용한 두 가지 방법은 $|P_{attack}|/N$와 $|P_{Reconfiguration,\ p}|/N$로 표현할 수 있다. 침입 추론 성공률은 제안 알고리즘에 의해 찾은 노드 $P_{Reconfiguration,\ p}$와 공격 경로상의 노드 P_{attack}의 일치 정도를 비교하여 계산하였다.

Figure 5와 Figure 6은 각각 토폴로지 1과 토폴로지 2의 공격 그래프에서, p% 만큼의 부모 노드에 침입감내 기술을 적용할 때의 오버헤드를 보여준다. 본 실험 결과를 통해, 제안 알고리즘은 p를 증가함에 따라 오버헤드가 증가하나, 모든 노드와 역방향으로 찾은 공격 경로에 주소 변경을 수행하는 경우보다는 오버헤드가 항상 적음을 알 수 있다. 또한, 모든 p에 대해 Figure 5의 결과가 Figure 6보다 작은 것을 통해, 옆으로 긴 형태의 토폴로지일수록 침입감내 오버헤드가 더 작음을 알 수 있다.

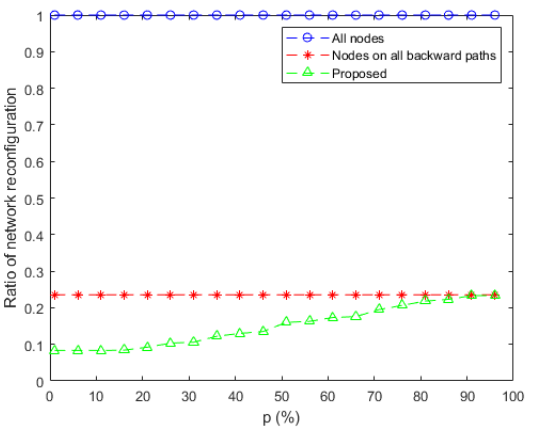


그림 5. Topology1의 침입감내 기술 오버헤드 측정 결과
Fig. 5. Overhead of intrusion tolerance method on Topology 1 (depth=5, nodes of each layers=10, out-degree=3)

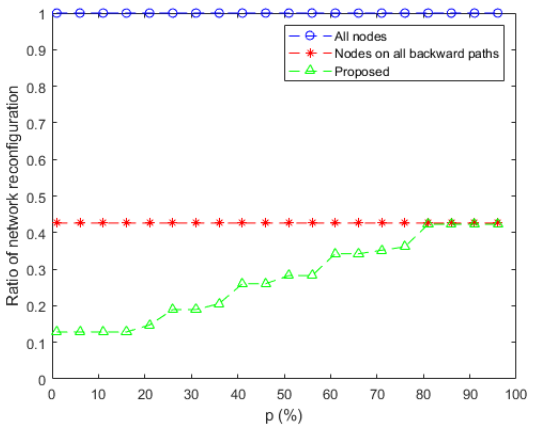


그림 6. Topology2의 침입감내 기술 오버헤드 측정 결과
Fig. 6. Overhead of intrusion tolerance method on Topology 2 (depth=10, nodes of each layers=5, out-degree=3)

Figure 7은 토폴로지 1과 토폴로지 2에 침입감내 기술을 적용하여 측정한 침입 추론 성공률을 보여준다. 침입 추론 성공률은 침해가 탐지된 노드의 Layer가 정해졌을 때, 그 Layer로 도달하기 위해 가능한 모든 공격 경로와 침입감내 기술을 적용하여 찾은 공격 경로의 일치 정도를 비교하여 구한다. Figure 7에서, x축은 침입감내 알고리즘을 적용할 때 설정하는 p값이며, y축은 공격 경로의 일치 정도이다. 본 결과를 통해, 두 개의 토폴로지 모두 p가 증가됨에 따라 침입 추론 성공률이 증가하였으며, 옆으로 긴 모양인 토폴로지 1의 공격 그래프의 경우가 위아래로 긴 모양의 토폴로지 2보다 전반적으로 침입 추론 성공률이 높음을 알 수 있다.

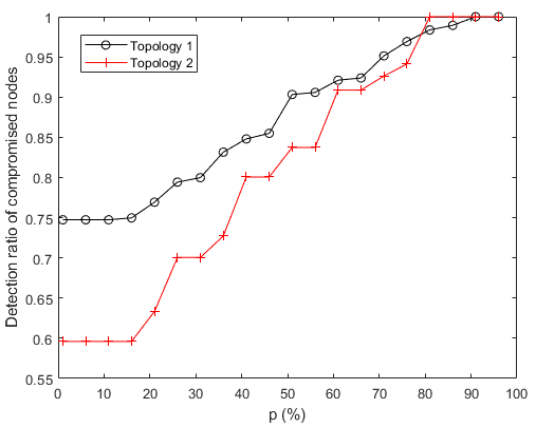


그림 7. 제안하는 침입 감내 기술의 침입 탐지율 측정 결과
Fig. 7. Intrusion inference success rate of intrusion tolerance method

5.2 침해 대응 서비스 단절 시간 평가

제안하는 침입감내 기술의 침해 대응 시간을 평가하기 위해, Mininet emulator를 활용하여 가상 네트워크 환경을 구축하였다. 구체적으로, SDN 스위치는 OVS (Open vSwitch)로 설정하고, 호스트는 Mininet emulator로 생성한 컨테이너를 Docker 기반 컨테이너로 변환하여 사용하였다. 여기에서, Docker 기반 컨테이너로 변환시킨 이유는 Mininet emulator의 호스트가 운영체제 및 서비스를 포함하고 있지 않아 취약점 분석이 어렵기 때문이다. 본 논문은 Mininet 호스트를 Docker 컨테이너로 변환한 가상 네트워크에서 다양한 실험을 수행하기 위해 개발된 Containernet 기술을 사용하였다[17]. 자동화된 침해 대응 프로그램을 개발하기 위하여, Mininet에서 제공하는 API (Application Programming Interface)를 이용하였으며, 생성한 가상 네트워크를 SDN 제어기 ONOS (Open Network Operating System)와 연결하여 망 정보를 제공하였다. 또한, Docker 컨테이너의 취약점을 분석하기 위해 컨테이너에 설치된 운영체제의 취약점, 응용 프로그램의 의존도 등을 탐지하는 취약점 스캐너인 Trivy를 사용하였다[18].

본 실험에서 측정한 침해 대응 서비스 단절 시간은 가상 네트워크에서 취약점을 분석하는 시간과 베이지안 공격 그래프를 활용하여 침해 대응이 필요한 노드를 추정하는 시간, 그리고 추정 결과에 해당하는 노드에 새로운 컨테이너를 배치하기까지의 소요 시간의 합으로 정의하였다. 즉, 이는 침해 발생한 시점부터 공격 피해가 예상되는 노드를 추정하고 이를 복구하여, 일시적으로 서비스가 단절되는 시간을 의미한

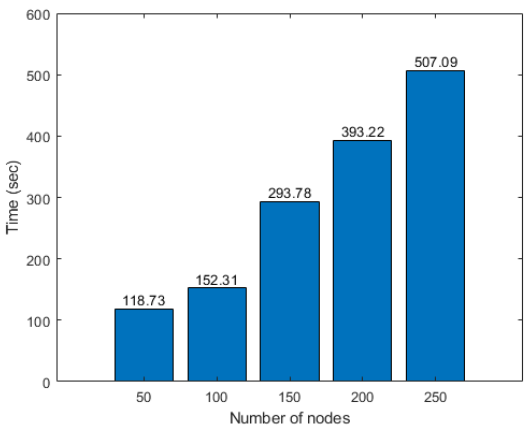


그림 8. 침해 대응 서비스 단절 시간 측정 결과
Fig. 8. Service downtime according to the number of nodes

다. Figure 8은 50개부터 250개까지의 노드로 구성된 가상 네트워크에서 측정한 침해 대응 소요 시간을 보여준다. 네트워크의 토폴로지는 스위치 하나 당 컨테이너 한 개를 연결하고 스위치 간의 연결은 앞 절에서 생성한 공격 그래프와 동일한 규칙으로 설정하였다. 결과 그래프를 통해 네트워크의 크기가 커짐에 따라, 침해 대응에 많은 시간이 필요함을 알 수 있다. 제안하는 침입감내 기술은 모든 노드가 아닌 침해 피해가 예상되는 노드만을 복구화하기 때문에, 침해 대응을 위한 서비스 단절 시간을 최소화할 수 있다.

Ⅵ. 결 론

침입감내 기술은 사이버 공격으로 인해 네트워크 자원 침해가 발생하여도 서비스를 지속적으로 제공하기 위해 개발되었다. 본 논문이 제안하는 침입감내 기술은 베이지안 공격 그래프에 추론 알고리즘을 적용하여 추후 악용될 수 있는 노드를 추정하고, 해당 노드들의 네트워크 주소만을 재설정하여 공격의 확산을 막는다. 또한, 선별적으로 서비스를 복구하여 침해 대응을 위한 서비스 단절시간을 최소화하였다. 실험 결과를 통해 공격 경로 내 노드들을 성공적으로 예측하여 서비스 단절시간을 최소화할 수 있음을 검증하였다.

References

[1] S. Yoon, J. H. Cho, D. S. Kim, T. J. Moore, F. Free-Nelson, and H. Lim, “Attack graph-based moving target defense in software-defined networks,” *IEEE Trans. Netw. and Serv. Manag.*, vol. 17, no. 3, pp. 1653-1668, Sep. 2020.

[2] S. Zhang and S. Song, “A novel attack graph posterior inference model based on Bayesian network,” *J. Inf. Secur.*, vol. 2, no. 1, pp. 8-27, Jan. 2011.

[3] MITRE Corporation, *A list of records for publicly known cybersecurity vulnerabilities*, Retrieved Jun. 1, 2020, from https://cve.mitre.org.

[4] Forum of Incident Response and Security Teams (FIRST), *A way to capture the principal characteristics of a vulnerability and produce a numerical score reflecting its severity*, Retrieved Jun. 7, 2020, from https://www.first.org/cvss.

[5] Open Networking Foundation, *Definition of software-defined networking*, Retrieved Jan. 17, 2021, from https://opennetworking.org/sdn-definition.

[6] P. Kampanakis, H. Perros, and T. Beyene, “SDN-based solutions for moving target defense network protection,” in *Proc. IEEE WoWMoM*, pp. 1-6, Sydney, NSW, Australia, Jun. 2014.

[7] J. Reynolds, J. Just, E. Lawson, L. Clough, R. Maglich, and K. Levitt, “The design and implementation of an intrusion tolerant system,” in *Proc. Int. Conf. Dependable Syst. and Netw.* (DSN), pp. 285-290, Washington, D.C., USA, Jun. 2002.

[8] P. Verissimo, A. Casimiro, and C. Fetzer, “The timely computing base: Timely actions in the presence of uncertain timeliness,” in *Proc. Int. Conf. Dependable Syst. and Netw.* (DSN), pp. 533-542, New York, USA, Jun. 2000.

[9] S. Heo, S. Lee, B. Jang, and H. Yoon, “Designing and implementing a diversity policy for intrusion-tolerant systems,” *IEICE Trans. Inf. and Syst.*, vol. 100, no. 1, pp. 118-129, Jan. 2017.

[10] A. Gorbenko, A. Romanovsky, O. Tarasyuk, and O. Biloborodov, “From analyzing operating system vulnerabilities to designing multiversion intrusion-tolerant architectures,” *IEEE Trans. Reliability*, vol. 69, no. 1, pp. 22-39, Mar. 2019.

[11] Y. Yoon and S. Heo, “Intrusion tolerant system for service availability,” *J. KIIT*, vol. 16, no. 5, pp. 83-90, May 2018.

[12] S. Yoon, J. Lee, Y. Kim, S. Kim, and H. Lim, “Fast controller switching for fault-tolerant cyber physical systems on software-defined networks,” *IEEE Pacific Rim Int. Symp. Dependable Comput.* (PRDC), pp. 211-212, Christchurch, New Zealand, Jan. 2017.

[13] N. M. Sahri and K. Okamura, “Fast failover mechanism for software defined networking:

Openflow based," in *Proc. Ninth Int. Conf. Future Internet Technologies* (CFI), pp. 1-2, Tokyo, Japan, Jun. 2014.

[14] N. Poolsappasit, R. Dewri, and I. Ray, "Dynamic security risk management using Bayesian attack graphs," *IEEE Trans. Dependable and Secure Comput.*, vol. 9, no. 1, pp. 61-74, Jun. 2011.

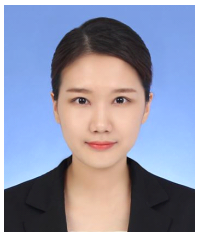
[15] K. Murphy, *Bayesnet toolbox for Matlab* (2014), Retrieved Jan. 2, 2021, from https://github.com/bayesnet/bnt.

[16] Mininet Team, *An instant virtual network on PC*, Retrieved Jan. 5, 2021, from http://mininet.org.

[17] M. Peuster, *Containernet: Mininet fork that allows to use Docker containers as hosts in emulated networks* (2016), Retrieved Jun. 10, 2020, from https://github.com/containernet/containernet.

[18] T. Fukuda, *A simple and comprehensive vulnerability scanner for containers, suitable for CI*, Retrieved Jun. 19, 2020, from https://github.com/aquasecurity/trivy.

김 혜 진 (Hyejin Kim)



2017년 2월 : 동아대학교 전자공학과 학사

2017년 3월~현재 : 광주과학기술원 인공지능대학원 석박사 통합과정

<관심분야> 소프트웨어 정의 네트워킹, 사이버 보안 모니터링, 침입감내기술, 인공지능

[ORCID:0000-0001-8448-0168]

윤 승 현 (Seunghyun Yoon)

2016년 2월 : 한동대학교 전산전자공학부 학사

2019년 8월~2020년 2월 : 미국 Virginia Tech 방문연구원

2021년 2월 : 광주과학기술원 전기전자컴퓨터공학부 박사

<관심분야> 소프트웨어 정의 네트워킹, 사이버 보안, 네트워크 모니터링, moving target defense, 강화학습

[ORCID:0000-0001-6264-976X]

김 성 환 (Sunghwan Kim)

2015년 2월 : 동국대학교 컴퓨터공학과 학사

2017년 2월 : 광주과학기술원 전기전자컴퓨터공학부 석사

2017년 3월~현재 : 광주과학 기술원 전기전자컴퓨터공학부 박사과정

<관심분야> 소프트웨어 정의 네트워킹, 사이버 보안, 네트워크 모니터링, 인공지능

[ORCID:0000-0001-8708-1699]

임 혁 (Hyuk Lim)

1996년 2월 : 서울대학교 전기공학부 학사

1998년 2월 : 서울대학교 전기공학부 석사

2003년 8월 : 서울대학교 전기컴퓨터공학부 박사

2006년~현재 : 광주과학기술원 인공지능대학원 교수

<관심분야> 컴퓨터 네트워크, 사이버 보안, 인공지능

[ORCID:0000-0002-9926-3913]