KIPS Trans. Comp. and Comm. Sys.
Vol.11, No.1 pp.23~34
ISSN: 2287-5891 (Print), ISSN 2734-049X (Online)

https://doi.org/10.3745/KTCCS.2022.11.1.23

Blocking Intelligent Dos Attack with SDN

Junhyeok Yun† · Sungsik Mun† · Mihui Kim††

ABSTRACT

With the development of network technology, the application area has also been diversified, and protocols for various purposes have been developed and the amount of traffic has exploded. Therefore, it is difficult for the network administrator to meet the stability and security standards of the network with the existing traditional switching and routing methods. Software Defined Networking (SDN) is a new networking paradigm proposed to solve this problem. SDN enables efficient network management by programming network operations. This has the advantage that network administrators can flexibly respond to various types of attacks. In this paper, we design a threat level management module, an attack detection module, a packet statistics module, and a flow rule generator that collects attack information through the controller and switch, which are components of SDN, and detects attacks based on these attributes of SDN. It proposes a method to block denial of service attacks (DoS) of advanced attackers by programming and applying honeypot. In the proposed system, the attack packet can be quickly delivered to the honeypot according to the modifiable flow rule, and the honeypot that received the attack packets analyzed the intelligent attack pattern based on this. According to the analysis results, the attack detection module and the threat level management module are adjusted to respond to intelligent attacks. The performance and feasibility of the proposed system was shown by actually implementing the proposed system, performing intelligent attacks with various attack patterns and attack levels, and checking the attack detection rate compared to the existing system.

Keywords : Software Defined Networking, Denial of Service Attack, Honeypot, Intelligent Attack, Adaptive System

SDN과 허니팟 기반 동적 파라미터 조절을 통한 지능적 서비스 거부 공격 차단

윤 준 혁† · 문 성 식† · 김 미 희††

요 약

네트워크 기술의 발달로 그 적용 영역 또한 다양해지면서 다양한 목적의 프로토콜이 개발되고 트래픽의 양이 폭발적으로 증가하게 되었다. 따라서 기존의 전통적인 스위칭, 라우팅 방식으로는 네트워크 관리자가 망의 안정성과 보안 기준을 충족하기 어렵다. 소프트웨어 정의 네트워킹(SDN)은 이러한 문제를 해결하기 위해 제시된 새로운 네트워킹 패러다임이다. SDN은 네트워크 동작을 프로그래밍하여 효율적으로 네트워크를 관리할 수 있도록 한다. 이는 네트워크 관리자가 다양한 여러 양상의 공격에 대해서 유연한 대응을 할 수 있는 장점을 가진다. 본 논문에서는 SDN의 이러한 특성을 활용하여 SDN 구성 요소인 컨트롤러와 스위치를 통해 공격 정보를 수집하고 이를 기반으로 공격을 탐지하는 위협 레벨 관리 모듈, 공격 탐지 모듈, 패킷 통계 모듈, 플로우 규칙 생성기를 설계하여 프로그래밍하고 허니팟을 적용하여 지능형 공격자의 서비스 거부 공격(DoS)을 차단하는 방법을 제시한다. 제안 시스템에서 공격 패킷은 수정 가능한 플로우 규칙에 의해 허니팟으로 빠르게 전달될 수 있도록 하였으며, 공격 패킷을 전달받은 허니팟은 이를 기반으로 지능적 공격의 패턴을 분석하도록 하였다. 분석 결과에 따라 지능적 공격에 대응할 수 있도록 공격 탐지 모듈과 위협 레벨 관리 모듈을 조정한다. 제안 시스템을 실제로 구현하고 공격 패턴 및 공격 수준을 다양화한 지능적 공격을 수행하고 기존 시스템과 비교하여 공격 탐지율을 확인함으로써 제안 시스템의 성능과 실현 가능성을 보였다.

키워드 : 소프트웨어 정의 네트워킹, 서비스 거부 공격, 허니팟, 지능적 공격, 적응적 시스템

1. 서 론

정보 통신 기술의 발달로 클라우드 기반 네트워크가 일반화되고 서버 가상화, 네트워크 기반 응용 프로그램 다양화 등 변화가 나타나고 있다. 이로 인해 네트워크 트래픽의 양은 폭발적으로 증가했고, 네트워크 애플리케이션 영역에 직접 영향을 미쳐 새로운 목적을 효율적으로 수행하기 위한 다양한 프로토콜이 개발되고 있다. 이렇듯 급변하는 네트워크 환경에서 네트워크 관리자가 기존의 전통적인 스위칭, 라우팅 방식으로 네트워크를 구성하고 관리하는 경우 일정한 안전성과 보안 기준을 충족시키기 어렵다. 소프트웨어 정의 네트워킹(Software Defined Networking, SDN)은 이러한 문제를 해결하기 위해 제시된 새로운 네트워킹 패러다임이다. SDN은 실제 트래픽 전달을 수행하는 데이터 플레인과 라우팅 규

※ 이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No.2018R1A2B6009620).
† 준 회 원 : 한경대학교 컴퓨터응용수학부 석사
†† 종신회원 : 한경대학교 컴퓨터응용수학부 교수
Manuscript Received : May 24, 2021
Accepted : August 19, 2021
* Corresponding Author : Mihui Kim(mhkim@hknu.ac.kr)

칙, 보안 규칙 등을 포함하는 컨트롤 플레인을 분리함으로써 소프트웨어 기반의 유연한 망 관리를 가능하도록 하는 기술이다. 데이터 플레인에는 SDN 스위치, 컨트롤 플레인에는 SDN 컨트롤러가 위치하며, SDN 스위치는 전통적인 스위치와 달리 주소 기반의 포워딩뿐만 아니라 SDN 컨트롤러가 지정한 플로우 규칙 기반 포워딩을 수행할 수 있다. 플로우 규칙을 유연하게 생성, 수정할 수 있는 애플리케이션을 개발하여 적용함으로써 네트워크 관리자가 네트워크 내에서 데이터 또는 리소스를 처리하는 방식을 신속하게 정의, 재구성할 수 있다[1].

본 논문에서는 SDN의 유연한 트래픽 관리 기능을 기반으로 서비스 거부(Denial of service, DoS) 공격을 탐지하고 차단하는 기법을 제시한다. DoS 공격은 공격자가 다양한 방법으로 장치의 정상적인 작동을 방해하여 정상 사용자가 컴퓨터 또는 기타 장치를 사용할 수 없도록 만드는 보안 공격이다. DoS 공격은 대상 시스템이 트래픽을 처리할 수 없을 때까지 대상 시스템에 대량의 요청을 반복하여 사용자가 시스템을 사용할 수 없도록 한다. 이러한 공격은 여러 기관이나 기업에 막대한 피해를 가져올 수 있어 적절한 대응법이 필요하다. 특히 공격자가 DoS 공격을 막기 위해 구성한 방어 시스템을 파악하고, 방어 시스템의 약점을 이용해 공격하는 등 지능적 공격 수법은 날이 갈수록 발전하고 있다. 그러므로 이에 대한 유연한 대응법이 필요하다[2].

SDN 환경에서 DoS 공격을 차단하는 여러 연구들이 논의되어 왔다[3]. 기존 방어 기법으로는 트래픽의 급격한 증가를 탐지하여 트래픽의 폭증을 막는 기법이나 패킷 정보를 모니터링하여 이상 징후가 발견되면 패킷을 차단하는 등의 기법들이 제시되었다. 하지만 이러한 연구들은 비교적 높은 오탐지율과 오버헤드 문제, 고정 값 타이머의 한계점 등 해결해야 하는 문제점을 포함하고 있다. 또한 기존 연구에서는 지능형 공격에 대한 방어 기법을 제시하지 못하였다.

본 논문에서는 소프트웨어 기반의 유연한 탐지 기능을 추가한 SDN과 지능형 공격자를 탐지하기 위한 정보 수집용 허니팟을 이용하여 기존 연구의 단점을 보완한 방어 기법을 설계하고 지능형 공격에도 대응할 수 있는 방법을 제시하고자 한다. 이를 위해 SDN의 소프트웨어 기반 유연한 트래픽 관리 특성을 기반으로 DoS 공격을 탐지하고 차단하는 기법을 설계한다. 제안 시스템에서는 SDN 스위치가 수집한 트래픽 분석 데이터를 기반으로 SDN 컨트롤러에 포함된 공격 탐지 모듈이 DoS 공격 발생 여부를 판단한다. DoS 공격이 탐지된 경우 SDN 컨트롤러는 공격 패킷을 허니팟으로 포워딩하도록 하는 플로우 규칙을 생성하여 SDN 스위치에 적용한다. 이를 통해 공격 패킷이 공격 타깃 시스템 대신 허니팟으로 전달되도록 할 수 있다. 허니팟은 공격이나 시스템 구성의 취약성을 모니터링 할 수 있는 컴퓨팅 리소스이다[4]. 허니팟을 통해 공격 패턴을 분석하거나 보안 취약점 등 개선이 필요한 부분들을 찾아낼 수 있다. 허니팟은 관리자가 원하는 부분의 보안성 조사를 가능하게 한다. 이를 기반으로 네트워크 침입 탐지 시스템(Network Intrusion Detection System, NIDS)의 늘어나는 오탐지율과 지속적으로 발전하는 공격자의 정교한 탐지 회피 기술 등의 문제점을 보완할 수 있다. 본 논문에서는 탐지된 공격 패킷을 허니팟으로 유도하여 공격 패킷을 분석하고 이를 기반으로 제안 시스템의 공격 탐지 모듈 공격 탐지 기준과 위협 레벨 관리 모듈의 타이머 시간 값을 동적으로 수정할 수 있도록 설계한다. 허니팟은 공격 패킷의 공격 패턴, 공격 수준 등을 분석하여 SDN 컨트롤러에 포함된 공격 탐지 모듈이 공격을 더 잘 탐지하고, 대응할 수 있도록 파라미터를 조정한다. 이를 통해 공격이 성공적으로 차단되었는지 여부를 확인하고 공격 패턴, 공격 수준을 조정하여 공격하는 지능형 공격 역시 차단할 수 있다.

본 논문에서 제시한 시스템의 DoS 공격 차단 성능을 보이기 위해 SDN 컨트롤러인 개방형 네트워크 운영 체제(Open Network Operating System, ONOS)[5]와, SDN 가상 스위치인 OpenVSwitch(OVS)[6]를 사용해 제안 시스템을 구현하였다. ONOS는 SDN 환경에서의 네트워크 기능 가상화(Network function virtualization, NFV) 기능을 제공하는 오픈 소스(Open source) SDN 컨트롤러이다[7]. OVS는 SDN 가상환경에서 사용되는 다계층 가상화를 지원하는 오픈 소스 가상 스위치이다[8]. OVS는 OpenFlow 프로토콜을 기반으로 작동한다. OpenFlow는 SDN에서 Southbound API를 통해 내부 플로우 테이블(Flow table)과 플로우 항목을 추가 및 제거하기 위한 표준화된 이더넷 스위치 기반 프로토콜이다[9]. 공격 탐지를 위한 패킷 통계 모듈은 OVS에 내장된 ovs-tcpdump를 사용해 구현했으며, 패킷 통계량은 TCP 소켓 통신을 통해 ONOS 컨트롤러로 전송되도록 하였다. 허니팟으로 포워딩된 공격 패킷의 분석을 위해 명령어 인터페이스(Command Line Interface, CLI) 기반 패킷 캡처 소프트웨어인 TShark[10]를 활용하였다. 소켓 통신을 포함한 이외의 시스템의 모든 통신 및 처리 기능은 Python을 사용해 직접 구현하였다. 구현된 시스템에서의 공격 상황을 시뮬레이션 하기 위해 네트워킹 시뮬레이션 툴인 mininet[11, 12]과 네트워크 공격 툴인 hping3[13]를 사용하였다.

본 논문의 구성은 다음과 같다. 2장에서 SDN, DoS 공격, 허니팟 등 관련 연구와 SDN을 활용한 기존의 DoS 공격 차단 기법을 소개한다. 3장에서 제안 시스템의 구조와 처리 흐름을 설명한다. 4장에서 널리 사용되고 있는 SDN 컨트롤러 ONOS, SDN 스위치 OVS, OpenFlow 프로토콜을 사용한 제안 시스템의 실제 구현 방법을 제시한다. 5장에서 제안 시스템의 공격 차단 성능을 분석하고 실현 가능성을 보인다. 6장에서 결론을 내린다.

2. 관련 연구

2.1 소프트웨어 정의 네트워킹

5G, LoRA 등 광대역, 초고속 네트워크 기술이 발달함에 따라 네트워킹 기술의 적용 영역이 다양해지고 있다[14]. 따라서 네트워크 관리자가 전통적인 네트워킹 방식으로 다양한

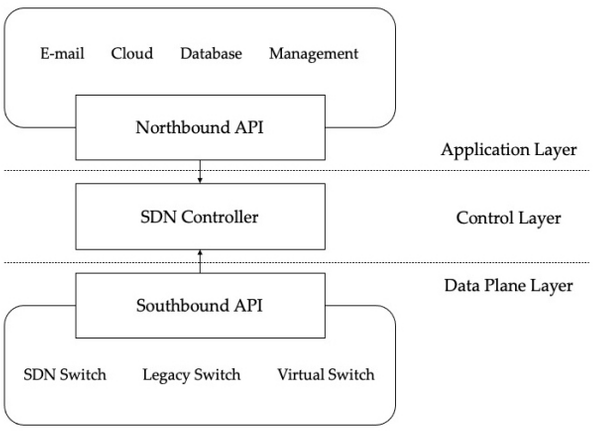


Fig. 1. SDN Architecture

프로토콜과 많은 데이터의 안정성과 보안 기준을 충족하기 어려워졌다. 특히 포워딩 규칙인 플로우 테이블과 실제 트래픽의 포워딩이 하나의 물리적 장비에서 이루어지는 전통적인 네트워크 환경에서는 네트워크 관리자가 기술의 변화에 따른 요구 조건을 수용하며 세밀한 관리 수준을 유지하기 어렵다. [15]의 논문에서는 이러한 문제를 해결하기 위해 관리자 영역과 데이터 통신 영역을 분리하여 효율성을 높인 네트워킹 메커니즘을 제안하였다. 이를 기반으로 트래픽을 소프트웨어 기반으로 제어하는 새로운 패러다임인 SDN이라는 기술이 제안되었다[16]. SDN 구조는 실제 트래픽 포워딩을 수행하는 데이터 플레인과 포워딩 규칙 등을 제어하는 컨트롤 플레인으로 구성된다. 데이터 플레인과 컨트롤 플레인이 분리된 구조를 통해 네트워크 제어를 집중화하고 네트워크 동작을 프로그램화함으로써 효율적으로 네트워크를 관리할 수 있도록 한다. Fig. 1은 SDN의 아키텍처를 보여주고 있다. SDN은 중앙 제어를 담당하는 컨트롤러를 중심으로 Northbound를 구성하는 응용 프로그램들과 Southbound를 구성하는 SDN 스위치들로 구성한다. 상위 계층에 포함된 애플리케이션과 컨트롤러는 REST API, SDN 컨트롤러가 제공하는 프로그래밍 언어용 SDK 등을 통해 통신한다. SDN 스위치와 컨트롤러는 OpenFlow 등 SDN 프로토콜을 통해 통신한다. SDN 컨트롤러는 플로우 규칙을 SDN 스위치의 플로우 테이블에 전달하여 의도하는 네트워크 정책을 구현한다. 애플리케이션은 플로우 규칙을 관리자가 정한 규칙, 이벤트 발생 등에 따라 동적으로 생성 및 수정하도록 함으로써 트래픽 제어를 유연하고 빠르게 수행할 수 있도록 한다.

2.2 서비스 거부 공격

DoS 공격은 악의적인 공격자가 장치가 정상적인 기능을 할 수 없도록 방해하여 정상 사용자가 컴퓨터 또는 기타 장치를 사용할 수 없도록 만드는 보안 공격이다. 2000년도 이후 여러 기관 및 유명 기업에 대한 공격이 이루어져 이슈화 된 이후 DoS 공격은 여전히 많은 기업과 기관들을 대상으로 시도되고 있으며 다양한 형태로 발전되었다. DoS 공격은 일반적으로 대상 시스템이 정상적으로 트래픽을 처리할 수 없을 때까지 요청을 반복하여 정상 사용자가 서비스를 사용할 수 없도록 한다[17]. 가장 일반적으로 많이 사용되는 DoS 공격으로는 대량의 패킷을 피해자에게 보내는 ICMP flooding 공격과 UDP(User Datagram Protocol)의 약점을 이용하여 대량의 패킷을 보내는 UDP flooding, DNS 메시지를 이용하여 트래픽을 증폭시키는 DNS flooding 공격 그리고 SYN flooding 공격 등이 있다. SYN flooding 공격은 TCP에서 서버-클라이언트 간의 신뢰성 있는 연결을 위한 메커니즘인 3방향 교신(3-Way-Handshake)과정을 악용한 공격으로 대량의 SYN 요청 패킷을 전송함으로써 서버의 큐가 SYN패킷으로 가득 채워 다른 연결을 할 수 없게 하는 공격이다.

또한 DoS 공격의 발전된 형태로 분산 서비스 거부(Distributed denial of service, DDoS) 공격이 있다. DDoS 공격은 수많은 여러 출발지로부터 하나의 목적지를 공격 타깃으로 공격 패킷을 보냄으로써 공격 타깃 컴퓨터가 작동 하지 못하도록 하는 공격이다[18]. 본 논문에서는 이러한 DoS 공격 중, SYN flooding 공격과 다수의 공격자로부터 공격 패킷이 오는 DDoS 공격을 가정하여 실험하였다. SYN 플러딩 공격 감지를 위해 SDN 스위치에서 플래그 데이터를 수집, 분석하고 이를 기반으로 공격을 감지하도록 설정하였다. SDN 스위치는 패킷이 들어오면 미리 정해진 통계 정보를 SDN 컨트롤러에 전송한다. 전송된 정보 중 SYN 플래그 비율이 전체 패킷 대비 설정한 임계치 이상의 비율이 되면 SYN flooding 공격이 발생한 것으로 판단한다. 또한, IP 주소의 Entropy값을 이용하여 DDoS 공격을 탐지하도록 설계 하였다. 하나의 목적지로 향하는 다수의 패킷이 다른 출발지 주소로부터 오는 경우, 출발지 주소 Entropy가 미리 설계해둔 임계치 이상으로 올라가면 DDoS 공격으로 판단한다.

이러한 서비스 거부 공격의 형태는 지능형 공격자의 의해 더욱 발달되어 공격을 감지하는데 점점 더 어려워지고 있다. 지능형 지속 공격(Advanced persistent threat, APT)은 공격을 가할 특정 대상이 액세스할 수 있는 데이터를 감시, 추출 또는 조작하려는 공격 유형이다. 공격자는 공격대상의 취약점을 분석하거나 발견될 때까지 감시하여 약점을 이용하거나, 여러 취약점들을 조합하여 공격으로 재구성할 수 있다[19].

2.3 기존 차단 기법

Table 1은 SDN을 활용하여 DoS 공격에 대응하는 기존 연구들과 제안 시스템을 비교하였다. SDN 환경에서 일반적인 DoS 공격을 방어하는 방법은 Rate Limiting 방식으로 DoS 공격을 차단 한다[20, 24, 25]. 하지만 이는 정상 패킷이 차단될 확률이 높고 공격자를 특정할 수 없다. SDN 특성인 PACKET_IN 메시지를 통한 모니터링 및 공격 탐지 기법들도 있다. [26]의 연구에서는 기존 연구에서 PACKET_IN 메시지와 ARP 메시지의 비교를 통해 지능적인 ARP Poisoning에 대응하는 방안을 제시하였다. 이 연구에서는 본 연구와 유사하게 타이머가 사용되었으며, 타이머를 우회하기 위한 지능

Table 1. Comparison of Existing Researches and Proposed System

	Choi[20]	You et al.[21]	Sanguankotchakorn [22]	Wang et al.[23]	Proposed System
Detection Criteria	Bandwidth Limit	IP entropy	IP entropy	Rule-based	Rule-based
Detection Subject	Controller	Controller	Switch	Controller	Controller + Switch
Attack Packet	Drop	Drop	Drop	Forward to Honeypot	Forward to Honeypot
Honeypot	X	X	X	O	O
Intelligent Attack Detection	X	X	X	X	O

적 공격의 대응책으로 PACKET_IN 메시지를 기반으로 하는 분석 수행하였다. 본 연구와 차이점은 분석이 수행되는 위치가 컨트롤러로, 높은 부하 발생의 가능성이 있지만 이 연구에서는 공격 탐지와 분석의 주체가 컨트롤러와 허니팟으로 각각 분리되어 있으므로 비교적 부하 발생의 가능성이 낮다. [21]의 연구 또한 PACKET_IN 메시지에 포함된 출발지 주소, 포트 정보 등을 기반으로 여러 공격자가 동일한 서비스에 접근을 시도하는지 검사 하였다. 해당 논문에서 제시한 PACKET_IN 메시지를 활용한 DDoS 공격 탐지 기법은 여러 공격자가 동시에 공격을 수행하는 DDoS 공격 탐지에는 적합하지만 실제 패킷의 플래그 통계 정보 등을 기반으로 해야 하는 SYN Flooding과 같은 공격은 탐지하기 어렵다. 또 다른 DDoS 탐지 기법으로 [22]의 연구에서는 기존 연구에서 IP 엔트로피를 활용해 DoS 공격 탐지하는 방법을 제시하였다. SDN 환경에서 DDoS 공격을 탐지하는 가능성을 보여주었고 본 논문은 이러한 방법을 차용하여 공격탐지모듈에서 DDoS 공격을 탐지하는 방법으로 활용하였다.

SDN 환경에서 허니팟을 활용한 기법들도 있었다. [23]의 연구에서는 플로우 규칙을 기반으로 하는 SDN에서의 허니팟 구조를 제시하였다. 하지만 본 연구와 달리 허니팟에서 수집된 정보를 공격 탐지에 다시 활용하는 방법이 제시되지는 않았다. 본 연구에서의 허니팟은 단순히 공격자를 잘못된 정보로 유도하고 공격 데이터를 수집하는 것을 넘어 수집된 데이터를 가공해 다시 공격 탐지에 활용할 수 있도록 설계 하였다. [27]의 연구에서는 거짓 뷰(View)를 생성하고 서비스에 접속하는 각 호스트마다 다르게 적용함으로써 허니팟으로 유도하는 방법을 제시하였다. 하지만 호스트의 수가 많아질수록 뷰를 관리하기 위한 오버헤드가 발생하는 문제점이 있다. SDN 환경에서 리소스의 관한 문제는 해결해야 하는 과제이다. [28]의 연구에서는 SDN 스위치에서의 모니터링 단계에서 발생하는 리소스 문제에 대해 논의하였고 이를 통한 최적의 조건들을 기술 하였다. 본 연구에서는 이러한 논의를 참조하여 모니터링 단계에서 부하가 발생하는 리소스 중 트래픽 부분에 대한 부하를 최소화하기 위해 SDN 스위치에서 1차 통계 데이터화 후 컨트롤러로 보내도록 설계하였다.

또 다른 기존 연구로서 SDN 컨트롤러에 대한 DoS 공격을 완화하기 위해 Flowsec과 Black-box를 이용한 방안이 제안되었다[29]. Flowsec은 공격자가 시간당 전송할 수 있는 패킷에 수를 제한하여 공격을 완화시키고자 하였다[30] 하지만 이는 일반적인 사용자의 패킷 또한 제한할 수 있는 문제점을 가지고 있다. Black-box는 위협이 되는 단계를 정의하는 유한 상태 머신을 활용하여 공격을 차단하는 방법을 고안하였다. 하지만 지능적 공격자가 방어의 동작을 파악하여 이를 피해 공격을 할 수 있는 약점이 있고 본 논문에서는 Black-box에서 제시한 유한 상태 머신을 활용하고 허니팟이라는 공격 우회 장치를 추가하여 더욱 동적으로 공격에 대응할 수 있는 방법을 제시한다. 허니팟을 사용하여 공격 탐지 시스템의 탐지 기준 등을 조정함으로써 지능적 공격에도 유연하게 대응할 수 있다.

3. 시스템 제안

본 논문에서 제안하는 시스템은 DoS 공격 탐지 모듈을 활용한 이벤트 기반 플로우 규칙 생성 애플리케이션을 적용함으로써 DoS 공격 발생 시 자동으로 공격 패킷을 허니팟으로 포워딩하도록 한다. 이러한 방법은 관리자가 직접 네트워크를 모니터링하여 규칙을 생성하는 전통적 방식에 비해 빠르게 공격 패킷을 허니팟으로 유도할 수 있다. 또한 허니팟에 수집된 공격 분석 데이터를 기반으로 공격 탐지 모듈의 탐지 기준을 조정함으로써 공격 패턴과 공격 수준을 조정하여 공격 탐지를 회피하려는 지능적 공격에도 유연하게 대응할 수 있다. 본 연구는 SDN과 허니팟을 통해 공격 패킷을 분석하고 지능형 공격이 탐지 될 수 있도록 설계하여 설정된 공격 탐지 모듈의 기준을 우회한 지능형 공격을 감지하면 이에 동적으로 대응하는 기준으로 변경할 수 있도록 설계하였다.

본 논문은 기존 연구들에 장점을 차용하고 단점들을 보완하여 시스템을 구현하였다. 공격을 탐지하는 기준으로 패킷 비율, IP 엔트로피 등 다양한 규칙 기반의 탐지 기준을 제시하였고 공격 탐지 주체 또한 SDN 컨트롤러와 SDN 스위치를 모두 활용하여 공격에 대한 탐지 부분을 분산하여 좀 더 안전하고 효율적으로 설계 하였다 또한 공격 패킷을 허니팟으로 전송하여 지능적 공격자의 공격 대응을 할 수 있도록 설정 하였다. 이는 기존 연구들에 장점들 차용하고 단점들을 보완하여 시스템을 구현하였다. SDN의 장점은 관리자가 트래픽 분석을 통해 전체 네트워크의 트래픽 흐름을 동적으로 조정하여 변화하는 요구를 충족 하고 직접 프로그래밍을 통해 공격의 대응할 수 있다는 것이다. 본 논문에서는 이러한 장점을 활용하여 SDN 컨트롤러와 SDN 스위치 각각의 분석모듈, 공격탐지 모듈, 위협관리 모듈 등을 프로그래밍하여 악의적인 사용자의 트래픽을 허니팟으로 유도하는 DoS 공격 차단 방법을 제시한다.

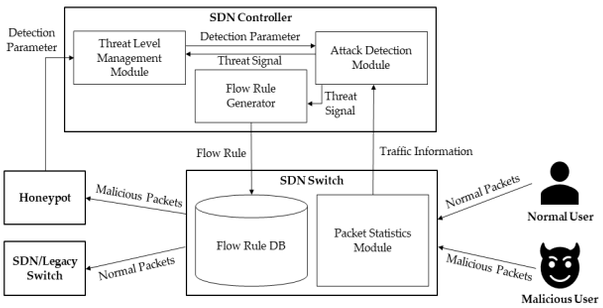


Fig. 2. Proposed System Architecture

Table 2. Honeypot Induction Flow Rule Example

#	Attack	Condition	Flow Rule
1	Simple DoS attack	ETH_SRC: aa:aa:aa:aa:aa:aa	OUTPUT: 3
2	Dos attack with source address modification	PKT_SIZE: 1024; TCP_FLAG: SYN	OUTPUT: 3
3	Dos attack with source address modification and massive packet	IP_FRAG: 5; TCP_FLAG: SYN	OUTPUT: 3

3.1 제안 시스템 구조

Fig. 2는 본 논문에서 제안하는 시스템의 구조를 도식화한 것이다. 제안 시스템은 SDN 컨트롤러와 SDN 스위치, 레거시 스위치, 허니팟, 사용자로 구성된다. 사용자는 정상 사용자와 악의적 사용자로 구분할 수 있다. SDN 컨트롤러는 SDN 스위치가 수집한 트래픽 정보를 기반으로 공격자를 판단하고, 공격 트래픽을 허니팟으로 유도하기 위한 플로우 규칙을 생성한다. SDN 스위치는 트래픽 정보를 수집하여 SDN 컨트롤러로 전달하고, 플로우 규칙에 따라 트래픽을 포워딩한다. SDN 스위치의 다음에 위치하는 SDN/레거시 스위치는 전달받은 정상 패킷을 목적지로 전달한다. 허니팟은 DoS 공격에 대한 정보를 수집하고, 이를 기반으로 지능적 공격에 대응하기 위한 위협 레벨 관리 모듈 타이머 조정 및 공격 탐지 기준 조정을 수행한다.

SDN 컨트롤러는 공격 탐지 모듈(Attack Detection Module)과 플로우 규칙 생성기(Flow Rule Generator), 위협 레벨 관리 모듈(Threat Level Management Module)을 포함한다. 공격 탐지 모듈은 SDN 스위치가 수집한 패킷 데이터를 기반으로 악의적 사용자의 트래픽을 판단한다. 이 때, DoS 공격의 유형에 따라 적절한 탐지 기준을 결정하고, 이에 따라 공격 여부를 판단한다. SYN Flooding 공격의 경우, 전체 TCP 패킷 중 SYN 플래그를 포함하는 패킷의 비율을 이용해 공격을 탐지할 수 있다. 여러 좀비 PC로부터 단시간에 많은 패킷이 동시에 전달되는 DDoS 공격의 경우에는 IP 주소의 엔트로피를 이용해 공격을 탐지할 수 있다. 동일한 목적지에 대해 평소와 다르게 비정상적으로 많은 출발지로부터 패킷이 전송되는 경우 이를 공격으로 간주하여 패킷의 해당 출발지로부터 발생한 패킷을 허니팟으로 포워딩하도록 한다. 플로우 규칙 생성기는 실제로 공격자의 패킷을 허니팟으로 유도하기 위한 플로우 규칙을 생성하고 SDN 스위치로 전달한다. Table 2는 SYN Flooding 공격 발생 시 플로우 규칙 생성기가 생성하는 허니팟 유도 플로우의 예시이다. 출발지 주소를 변조하지 않는 가장 단순한 형태의 DoS 공격에서 악의적 사용자의 MAC 주소는 aa:aa:aa:aa:aa:aa이라 가정한다. 허니팟의 포트 번호는 3이다. 1의 플로우 규칙을 적용함으로써 해당 MAC 주소를 가지는 악의적 사용자로부터 발생한 모든 패킷은 허니팟으로 전달되어 기록된다. 이러한 플로우 규칙을 회피하기 위해 공격자는 출발지 주소를 변조하여 공격 패킷을 전송할 수 있다. 이러한 경우 비정상적으로 큰 크기의 패킷을 반복적으로 보내는 SYN Flooding 공격의 특징에 기반을 두어 IP 헤더의 fragment offset 값이나 패킷 크기를 기준으로 공격 패킷을 허니팟으로 포워딩하도록 할 수 있다. 2의 플로우 규칙에서는 공격자가 1024바이트 크기의 패킷을 지속적으로 보내는 상황을 가정하여 패킷 사이즈가 1024인 SYN 패킷을 허니팟으로 포워딩한다. 크기가 큰 패킷의 경우 3의 플로우 규칙과 같이 fragment offset 기준을 정하여 해당 값보다 큰 값을 가진 SYN 패킷을 허니팟으로 포워딩한다. 예시로 든 공격 기법 외에도 공격 기법, 공격 양상에 따라 시스템 관리자가 허니팟 유도 플로우 규칙은 다양하게 정의할 수 있다. 미리 설정한 허니팟 유도 플로우 규칙에 의해 DoS 공격 패킷이 정상적으로 허니팟으로 포워딩되지 않는 경우 관리자는 규칙을 적절히 수정할 수 있어야 한다.

위협 레벨 관리 모듈은 DoS 공격의 발생 여부를 나타내고, 위협 레벨을 관리한다. 위협 레벨 관리 모듈의 위협 레벨은 위협 없음(No Threat), 낮은 위협(Low), 높은 위협(High)으로 구분된다. 위협 레벨에 따라 공격 탐지 모듈의 공격 탐지 규칙을 조정한다. 이를 통해 위협이 없는 상태에서는 탐지 기준을 높게 설정하여 정상 패킷의 오탐지를 줄이고, 높은 위협 상태에서는 탐지 기준을 낮춰 공격을 더 예민하게 탐지할 수 있다. 탐지 기준은 DoS 공격의 종류에 따라 다르게 설정할 수 있다. 위협 레벨 관리 모듈은 위협 레벨의 하향 조정을 위해 타이머(Threat Level Management Module Timer, TLMM Timer)를 사용한다. 위협 레벨 관리 모듈 타이머는 허니팟에 수집된 공격 패킷을 기반으로 갱신되어 제안 시스템이 지능적인 공격자에 의한 공격에 유연하게 대응할 수 있도록 한다. Fig. 3은 위협 레벨 관리 모듈의 위협 레벨 조정 조건과 순서를 도식화한 것이다. 초기 상태인 위협 없음 단계에서 최초 공격이 탐지된 경우 위협 레벨을 낮은 위협으로 변경하고, 위협 레벨 관리 모듈 타이머를 가동한다. 위협 레벨 Low 단계에서 추가 공격이 탐지되면 위협 레벨을 High로 조정한다. 탐지된 모든 패킷은 목적지로 전송하지 않고 허니팟으로 전송된다. 추가 공격이 감지되지 않고 타이머가 종료된 경우 위협 레벨을 No Threat로 조정한다. 위협 레벨이 High인 경우 DoS 공격에 더 빠르고 예민하게 대응하기 위해

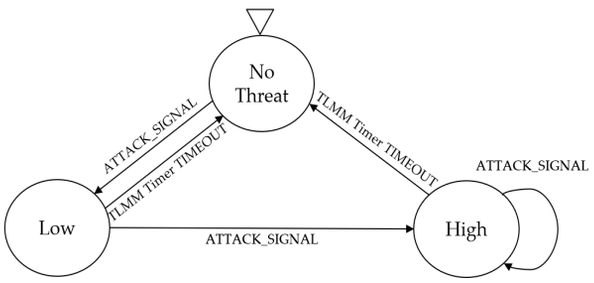


Fig. 3. Threat Level Management Module States

공격 탐지 기준을 상향시키고, 타이머가 종료되면 위협 레벨을 No Threat로 조정한다. 위협 레벨 관리 모듈의 타이머 값은 허니팟에 수집된 공격 패킷 데이터를 기반으로 조정될 수 있다. 이를 통해 위협 레벨 관리 모듈이 타이머를 기준으로 위협 레벨을 하향 조정한다는 사실을 파악한 지능형 공격자의 공격 패턴 기반 지능적 공격을 대응할 수 있다. 공격 패턴 기반 지능적 공격은 의도적으로 타이머보다 긴 시간대기 후 공격 패킷을 전송하여 위협 레벨을 낮은 상태로 유지시키는 공격을 의미한다. 위협 레벨 관리 모듈은 위협 레벨 조정 시 위협 레벨 상향/하향 메시지를 허니팟으로 전송한다. 위협 레벨 조정 메시지 전달받은 허니팟은 이를 기반으로 타이머 값의 적합성을 분석하여 새로운 위협 레벨 관리 모듈 타이머 값을 계산하고 해당 값을 위협 레벨 관리 모듈에게 전달한다. SDN 스위치는 패킷 통계 모듈과 플로우 규칙 데이터베이스를 포함한다. 패킷 통계 모듈은 SDN 스위치로 들어온 모든 패킷을 수집하여 통계 정보를 생성하고, 이를 SDN 컨트롤러로 전송한다. 플로우 규칙 데이터베이스는 SDN 컨트롤러부터 전송받은 플로우 규칙을 저장한다. SDN 스위치는 플로우 규칙 데이터베이스에 저장된 플로우에 따라 스위치로 들어오는 패킷을 포워딩 한다. 플로우 규칙 데이터베이스는 기본적으로 모든 패킷을 주소에 따라 다음 스위치로 전달하는 반응형 포워딩(Reactive Forwarding) 플로우 규칙을 포함한다. 허니팟 유도 플로우는 반응형 포워딩 플로우보다 높은 우선순위를 가지도록 생성되어 악의적 사용자의 패킷이 목적지 주소와 상관없이 허니팟으로 전달될 수 있도록 한다.

허니팟에 수집된 공격 패킷 데이터를 기반으로 파라미터를 조정함으로써 막을 수 있는 또 다른 지능적 공격으로는 공격자의 순차적 공격 수준 조정을 통해 시스템이 DoS 공격으로 판단하는 기준을 파악하여, 공격이 탐지되지 않는 수준에서 공격을 수행할 경우이다. 공격 수준 기반 지능적 공격은 공격자가 공격 패킷 수를 조정하여 시스템이 가지고 있는 탐지 파라미터보다 낮은 수준의 패킷(SYN flooding 공격의 경우 패킷에서의 SYN 비율을 기준으로 탐지 파라미터 설정)의 공격 패킷을 전송하여 위협 레벨을 낮은 상태로 유지시키는 공격을 의미한다. 위협 레벨 모듈은 위협 레벨 조정 시 위협 레벨을 상향 조정하고, 공격 탐지 모듈은 이에 따라 탐지 기준을 높인다.

SDN 스위치 다음에 위치한 SDN/레거시 스위치는 SDN

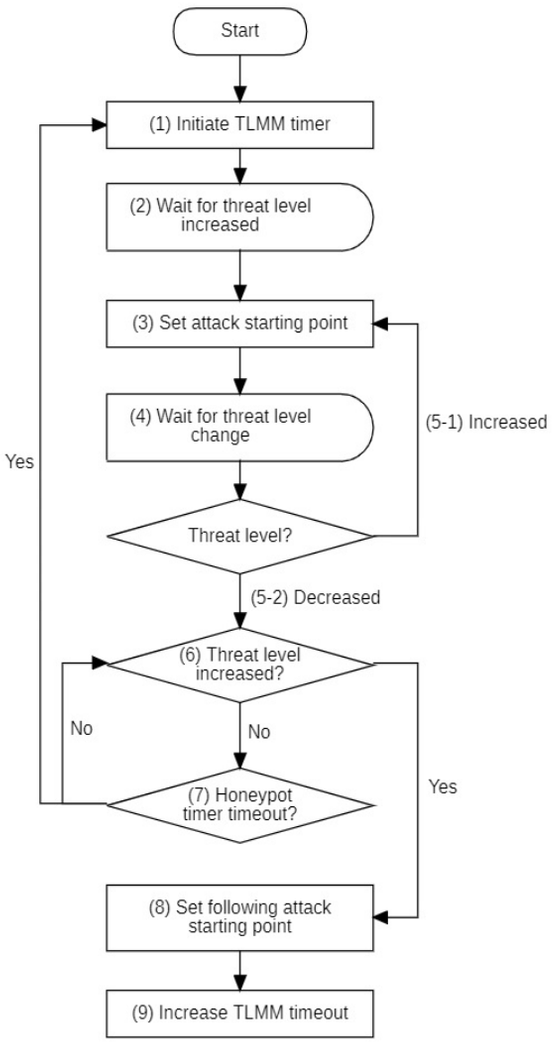


Fig. 4. Honeypot Time Adjustment Flow

스위치로부터 전달받은 패킷의 주소 정보에 따라 실제로 패킷을 포워딩한다. 허니팟은 공격 탐지 모듈이 공격으로 판단한 패킷을 수집하고 이를 분석한다. 공격자들의 공격 패턴, 공격 수준 유형을 분석하여 공격의 지속성을 파악하고 공격 패킷을 차단한다. 또한 이를 바탕으로 SDN 컨트롤러에 포함된 위협 레벨 관리 모듈의 타이머를 조정한다.

Fig. 4는 허니팟에서 위협 레벨 관리 모듈 타이머를 조정하는 동안의 처리 흐름을 도식화한 것이다. (1)허니팟 최초 가동 시 타이머의 타임아웃 값을 초기화한다. 타임아웃 초기 값은 관리자가 필요에 따라 조정할 수 있다. (2)허니팟은 SDN 컨트롤러에 포함된 위협 레벨 관리 모듈의 위협 레벨 변동 신호를 수신하기 위해 대기하며 공격 패킷을 수집, 분석한다. (3)위협 레벨이 상향되면 첫 번째 공격의 시작 지점을 기록한다. (4)이 후 위협 레벨의 변화를 감지하며 대기한다. 위협 레벨의 변화가 있을 시 (5-1)위협 레벨이 상승하면 (3)로 이동하여 다시 해당 공격을 첫 번째 공격의 시작 시점으로 기록하고 (5-2)위협 레벨이 하향되면 (6)위협 레벨의 상향 조정을

대기한다. (7)위협 레벨이 상향되지 않으면 시간 만료 타이머가(Honeypot timer) 종료 여부를 확인하며 대기한다. 시간 만료 타이머는 공격 여부에 따른 상태 전이의 기준이 되는 타이머로, 현재의 위협 레벨이 어느 단계에 있더라도 시간 만료 타이머가 종료되면 위협 레벨은 위협 없음(No Threat)로 전이한다. 위협 레벨의 상승을 탐지하고 시간 만료 타이머가 종료되면 (1)로 이동하여 위협 레벨 관리 모듈 타이머 타임아웃 값을 초기화한다. 위협 레벨이 상향 되면 (8)두 번째 공격 시작 포인트를 기록하고 (9)기록된 공격 시점 정보를 기반으로 새로운 타임아웃 값을 계산하고 위협 레벨 관리 모듈의 타이머 값을 계산한 값으로 설정한다.

$$A = \{a_1, a_2, \cdots, a_n\} \quad (1)$$
$$D = \{d_i | d_i = t(a_{i+1}) - t(a_i), 1 \le i \le n-1\}$$
$$d_{threshold} = \frac{\sum_{i=1}^{n} d_i}{|D|}$$
$$d_{timeout} = \max(D) + C$$

Equation (1)은 허니팟이 수집한 공격 데이터를 기반으로 새로운 위협 레벨 관리 모듈 타이머 타임아웃을 계산하는 과정을 나타낸 수식이다. 허니팟이 수집한 공격 시작 지점 패킷 $a_1, a_2, \cdots, a_n$은 배열 A에 저장한다. $t(a_i)$는 i번째 공격 시작 지점 패킷의 도착 시간을 의미한다. 위협 관리 모듈 타이머가 끝나기 전에 또 다른 공격 패킷이 들어오면 이를 첫 번째 공격으로 취급한다. 위협 관리 모듈 타이머가 끝난 뒤에 들어온 공격 패킷은 이어지는 공격으로 설정된다. 배열 D는 A에 저장된 모든 공격 사이의 시간 간격 $d_n = t(a_{n+1}) - t(a_n)$을 원소로 포함한다. D에 포함된 모든 공격 간 시간 간격의 평균을 구하고 이 값을 $d_{threshold}$로 한다. 이 값은 공격이 수행되는 동안의 패킷 간 시간 간격과 공격자가 의도적으로 타임아웃을 대기하는 동안의 시간 간격을 구분하기 위한 기준 값으로 사용한다. $d_{timeout}$은 기존 타임아웃 보다 큰 값을 가지는 시간 간격 중 가장 큰 값의 상수를 더한 것으로 한다. 이렇게 함으로써 위협 레벨 탐지 모듈은 이전 공격의 패턴을 기반으로 더 안전한 타임아웃 값을 가질 수 있다.

3.2 공격 탐지 흐름

Fig. 5는 공격 탐지 흐름을 도식화 하였다. SDN 컨트롤러는 공격 탐지 모듈과 플로우 규칙 생성기, 위협 레벨 관리 모듈을 포함한다. SDN 스위치는 흐름 규칙 데이터베이스와 패킷 통계 모듈을 포함한다. (1)사용자로부터 패킷이 수신되면 (2)패킷 통계 모듈은 패킷 정보를 SDN 컨트롤러로 전달한다. 패킷 정보는 패킷의 송신자 주소, 패킷 종류, 플래그를 포함한다. 플래그는 SYN Flooding 공격 등 플래그를 사용한 DoS 공격에 사용한다. 패킷 전체를 전달하는 대신, 패킷 일부만을 전달함으로써 트래픽 부담을 최소화할 수 있다. (3)SDN 컨트롤러는 SDN switch의 패킷 수집 모듈로부터 패킷이 전달되면 공격 탐지 모듈은 SDN 스위치가 수집한 패킷 데이터를 기반으로 악의적 사용자를 판단한다. 악의적 사용자 판단에는 단위 시간 당 패킷 수, 대역폭 사용량, 패킷 종류별 비율 등의 판단 기준을 사용할 수 있다. 악의적 사용자 판단에는 단위 시간 당 패킷 수, 대역폭 사용량, 패킷 종류별 비율 등의 판단 기준을 사용할 수 있다. (4)악의적 사용자로 의심되는 사용자가 탐지된 경우, 플로우 규칙 생성기와 위협 레벨 관리 모듈에게 위협 신호를 각각 전달한다. 위협 신호는 공격자 주소를 포함한다. (5)위협 레벨 관리 모듈은 위협 신호를 분석하여 위협 레벨을 업데이트한 후 (6)위협 레벨에 따른 위협 감지 파라미터를 공격 탐지 모듈에게 전달한다. 위협 감지 파라미터는 단위 시간 당 패킷 개수, 대역폭 사용량, 패킷 종류에 따른 비율 등이 될 수 있다 (7)플로우 규칙 생성기는 위협 신호를 파악하여 플로우 규칙을 생성하고 (8)이를 SDN 스위치로 전송한다. SDN 스위치는 SDN 컨트롤러로부터 플로우 규칙이 전달되면 이를 플로우 규칙 데이터베이스에 저장하고 (9-1)플로우 규칙에 따라 악의적 사용자에 의한 패킷은 허니팟으로 전달하고, (9-2)정상 패킷은 다음 스위치로 전달한다. (10) 허니팟은 수집된 공격자의 공격 방식, 공격 패턴, 공격 유형, 공격 범위, 공격 시간 등을 분석하여 이를 바탕으로 공격 탐지 모듈의 위협 감지 파라미터를 조정하고 위협 레벨 관리 모듈에 반영한다.

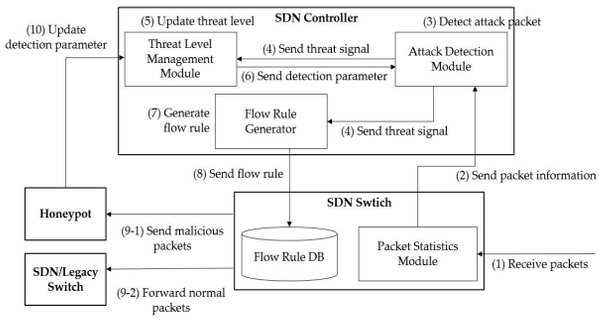


Fig. 5. Attack Detection Flow

4. 성능 평가

본 논문에서 제시한 시스템의 DoS 공격 차단 성능을 보이기 위해 SDN 컨트롤러인 ONOS, SDN 스위치인 OpenVSwitch를 사용해 시스템을 구현하였다. 공격 탐지를 위한 패킷 통계 모듈은 OpenVSwitch에 내장된 ovs-tcpdump를 사용해 구현했으며, 패킷 통계량은 TCP 소켓 통신을 통해 ONOS 컨트롤러로 전송되도록 하였다. 허니팟으로 포워딩된 공격 패킷의 분석을 위해 CLI 기반 패킷 캡처 소프트웨어인 TShark를 활용하였다. 소켓 통신을 포함한 이외의 시스템의 모든 통신 및 처리 기능은 Python을 사용해 직접 구현하였다. 구현된 시스템에서의 공격 상황을 시뮬레이션 하기 위해 네트워킹 시뮬레이션 툴인 mininet과 네트워크 공격 툴인 hping3를 사용하였다. 실험에 사용된 시스템의 사양은 Table 3과 같

Table 3. Experimental System Environment

Controller (ONOS)	
CPU	Intel i5-3220M Quad Core Processor
RAM	8GB
LAN	1Gbps LAN
SDN Switch (OVS)	
CPU	Mediatek MT7620A
RAM	128MB
LAN	4-ports 1Gbps LAN

Fig. 6. Experiment Environment

다. Fig. 6은 실제 실험 수행을 위해 실험 환경을 구성한 모습이다. Switch1은 스위치용 커스텀 운영체제인 OpenWRT를 설치하여 OpenVSwitch 서비스를 설치, 구동하였다. Laptop1은 Switch1과 연결하여 컨트롤러와 허니팟의 역할을 수행하도록 하였다. 이 때, 커널 기반 가상화 툴인 docker를 사용해 컨트롤러 가상머신과 허니팟 가상머신을 분리하여 구동하였다. Laptop2는 공격 대상으로, 80번 포트를 개방하여 웹 서버를 구동하였으며, CPU 사용량 정보를 지속적으로 기록하도록 설정하였다.

Fig. 7은 실험 수행을 위해 ONOS 컨트롤러와 mininet을 활용해 구축한 시뮬레이션 환경의 토폴로지다. c0은 SDN 컨트롤러로, 작업 환경 호스트에서 작동하는 ONOS와 원격으로 연결했다. s1은 SDN 스위치, s2는 레거시 스위치이다. h1은 공격자로 해당 노드에서 hping3를 이용해 피해자인 h2로 SYN Flooding 공격을 수행했다. h3는 허니팟 노드로, SDN 스위치의 2번 포트에 연결되어 있다. SDN 스위치는 공격 패킷을 2번 포트로 포워딩하도록 하는 플로우 규칙을 포함한다. Fig. 8은 커널 기반 가상화 툴인 docker를 활용해 실제로 공격 상황을 시뮬레이션하고 시스템의 작동을 실험하는 화면이다. 좌측 상단은 ONOS가 작동 중인 컨트롤러 가상머신이다. 우측 상단은 컨트롤러 가상머신에서 위협 레벨 관리 모듈이 작동 중이다. 좌측 하단은 허니팟 가상머신으로, 자신에게 들어오는 공격 패킷을 분석하고 새로운 타임아웃 값을 계산하여 컨트롤러 가상머신의 위협 레벨 관리 모듈로 타임아웃 갱신 메시지를 전송한다. 우측 하단은 공격자로 피해자 노드(172.17.0.3)로 SYN Flooding 공격을 전송한다.

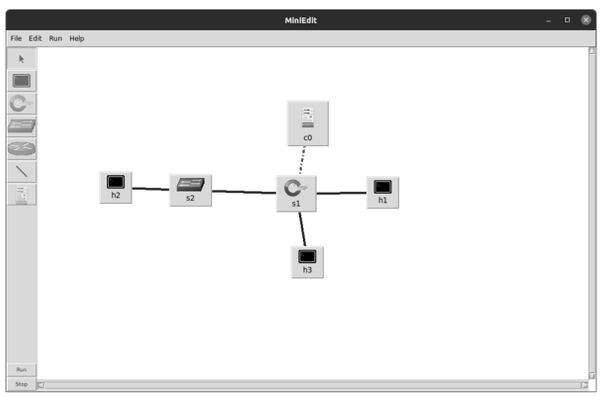


Fig 7. Simulation Environment Topology

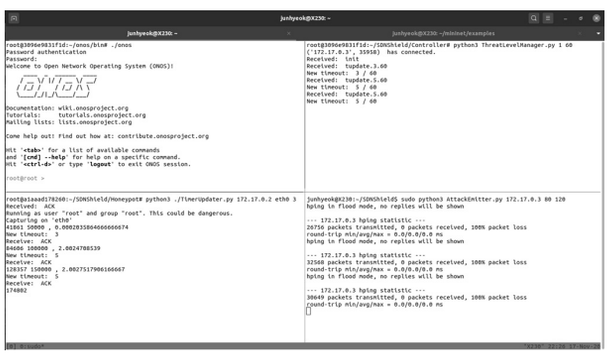


Fig. 8. Experiment Process

4.1 공격 패턴 기반 지능적 공격 차단 성능

제안 시스템의 공격 차단 성능을 보이기 위해 악의적 공격자에 의한 지능적 공격 시나리오를 구성하고 실험을 수행하였다. 공격자는 본 시스템이 위협 레벨 조정을 위한 타이머를 가지고 있다는 것을 알고 있으며, 낮은 위협 레벨을 유지하기 위해 의도적으로 타이머보다 긴 시간대기 후 DoS 공격 패킷을 전송한다고 가정하였다. Fig. 9는 실험에 사용된 지능적 공격 자동화 소프트웨어의 작동 화면이다. DoS 공격 툴인 hping3를 사용해 SYN Flooding 공격을 수행했으며, 총 3회에 나누어 공격 패킷을 전송했다. 각각 36,935개, 32,506개, 34,136개의 SYN 패킷을 전송했으며, 각 공격 사이의 시간 간격은 2분으로 설정했다.

Fig. 10은 허니팟에 포함된 타이머 업데이트 모듈이 작동

```
--- 172.17.0.3 hping statistic ---
36936 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
hping in flood mode, no replies will be shown

--- 172.17.0.3 hping statistic ---
32506 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
hping in flood mode, no replies will be shown

--- 172.17.0.3 hping statistic ---
34136 packets transmitted, 0 packets received, 100% packet loss
round-trip min/avg/max = 0.0/0.0/0.0 ms
```

Fig. 9. Pattern-based Intelligent Attack

```
343968 350000 , 2.002592850333334
New timeout:  12
Receive:  ACK
376953 400000 , 2.002592850333334
New timeout:  12
Receive:  ACK
417257 450000 , 2.002592850333334
New timeout:  12
Receive:  ACK
488239 500000 , 2.002592850333334
New timeout:  12
Receive:  ACK
```

Fig. 10. Honeypot Timer Update Module Operation

하는 화면이다. 타이머 상수 C는 10으로 설정하였다. 위협 레벨 관리 모듈의 초기 타임아웃 값은 1분으로 설정하였다. 공격자는 최초 공격 후 2분간 대기 후 공격 패킷을 전송하였다. 공격자가 2분간 대기한 후 공격 패킷을 전송했을 때 타이머 업데이트 모듈은 이러한 패턴을 분석하여 대기 시간 2분 + 상수 10분을 더한 12분을 새로운 위협 레벨 관리 모듈의 타이머 값으로 설정하였다. Fig. 9에서와 같이 343968, 376953,417237개의 패킷이 전송됨을 볼 수 있고 각각 2분 간격으로 들어온 공격에 대해 허니팟이 상수 10을 더한 새로운 타임아웃 값으로 모듈을 작동시키는 것을 확인할 수 있다. 공격 패킷 수집을 통해 의도적으로 대기 시간을 갖는 지능적 공격의 패턴을 분석함으로써 위협 레벨 관리 모듈은 지능적 공격에도 적절한 수준의 위협 레벨을 유지하고 DoS 공격에 대응할 수 있다.

Fig. 11은 지능적 공격 발생에 따른 제안 시스템과 허니팟을 활용한 타이머 갱신이 없는 시스템의 위협 레벨 변화를 나타낸 것이다. 굵은 점선으로 표시된 것이 제안 시스템의 위협 레벨, 짧은 점선으로 표시된 것이 타이머 갱신이 없는 시스템의 위협 레벨이다. 동적 위협 레벨,(Adaptive Threat level, AT) 고정 위협 레벨(Fixed Threat level, FT)로 표시된 음영 부분은 제안 시스템과 타이머 갱신이 없는 시스템 모두에서 공격 탐지에 실패한 부분이며, (FT)로 표시된 음영 부분은 타이머 갱신이 없는 시스템에서 공격 탐지에 실패한 부분이다. 총 22,590,240개의 SYN Flooding 공격 패킷을 2분 공격 후 2분대기 패턴으로 전송하였다. 타이머 갱신이 없는 시스

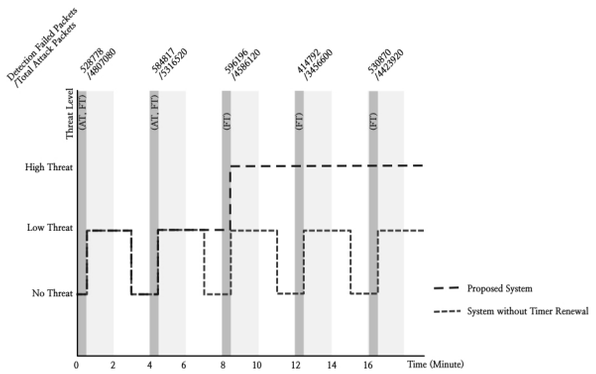


Fig. 11. Threat Level Change According to Attack Pattern-based Intelligent Attack

Table 4. Intelligent Attack Packet Detection Rate Comparison

	Proposed System	System without Timer Update
Detection Rate	95.7811%	89.2772%

템의 경우, 낮은 위협 - 타이머 만료 - 위협 없음 상태가 반복적으로 나타내며 대기 시간을 둔 지능적 공격에 적절하게 대응하지 못하고 있음을 확인할 수 있다. 반면, 제안 시스템의 경우 지능적 공격이 발생한 직후에는 타이머 갱신이 없는 시스템과 동일하게 위협 레벨이 하향 조절되었으나, 공격 패킷 분석을 통해 이후 발생하는 지능적 공격을 하나의 공격으로 인식하여 높은 위협 레벨을 유지하고 있음을 확인할 수 있다. 결과적으로 Table 4에서 나타난 것처럼 제안 시스템의 공격 패킷 차단율이 95.2832%로 타이머 갱신이 없는 시스템의 86.5424% 보다 높게 나타남을 확인할 수 있다.

4.2 공격 수준 기반 지능적 공격 차단 성능

공격자는 순차적 공격 수준 조정을 통해 시스템이 DoS 공격으로 판단하는 기준을 파악하고, 공격이 탐지되지 않는 수준에서 공격을 수행할 수 있다. 제안 시스템의 공격 수준 기반 지능적 공격 탐지 성능을 보이기 위해 공격 시나리오를 구성하고 시험을 수행하였다. 공격자는 SYN Flooding 공격을 수행하며, 순차적으로 단위 시간 당 패킷 발생 수를 증가시킨다. 실험에서는 hping3의 공격 파라미터 중 SYN 패킷의 크기를 조정하여 단위 시간 당 발생하는 SYN 패킷의 수를 조절하였다. 공격자는 8000바이트 크기의 공격으로 시작하여 8000바이트 단위로 공격 패킷의 크기를 늘려간다. 공격이 차단되었음을 확인하면 공격자는 이전에 성공한 공격 패킷 크기로 패킷 크기를 재설정한 후, 패킷 크기 증가폭을 2000바이트로 설정하여 다시 공격을 시도한다. 실험에 사용된 위협 레벨에 따른 SYN Flooding 공격 탐지 기준은 Table 5와 같다. SYN Flooding 공격의 경우 크기가 더 큰 패킷을 더 많이 전송할수록 희생자 시스템에 큰 부하를 일으킬 수 있다. 따라서 지능적 공격자가 순차적으로 공격 패킷의 크기와 수를 늘려갈 것으로 예상할 수 있다. 탐지 기준 비율을 높은 위협에서 더 낮게 설정함으로써 공격 발생 시 공격의 효과를 최소화할 수 있다. 위협 없음 단계에서부터 탐지 기준 비율을 낮게 설정할 수도 있으나 이 경우 정상 패킷이 공격 패킷으로 오탐지될 가능성이 높기 때문에 위협 없음 단계에서는 비교적 높은 50%를 기준으로 설정했다.

Fig. 12는 제안 시스템을 대상으로 공격 수준 기반 지능적 공격을 수행한 결과이다. 꺾은선 그래프는 공격자의 공격 패

Table 5. Thread Level - Detection Criteria

Thread Level	Detection Criteria
No Threat	50%
Low	40%
High	20%

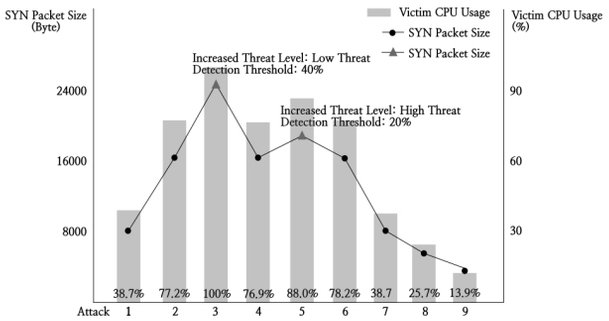


Fig. 12. Intelligent Attack Based on Attack Level in the Proposed System

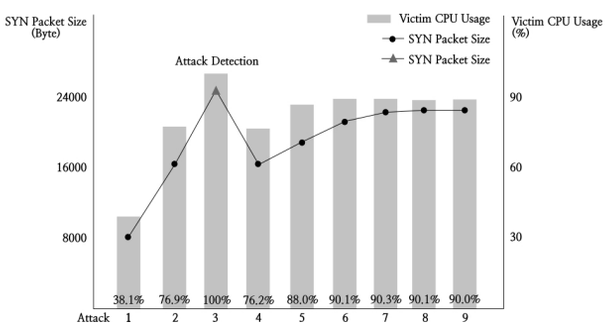


Fig. 13. Intelligent Attack Based on Attack Level in the System of Tian et al.[29]

킷 크기, 즉 공격 수준을 나타낸 것이며, 막대그래프는 각 공격 수행 시 피해자 시스템의 CPU 사용률을 나타낸 것이다. 공격 1~3까지 공격자는 8000바이트를 시작으로 공격 패킷의 크기를 증가시킨다. 공격 패킷 크기를 24000바이트로 설정했을 때, 전체 TCP 패킷 중 SYN 플래그가 달린 패킷의 비율이 50%를 넘어 공격으로 판단한다. 이 때 피해자 시스템의 CPU 사용률은 100%를 달성하여 정상적인 서비스가 불가능한 상태이다. 위협 레벨 관리 모듈은 위협 레벨을 낮은 위협으로 상향 조정하고, 공격 탐지 모듈은 이에 따라 탐지 기준을 40%로 조정한다. 공격 4에서 공격자는 24000바이트 크기의 공격 패킷이 탐지되었음을 확인하고 이전에 성공한 16000바이트 크기 패킷으로 공격을 수행한다. 공격 5에서 공격자는 패킷 크기 증가폭을 2000바이트로 줄여 18000바이트 크기 패킷으로 공격을 수행한다. 이 때, 하향 조정된 공격 탐지 기준에 따라 공격 탐지 모듈은 이를 공격으로 판단한다. 위협 레벨 관리 모듈은 위협 레벨을 높은 위협으로 상향 조정하고, 공격 탐지 모듈은 탐지 기준을 20%로 조정한다. 공격 6에서 공격자는 16000바이트 크기의 패킷으로 공격을 계속 시도하지만 하향 조정된 탐지 기준에 따라 이는 공격으로 판단되어 차단된다. 공격 7~9까지의 공격이 모두 공격으로 탐지됨에 따라 공격자는 공격 수준을 계속해서 하향 조정한다. 결과적으로 공격 패킷 크기 4000바이트 수준으로 공격 수준을 떨어뜨릴 수 있으며, 공격 9에서 피해자 시스템의 평균 CPU 사용량은 13.9%로 떨어졌다. 따라서 제안 시스템에서는 공격 수준 기반 지능적 공격을 차단했다고 볼 수 있다. 지능적 공격자가 공격을 실패하여 공격을 중지 했을 경우 허니팟에 시간 만료 타이머가 만료되는 순간 파라미터는 처음 단계인 위협 없음으로 되돌아간다.

Fig. 13은 Black-box[29]의 시스템에서 동일한 방법으로 공격 수준 기반 지능적 공격을 수행한 것이다. 해당 시스템은 공격 탐지 기준이 실시간으로 변동되지 않기 때문에 SYN 플래그가 달린 패킷 비율이 50%일 때 공격으로 판단하도록 설정하였다. 공격 1~3까지는 제안 시스템에서의 실험과 같은 양상을 보인다. 공격 3에서 시스템은 이를 공격으로 판단한다. 공격자는 이전에 성공한 16000바이트 크기 패킷으로 공격을 수행한다. 공격 5부터 공격자는 공격 패킷 크기의 증가폭을 줄여 이전에 차단된 24000바이트 이하까지 천천히 늘려나간다. Black-box 시스템의 경우 공격 탐지 기준이 변하지 않기 때문에 공격자는 탐지되지 않고 공격을 수행할 수 있다. 결과적으로 공격 9까지 수행하는 동안 공격자는 22000바이트 크기의 공격 패킷을 차단 없이 전송할 수 있다. 이 때 피해자 시스템의 평균 CPU 사용량은 90% 수준으로 정상적으로 서비스가 불가능한 상태이다. 따라서 해당 시스템에서는 공격 수준 기반 지능적 공격을 차단하지 못했다고 볼 수 있다.

5. 결 론

본 연구는 SDN과 허니팟의 특성을 활용하여 다양한 DoS 공격에 효과적으로 대응하는 방안을 제시하였다. 프로그램을 통해 트래픽을 제어하는 SDN의 유연성과 공격 패킷 분석에 적합한 허니팟의 장점을 기반으로 다양한 공격 수준 및 패턴을 지닌 지능적 DoS 공격에 대응할 수 있다. 이를 통하여 네트워크 기술 발달로 급격하게 방대해진 트래픽 양을 효과적으로 관리할 수 있고, 각 기관 및 기업에 막대한 피해를 발생시키는 다양한 지능적 DoS 공격을 효과적으로 차단할 수 있다. 특히 허니팟을 활용한 공격 분석을 통하여 시스템의 방어 방법을 파악하여 해당 방어 체계를 우회하는 방식으로 공격하는 지능형 공격자의 공격에 동적으로 대응할 수 있다는 것을 검증하였다.

본 연구에서 제안하는 시스템은 SDN 스위치가 수집한 트래픽을 분석하여 이를 기반으로 공격을 탐지하는 공격 탐지 모듈, DoS 공격의 발생 여부를 확인하여 위협 레벨을 관리함으로써 DoS 공격의 대응을 효과적으로 하는 위협 레벨 관리 모듈, 그리고 이러한 모듈들이 전송한 정보를 분석하여 동적인 대응 방법을 전송하는 허니팟을 프로그래밍하여 소켓 통신을 통한 유기적인 통신으로 지능형 공격에 효과적으로 대응하는 방법을 제시하였다. 제안 시스템에서 공격 패킷은 수정 가능한 플로우 규칙에 의해 허니팟으로 빠르게 전달될 수 있도록 하였고, 공격 패킷을 전달받은 허니팟은 이를 기반으로 지능적 공격의 패턴을 분석하도록 하였다. 분석 결과에 따라 지능적 공격에 대응할 수 있도록 공격 탐지 모듈과 위협 레벨 관리 모듈을 조정한다. 제안 시스템과 공격 패턴 및 공격 수준을 다양화한 지능적 공격을 직접 구현하여 실험하였

고, 위 방법과 기존의 DoS 공격 차단 방법의 하나인 Black-box와의 실험 결과 비교를 통해 제안 시스템의 공격 탐지율이 우위에 있음을 확인하였으며, 제안 시스템의 실현 가능성을 검증하였다.

항후 연구에서는 제안 시스템이 빅데이터를 기반으로 자율적으로 작동할 수 있도록 하기 위해 지식 기반 네트워킹(Knowledge Defined Network, KDN)과 SDN 컨트롤러의 분산기능을 도입하여 동영상 스트리밍, 대용량 파일 전송 등 DoS 공격으로 판단될 가능성이 있는 다양한 환경에서도 동적으로 적응하고 위협을 감지하며 이를 스스로 관리할 수 있는 시스템을 설계하고자 한다.

References

[1] K. Kirkpatrick, "Software-defined networking," *Communitcations of ACM*, Vol.56, No.9, pp.16-19, 2013.

[2] J. Choi, W. Park, and K. Kook, "Analysis of the advanced persistent threat (APT) - Targeting the Korean defense industry in 2009-2012," *Journal of the Korean Association of Defense Industry Studies*, Vol.19, No.2, pp.73-89, 2012.

[3] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Communications Surveys & Tutorials*, Vol.18, No.1, pp.602-622, 2015.

[4] N. Provos, "A virtual honeypot framework," in *USENIX Seurity Symposium*, Berkeley, CA: USENIX Association, pp.1-14, 2004.

[5] Open Networking Foundation, ONOS [Internet], https://opennetworking.org/onos.

[6] Linux Foundation, OpenvSwitch [Internet], https://www.openvswitch.org.

[7] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, and G. Parulkar, "ONOS: towards an open, distributed SDN OS," in *Proceedings of the Third Workshop on Hot Topics in Software Defined Networking*, New York: Association for Computing Machinery, pp.1-6, 2014.

[8] B. Pfaff, J. Petttit, T. Koponen, E. Jackson, A. Zhou, J. Rajahalme, and K. Amidon, "The design and implementation of open vswitch,", in *12th {USENIX} Symposium on Networked Systems Design and Implementation*, Santa Clara, CA: USENIX Association, pp.117-130, 2015.

[9] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexfor, and J. Turner, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM COmputer Communication Review*, Vol.38, No.2, pp.69-74, 2008.

[10] Wireshark Foundation, tshark [Internet], https://www.wireshark.org/docs/man-pages/tshark.html.

[11] Mininet Team, Mininet [Internet], http://mininet.org.

[12] K. Kaur, J. Singh, and N. S. Ghumman, "Mininet as software defined networking testing platform," in *International Conference on Communication, Computing & Systems*, Chennai, India: IEEE, pp.139-142, 2014.

[13] Salvatore Sanfilippo, Hping3 [Internet], http://www.hping.org.

[14] M. Agiwal, A. Roy, and N. Saxena, "Next generation 5G wireless networks: A comprehensive survey," *IEEE Communications Surveys & Tutorials*, Vol.18, No.3, pp.1617-1655, 2016.

[15] H. Kim and N. Feamster, "Improving network management with software defined networking," *IEEE COmmunicatinos Magazine*, Vol.51, No.2, pp.114-119, 2013.

[16] M. Casado, M. J. Feedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: Taking control of the enterprise," *ACM SIGCOMM Computer Communication Review*, Vol.34, No.4, pp.1-12, 2007.

[17] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, "Inferring internet denial-of-service activity," *ACM Transactions on Computer Systems*, Vol.24, No.2, pp.115- 139, 2006.

[18] S. T. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *IEEE Communications Surveys & Tutorials*, Vol.15, No.4, pp.2046-2069, 2013.

[19] T. Haq, J. Zhai, and V. K. Pidathala, "U.S. Patent No. 9,628,507," U.S. Patent and Trademark Office, 2017.

[20] Y. Choi, "Implementation of content-oriented networking architecture (CONA): a focus on DDoS countermeasure," in *Proceedings of European NetFPGA Developers Workshop*, Cambridge, UK: NetFPGA, 2010.

[21] X. You, Y. Feng, and K. Sakurai, "Packet In message based DDoS attack detection in SDN network using OpenFlow," in *2017 Fifth International Symposium on Computing and Networking*, Aomori, Japan: IEEE, pp. 522-528, 2017.

[22] T. Sanguankotchakorn and S. K. Arugonda, "Hybrid Controller for Securing SDN from Switched DDoS and ARP Poisoning Attacks," In *2019 20th Asia-Pacific Network Operations and Management Symposium*, Matsue, Japan: IEEE, pp.1-6, 2019.

[23] H. Wang and B. Wu, "SDN-based hybrid honeypot for attack capture," in *2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference*, Chengdu, China: IEEE, pp.1602-1606, 2019,

[24] X. Liu, H. Xue, X. Feng, and Y. Dai, "Design of the multi-level security network switch system which restricts covert channel," in *2011 IEEE 3rd International Conference on Communication Software and Networks*, Xi'an, China: IEEE, pp.233-237. 2011.

[25] T. Lotlikar and D. Shah, D. "A defense mechanism for DoS attacks in SDN (Software Defined Network)," in *2019 International Conference on Nascent Technologies in Engineering*, Maltepe, Turkey: IEEE, pp.1-7, 2019.

[26] Y. Kim, S. Ahn, N. C. Thang, D. Choi, and M. Park, "ARP poisoning attack detection based on ARP update state in software-defined networks," in *2019 International Conference on Information Networking*, Kuala Lumpur, Malaysia: IEEE, pp.366-371, 2019.

[27] C. Y. J. Chiang, Y. M. Gottlieb, S. J. Sugrim, R. Chadha, C. Serban, A. Poylisher, and J. Santos, "ACyDS: An adaptive cyber deception system," in *2016 IEEE Military Communications Conference*, Baltimore, MD: IEEE, pp.800-805, 2016.

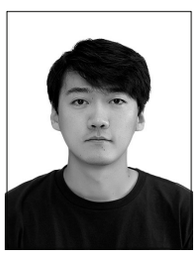
[28] Z. Zha, A. Wang, Y. Guo, D. Montgomery, and S. Chen, "Instrumenting open vSwitch with monitoring capabilities: designs and challenges," in *Proceedings of the Symposium on SDN Research*, New York: Association for Computing Machinery, pp.1-7, 2018.

[29] Y. Tian, V. Tran, and M. Kuerban, "DOS attack mitigation strategies on SDN controller," in *2019 IEEE 9th Annual Computing and Communication Workshop and Conference*, Nevada, LA: IEEE, pp.701-707, 2019.

[30] M. Kuerban, Y. Tian, O. Yang, Y. Jia, B. Huebert, and D. Poss, "FlowSec: DOS attack mitigation strategy on SDN controller," in 2016 IEEE International Conference on Networking, Architecture and Storage, Long Beach, CA: IEEE, pp.1-2, 2016.

윤 준 혁

https://orcid.org/0000-0001-6240-4455
e-mail : junhyeok@hknu.ac.kr
2021년 한경대학교 컴퓨터응용수학부(석사)
관심분야 : Nework Security, Data Security, Data Science, and Machine Learning



문 성 식

https://orcid.org/0000-0003-3477-8219
e-mail : door55ik@hknu.ac.kr
2020년 한경대학교 컴퓨터응용수학부(석사)
관심분야 : Network Security, Software Defined Networking

김 미 희

https://orcid.org/0000-0002-4896-7400
e-mail : mhkim@hknu.ac.kr
1997년 이화여자대학교 전자계산학과 (공학사)
1999년 이화여자대학교 컴퓨터학과 (공학석사)
1999년 ~ 2003년 한국전자통신연구원 연구원
2007년 이화여자대학교 컴퓨터학과(공학박사)
2007년 ~ 2009년 이화여자대학교 컴퓨터학과 전임강사
2009년 ~ 2010년 노스캐롤라이나주립대학교 연구원
2011년 ~ 현 재 한경대학교 컴퓨터응용수학부 교수
관심분야 : 네트워크 성능 분석 및 보안, 무선네트워크 보안, 침입대응, 크라우드센싱, 블록체인