DBpia

논문 17-42-08-13 The Journal of Korean Institute of Communications and Information Sciences '17-08 Vol.42 No.08
https://doi.org/10.7840/kics.2017.42.8.1581

Software Defined Networking을 위한 다중 기계학습 결합 기반의 DDoS 탐지 시스템

김 영 빈•, 최 동 호*, 판 반 트렁*, 마이 렁*, 박 민 호°

DDoS Detection System Based on Multiple Machine Learning Combination for Software Defined Networking

Young-pin Kim•, Dong-ho Choi*, Trung P.Van*, Mai Tieu Long*, Min-ho Park°

요 약

본 논문에서는 네트워크 DDoS 공격 트래픽에 분류 성능을 높이기 위한 다중기계학습 기반의 처리방법을 제안한다. 이 연구에서 제안된 결합 메커니즘은 두 가지 분류 알고리즘, Support Vector Machine (SVM), Decision Tree(DT)의 이점을 활용 하는 데에 초점을 둔다. SVM은 높은 정확도로 네트워크 flow를 분류하는데 적은 시간이 걸리고, DT는 미리 학습된 data mining기술로 더욱더 확실하게 flow를 예측 한다. 분산 서비스 거부 공격을 다루기 위해 SVM과 DT를 결합한 메커니즘을 제안하고, Software-Defined Networking에서 자원 고갈로부터 네트워크 컴포넌트를 보호한다. SVM은 첫 번째로 OpenFlow switch 들로부터 flow-tables에 존재하는 전체 flow를 분류한다. 분류된 flow가 SVM을 나타내는 그래프안의 공격flow 및 보통flow의 기준 선에서 공격인지 아닌지 확실하게 정의하기 힘든 희미하거나 애매한 부분이나 선의 가장자리 사이에 위치 하는 경우는 공격flow인지 보통flow인지 확인이 어렵기 때문에 최종 결정을 위하여 DT로 전달하여 더 확실히 공격 flow를 탐지한다. 그 후에 Attack Classifier와 Policy Enforcement 모듈은 공격 감소와 SDN controller 보호를 위해 flow에 적용되어 진다. 또한, Software-Defined Networking에서 분산 서비스 거부 공격의 새로운 관점에 대해 소개한다. 이 연구에서 Software-Defined Networking에서 실현가능한 실험을 해서 제안된 분류 결합 메커니즘이 기존 메커니즘보다 더 성능이 좋은 것을 실험으로 증명하였다. 새로운 SVM-DT 결합 메커니즘은 SDN controller와 OpenFlow switch들을 과 부화로 부터 보호하고 분산서비스 거부 공격에 대응하기 위한 효과적이고 획기적인 방법이다.

Key Words : Software-Defined Networking(SDN), Support Vector Machine(SVM), Decision Tree(DT), SDN controller, Openflow

ABSTRACT

In this paper, we introduce a multiple machine learning-based mechanism to increase the classification performance for network DDoS attack traffic. in this work, The proposed combination mechanism focuses on exploiting the advantages of two classification algorithms: Support Vector Machine (SVM), Decision Tree(DT). SVM takes less time to classify network flows with high accuracy and DT predicts flow more reliably with pre-learned data mining techniques. We propose a mechanism to combine SVM and DT to handle distributed

※ 본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 대학ICT연구센터육성 지원사업의 연구결과로 수행되었습니다.(IITP-2017-2012-0-00646)

• First Author : Department of ICMC convergence technology, Soongsil University, ypk@ssu.ac.kr, 학생회원
° Corresponding Author : School of Electronic Engineering, Soongsil University, mhp@ssu.ac.kr, 종신회원
* Co-Author : Department of ICMC convergence technology, Soongsil University, dhc@ssu.ac.kr, 학생회원, trungpv@ssu.ac.kr, longmaisg@ssu.ac.kr

논문번호 : KICS2017-05-135, Received May 1, 2017; Revised July 31, 2017; Accepted August 1, 2017

www.dbpia.co.kr

denial of service attacks and protect network components from resource depletion in Software-Defined Networking. SVM first classifies the entire flow in flow-tables from OpenFlow switches. It is difficult to determine whether the classified flow is an attack flow in the graph representing the SVM or an attack flow or an normal flow if it is located between the edge of the line or a blurred or obscure part that is difficult to define clearly whether it is an attack on the baseline of the flow The final decision is forwarded to DT to detect the attack flow more clearly. After that, the Attack Classifier and Policy Enforcement modules are applied to the flow for attack mitigation and SDN controller protection. In addition, we introduce a new viewpoint of distributed denial of service attack in Software-Defined Networking.In this work, experiments that can be realized in Software-Defined Networking have proven that the proposed classification combination mechanism has better performance than the existing mechanism. The new SVM-DT combination mechanism is effective and innovative way that protects the SDN controller and OpenFlow switches form overloading to response distributed denial of service attacks.

Ⅰ. 서 론

Software-Defined Networking(SDN)은 최근 유망한 차세대 네트워크 기술로 다양하게 연구되어지고 있고[1,2], 이 네트워크 모델에서 control plane과 data plane의 분리는 네트워크 모니터링과 제어를 위해 많은 이익을 준다. SDN의 중심이 되는 SDN cotroller는 OpenFlow switch를 통해 flow-based 트래픽을 제어하고 감시한다. SDN cotroller가 해당 flow의 경로를 계산 하여 설정한 후 OpenFlow switch에게 전송하여 OpenFlow switch는 Forwarding만 수행하기 때문에 많은 이점을 가져온다. 이런 이유로 이 네트워크 구조는 탐지 및 방지의 관점에서 공격을 위한 최적화 솔루션을 제공한다. 이 메커니즘을 위해 OpenFlow[3,4]는 controller 와 switch 사이에 첫 번째로 표준 통신 인터페이스로 정의되어졌다. OpenFlow는 packet-receive, send-packet-out, modify-fowarding-table, get-starts 등등과 같은 메시지를 SDN controller와 OpenFlow switch에게 제공한다는 것이 네트워크 관리에서 큰 이점인데 반해, 분산 서비스 거부 공격에 취약점이 될 수 있다. 분산 서비스 거부 공격의 기본 아이디어는 botnet들로부터 거대한 플로우를 발생시켜 피해 서버에게 전송하는 것인데 SDN 과 OpenFlow가 DDoS 공격에 취약한 이유는 OpenFlow switch가 보통 최대 백 만개의 flow를 유지시킬 수 있기 때문이다. 하지만 네트워크가 DDoS 공격을 받아서 수많은 flow가 OpenFlow switch에게 보내지면 서버나 목표 네트워크가 분산 서비스 거부 공격의 피해자가 될 뿐만 아니라 SDN cotroller나 OpenFlow switch도 자원의 고갈 때문에 작동을 멈추게 된다. 그렇기 때문에 정당한 유저나 flow는 고갈로부터 피해를 입은 전체 네트워크를 제대로 이용 할 수 없게 된다. Software-Defined Networking에서 분산 서비스 거부 공격을 해결하기에 큰 어려움을 야기한다. 기존 네트워크에서 DDoS의 영향을 최소화하기 위해 많은 방법[5-9]이 제안되어졌다. Software-Defined Networking 모델에서 AVANT-GUARD [10]은 연결 이동 툴 사용에 의한 bottleneck문제를 극복하기 위해 권고 되어졌고, FloodGuard는 네트워크 정책 시행을 지키고 SDN controller를 보호하기 위해 proactive flow rule analyzer와 packet migration 두 가지 모듈을 제안 했다. Fuzzy Logic의 사용은 SDN [11]과 Fonseca에서 flooding 공격 방어 메커니즘에 적용되어진다. ident++ protocol은 [12]에서 설명했고, 이는 SDN controller에 flooding 공격에 대한 효과적인 방법을 제공한다. Barga [13]은 Self-Organizing Map을 사용한 분산 서비스 거부 공격 메커니즘을 제안했다. [14]에서 DDoS Blocking Scheme는 표준 OpenFlow 인터페이스를 사용한 Botnet기반 공격을 다루는 것을 소개했다. 이러한 메커니즘들은 새로운 방법이고 분산 서비스 거부 공격과 네트워크 보호라는 같은 목표를 가지고 있다. 그리고 [15]에서는 SDN환경에서 Machine Learning 알고리즘 SVM과SOM을 결합하여 DDoS 공격을 방어하는 메커니즘을 소개했다. 하지만 본 논문에서는 이전의 방법 보다 더 정확하고 빠른 새로운 Machine Learning 알고리즘인 Decision Tree(DT)와 Support Vector Machine(SVM)의 결합을 사용하여 SDN환경에서 DDoS 공격을 방어하는 메커니즘을 제안한다.

이 연구에서, 지금까지의 연구보다 빠르고 정확하게 처리가 가능한 SVM-DT 결합 DDoS 공격 처리 메커니즘을 소개한다. 첫 번째로 네트워크트래픽을 위

www.dbpia.co.kr

해 flow 분류의 성능을 높이기 위해 Support Vector Machine(SVM) [16]과 Decision Tree(DT) 결합된 사용법을 제안한다. 여기서 SVM 은 패턴을 알아보고 데이터를 분석하기 위한 지도 학습 모델이고 Decision Tree(DT)은 SVM을 나타내는 그래프안의 공격flow 및 보통flow의 기준 선에서 공격인지 아닌지 확실하게 정의하기 힘들어서 처리되어지지 않은 희미하거나 애매한 네트워크 flow를 가장 효율적으로 분류하기위한 방법 중 하나로 사용된다. SVM과DT 결합은 더 정확한 결과를 생성 할 뿐만 아니라, 처리시간을 감소시키는 SVM과 DT의 장점을 모두 가져온다. 두 번째로 SDN network component의 자원 고갈과 DDoS 공격 효과를 줄이기 위해 제안된 메커니즘 SVM-DT 결합을 지원한다.

본 연구에서 제안하는 메커니즘에 대한 주요 관련된 연구는 아래와 같다.

- 네트워크 트래픽 분류 성능을 높이기 위한 Decision Tree와 Support Vector Machine의 사용에 대한 연구
- SDN 환경에서 DDoS공격의 새로운 관점을 발표 하고, 평범한 네트워크에서 DDoS 공격의 일반적인 종류를 조사
- DDoS 공격 효과를 줄이기 위한 SVM-DT 결합 메커니즘을 제안하고, 과부화로부터 SDN controller와 OpenFlow switch를 방어
- 실험은 POX controller를 사용한 SDN 환경에서 시행하였다. 그러나 POX controller이외에 다른 controller에도 쉽게 적용될 수 있다. 그리고 CAIDA dataset[17,18]은 SVM과 DT 학습을 위해 사용되어진다.

본 논문의 나머지 부분은 다음과 같이 구성되어져 있다.

관련연구 에서는 Software-Defined Networking에서 새로운 SVM과 DT에 대해 간략하게 설명하고, DDoS 공격 종류를 분류한다. 그리고 이 연구에서 제안된 SVM-DT결합 DDoS 공격 처리 방법을 소개한다. 실험에서는 실험을 설명하고 결과와 성능 평가를 한다. 마지막으로 결론과 향후 연구를 발표한다.

Ⅱ. 관련연구

관련연구에서는 선으로 된 Support Vector Machine 과 Decision Tree(DT)에 대해 간단하게 설명한다. [19-21]과 [22],[23]에서 SVM과 DT 알고리즘에 대한 더 자세한 정보를 참조 할 수 있다. 그리고 본 논문에서 소개하는 Software-Defined Networking에서 DDoS 공격의 종류를 소개한다.

2.1 SVM과 DT

2.1.1 Decision Tree(DT)

Decision Tree(DT)[24]는 Data Mining에서 일반적으로 사용되는 방법론으로, 몇몇 입력 변수를 바탕으로 목표 변수의 값을 예측하는 모델을 생성하는 것을 목표로 한다. 트리 구조에서, 각 내부 노드들은 하나의 입력 변수에, 자녀 노드들로 이어지는 가지들은 입력 변수의 가능한 값에 대응된다. 잎 노드는 각 입력 변수들이 루트 노드로부터 잎 노드로 이어지는 경로에 해당되는 값들을 가질 때의 목표 변수 값에 해당된다. DT를 구성하는 알고리즘에는 주로 하향식 기법이 사용되며, 각 진행 단계에서는 주어진 데이터 집합을 가장 적합한 기준으로 분할하는 변수 값이 선택된다. 서로 다른 알고리즘들은 분할의 적합성을 측정하는 각자의 기준이 있고 이러한 기준들은 보통 부분 집합 안에서의 목표 변수의 동질성을 측정하며, 이 기준들은 가능한 데이터 집합 분할의 경우의 수마다 적용되며, 그 결과 값들은 병합되어, 즉 평균값이 계산되어, 데이터 집합의 분할이 얼마나 적합한지 측정하는데 사용된다.

본 논문에서 DT를 사용하는 이유는 결과 해석이 용이하고 자료를 가공할 필요가 거의 없으며 안정적이여서 사용되는 명제가 다소 손상이 되더라도 동작이 잘되고, 방대한 분량의 data set에서도 잘 동작하고 빠른 시간 내에 분석이 가능하기 때문이다.

2.1.2 SVM(Support Vector Machine)

Support Vector Machine[25]은 고정되어 있지만 알려지지 않은 확률 분포를 갖는 데이터에 대해 잘못 분류하는 확률을 최소화하는 구조적인 위험 최소화방법에 기초하고 있다. 또한 패턴을 고차원 특징 공간으로 사상시킬 수 있다는 점과 대역적으로 최적의 식별이 가능한 특징을 가지고 있다. SVM은 입력 공간에 있는 분류 데이터에서 Margin 값을 최대로 하는 초평면을 찾아내어 이진 분류를 한다.[26]

본 논문에서 SVM을 사용하는 이유는 결과 해석이 용이하고, 실제 응용에 있어서 빠르고 높은 성과를 내고, 많지 않은 학습 자료만으로 신속하게 분별학습을 수행할 수 있기 때문이다.

2.2 Software-Defined Networking에서 DDoS 공격

먼저 기존 네트워크에서 DDoS 공격을 간단히 설명한다. 본 논문에서는 bandwidth 고갈 공격과 자원

www.dbpia.co.kr

고갈 공격 두 가지 주된 종류로 요약한다. bandwidth 고갈 공격에서 공격자는 피해자 네트워크의 bandwidth를 고갈시키는 원하지 않은 트래픽을 피해자에게 보내는 공격이다. 이 공격은 정상 트래픽이 피해 네트워크를 접근할 수 없게 만든다. 예를 들어 UDP flooding, ICMP flooding 또는 Smurf 나 Fraggle 공격들이 bandwidth 고갈 공격이다. 자원 고갈 공격은 공격자가 변조된 IP 패킷을 보내거나 악용 네트워크 protocol을 보내는 것을 목표로 한다. 그 결과, 피해서버는 자원 고갈로부터 피해를 입고 연결 용량이 충분할 때, 피해서버는 작업을 할 수 없다. TCP SYN flooding이 이와 관련된 공격인데, TCP 연결 시작 전에 송신자와 수신자사이에 three-handshake protocol기반 시에 사용되는 공격으로 이 공격에 좋은 예제 이다. flow-based 네트워크 모델인 Software-Defined Networking의 측면으로부터 두 가지 주된 DDoS 공격 종류를 분류한다.

2.2.1 Type 1

주된 아이디어는 출발지 주소로부터 packet이나 data 수신은 용량에 의존한다. 네트워크 시스템이 이 DDoS 공격에 공격받을 때, 중요한 특징은 출발지 IP 주소가 각 flow에서 packet의 용량을 상위레벨에서 한 개나 두 개 의 flow발생에 의해 피해 네트워크에 연결되는 것이다. 예를 들어 ICMP flooding 공격, Smurf 와 Fraggle 공격이 이러한 공격이다. 이러한 종류의 공격이 SDN 환경에서 발생되어진다면, 공격당하고 있는 링크의 과부하가 발생하게 되고 이는 정상 사용자의 flow가 해당 링크를 지나갈 수 없게 된다.

2.2.2 Type 2

두 번째 타입은 피해 네트워크 시스템을 무너뜨리기 위한 flow의 수와 용량 기반 공격이다. 기본 아이디어는 짧은 시간에 공격자는 피해 주소에 큰 수에 flow를 발생시키는 것이다. 좋은 예제는 TCP SYN flooding 공격인데 변조된 IP 주소를 가진 공격자는 목표 Web 서버에 수 천개의 request를 보낸다. 이 공격은 피해 서버를 만들 뿐 아니라, OpenFlow switch 나 SDN controller와 같은 네트워크 장치에도 피해를 준다. 이러한 종류 공격이 SDN 환경에서 발생되어진다면, SDN Switch는 이 공격에 의한 flow처리를 위해 대규모 Packet_in message를 보내게 될 것이고 그렇게 된다면 SDN Controller 또한 대규모 Packet_in message를 처리를 위해 자원을 소모해야하고 이는 SDN Controller의 자원 고갈과 과부하로 이어지게 된다. 만약 SDN Controller가 공격에 의한 flow를 모두 처리 하였다 하더라도 SDN Switch의 flow table에 많은 량의 대규모 flow rule을 처리해야하기 때문에 자원 고갈과 과부하가 발생한다.

표 1. Software-Defined Networking에서의 DDoS 공격들
Table 1. DDoS Attacks In Software-Defined Networking.

Type 1	Type 2
ICMP flooding attack	TCP SYN attack
IGMP flooding attack	UDP flooding attack
Fraggle attacks	PUSH + ACK attack
Smurf attacks	Malformed Packet attack

결론적으로, Software-Defined Networking 환경에서 새로운 관점의 포인트와 분석으로부터, 이러한 일반적인 DDoS 공격은 Table I 가 보여주는 것처럼 요약되어질 수 있다.

Ⅲ. 시스템 구조 및 알고리즘

3.1 Software-Defined Networking에서 DDoS공격을 처리할 SVM-DT 결합 메커니즘

다음은 본 논문에서 제안된 메커니즘의 자세한 설명이다. 그림 2는 Software-Defined Networking구조에서 SVM-DT 결합 메커니즘 개요를 설명하는데 이 구조는 control plane이 확장된 것이다. 이 연구에서 메커니즘은 SDN controller에 위치하고, 8개의 분리된 모듈로 구성된다. Flow Collector, Feature Extractor, Traffic Classifier, SVM-i, Traing Database, DT, Attack Classifier, Policy Enforcement.

이 연구에서 제안되어진 모듈이 실행되기 전에 SVM-i 와 DT은 학습 Database로부터 준비된 dataset에 의해 학습되어진다. 첫 번째 단계에서, Flow Collector는 미리 결정된 시간 간격에 OpenFlow switch로부터 flow 정보를 수집한다. 그 후, 이 정보는 flow의 속성을 추출하기 위해 Feature Extractor에게 전송된다. 그 다음, Traffic Classifier 모듈은 flow의 traffic 타입을 분류하고 다음 과정에서 상응하는 SVM-i에 전달한다. 각 SVM-i에서 영역 결정은 SVM-i에서 flow의 위치에 기반 하여 주어질 것이다. 만약 flow의 위치가 공격 영역에 위치된다면, 즉시 Attack Classifier에게 보내질 것이다. 그렇지 않으면 SVM을 나타내는 그래프안의 공격flow 및 보통flow의 기준 선에서 공격인지 아닌지 확실하게 정의하기 힘든 flow가 애매한 영역에 속하는지를 검사할 것이

www.dbpia.co.kr

다. 만약 flow의 영역이 공격인지 아닌지 확실하게 정의하기 힘든 애매한 부분에 있다면, DT는 flow의 속성을 더 가져올 것이고, 예측 한다. 그리고 그 flow는 다른 경우에서 보통 flow로 결론을 내린다. DT 모듈은 입력 속성을 사용하여 결정을 내린다. 이 경우에, flow는 공격 flow로 간주하고, Attack Classifier에게 전송된다. Classifier에서 공격 flow는 protocol기반 DDoS 공격의 두 가지 타입으로 분류 되어 진다. 마지막 절차는 공격의 두 가지 타입에 대한 정책을 만들고 공격 축소를 위한 rule을 OpenFlow switch에게 보내는 Policy Enforcement 모듈이다. 해당 flow가 정상 flow라면 flow-table에 유지된 rule에 아무것도 하지 않는다. 이러한 과정들이 공격 flow를 탐지하기 위해 사용되고, 시간 간격동안에 더 이상의 flow 정보가 없을 때 까지 반복한다.

이 메커니즘에서, 학습 Database는 위의 과정으로부터 모아진 flow의 속성을 계속해서 업데이트 한다. 네트워크 관리자에 의해 정의되어진 미리 설정된 시간에, SVM-i와 DT는 업데이트된 Database을 사용해서 학습되어진다. 이렇게 함으로써, 제안된 메커니즘은 다양한 네트워크 시스템에 유연하게 조정 할 수 있다. 아래에 세부항목에서 제안된 메커니즘의 모듈을 더 정확하게 설명한다.

3.1.1 Flow Collector

이 모듈은 SDN controller에서 실행되는 간단한 모듈이다. Flow Collector는 미리 설정된 시간에 OpenFlow swtich에게 StartsRequest 메시지를 내보내고 StartsResponse 메시지를 받는다. [4], [27]

3.1.2 Feature Extractor

Feature Extractor는 flow의 네 가지 속성을 가져오는 StartsResponse 메시지로부터 flow 정보를 추출한다. 그 중 두 가지 속성은 SVM-i에 입력하고, DT에서 네 가지 모든 속성을 처리할 것이다. 추출된 속성을 Traffic Classifier로 보낸다.

3.1.3 Traffic Classifier

Feature Extractor로부터 받은 속성을 바탕으로 Traffic 속성을 분류한다. 이 모듈은 상응하는 SVM-i에 flow의 속성을 전송을 책임진다. 예를 들어 protocol 필드 ICMP의 flow에서 flow 정보는 분류를 위해 SVM-ICMP에 보내질 것이다.

3.1.4 SVM-i Module

Traffic Classifier로부터 받은 정보를 바탕으로 SVM-i 모듈을 실행한다. SVM-i 모듈은 ICMP, TCP, UDP와 같은 네트워크 traffic의 각 종류별로 분류자로 정의되어 진다. 그러므로, 이 메커니즘에서는 네트워크 traffic을 분류하기 위한 다중 SVM을 사용한다. SVM-i는 정상과 비정상 샘플 데이터를 둘 다 학습하는 것을 포함한다. 학습 절차가 완료되어진 후에, SVM-i는 데이터 분산 그래프를 만들고 hyperplane을 정의한다.

(단 i는 프로토콜을 의미한다. 예를 들어 SVM-TCP, SVM-ICMP)

3.1.5 DT Module

DT모듈은 만약 SVM에서 분류하지 못한 flow를 더 정확하게 분류하기 위한 모듈이다. DT는 훈련용 데이터를 이용하여 독립변수의 차원공간을 반복적으로 분할하고 평가용 데이터를 이용하여 오 분류율을 크게 할 위험이 높거나 부적절한 규칙을 가지고 있는 분석데이터를 제거한 뒤 해당 결과에 근거한 의심스러운 패턴에 대한 분류를 정의한다. flow의 입력 벡터는 네 가지 속성을 포함하는 4개 튜플(packet의 수, byte의 수, 기간, protocol)에 의해 명시되어진다. DT 출력은 의심스러운 flow의 최종 분류 결과를 나타낸다.

3.1.6 Attack Classifier

그림 2 에서 보여지는 것 처럼, Attack Classifier는 SVM-i 와 DT 모듈로부터 공격 flow의 정보를 받는다. 그 다음, protocol 기반 두 개 타입으로 비정상 flow를 분류한다.

3.1.7 Policy Enforcement

DDoS 공격의 안 좋은 효과를 줄이고 막기 위해, 각 공격 타입을 위한 다양한 방어 기술을 사용한다. client로 부터 하나 또는 두 개의 flow를 생성하려고 하지만 패킷의 막대한 수를 전송하려는 공격(예를들어 ICMP Flooding) Type 1과 같은 flow를 위한 drop action을 사용한다. 피해 서버에 막대한 flow를 발생시키는 Type 2 공격(TCP SYN Flooding)과 관련하여, 분류 절차가 끝난 후에, 비정상 flow는 SDN controller에 의해 flow-table로부터 제거되어진다.

3.1.8 Training Database

Training Database 는 SVM-i 와 DT 학습을 위한 학습 샘플을 저장한다. 먼저, 입력 샘플은 준비된

www.dbpia.co.kr

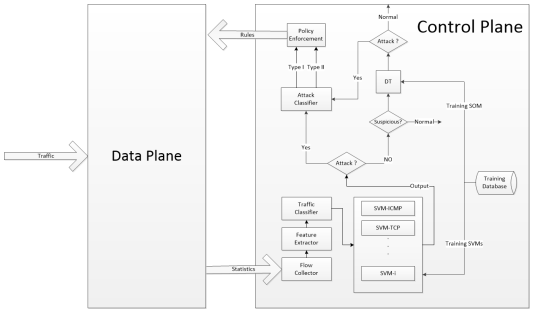


그림 1. 제안된 SDN에서 DDoS탐지를 위한 SVM-DT 메커니즘
Fig. 1. The proposed SVM-DT mechanism to detect DDoS attack in SDN.

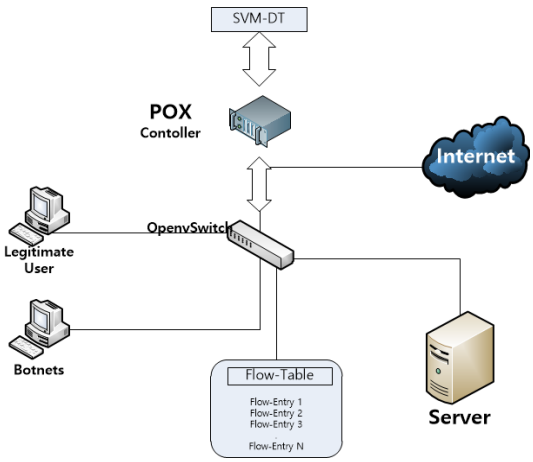


그림 2. SVM-DT 구축 Topology
Fig. 2. The SVMs-DT Implementation Topology.

dataset으로부터 발생되어지지만, 네트워크 시스템은 메커니즘이 실행되는 동안 입력 샘플을 업데이트 할 수 있다. 그러므로 Training Database는 항상 최신버전을 유지한다.

Ⅳ. 실험 및 환경 구축

Software-Defined Networking에서 DDoS 공격을 대처하기 위한 SVM-DT를 결합한 flow기반 메커니즘의 자세한 구현을 설명한다. 그 중에서 분류 성능을 비교하기 위해 SVM과 DT 기본 알고리즘 또한 구현한다. 그림 2는 SDN controller (POX), OpenvSwitch, web server, 정상 유저, botnet, 인터넷 연결로 구성된 topology를 테스트하는 것을 보여준다.

실험은 SVM과 DT을 학습시키기 위해 보통 dataset과 공격 dataset을 사용한다. 게다가, DDoS 공격 툴은 비정상적인 flow를 발생시키는데 사용하고 정상 flow는 인터넷상의 실제 웹 사이트에 접근에 의해 만들어진다.

4.1 CAIDA Dataset

CAIDA Dataset [28],[29]의 학습 샘플을 사용한다. CAIDA는 Web, FTP, Ping과 같은 다양한 실제 네트워크 타입을 수집하는 신뢰할 수 있는 dataset 중에 하나이다. 정상적인 dataset에서 TCP protocol packet은 89%를 발생하고, 단지 6%만 ICMP packet이고, 다른 protocol은 오직 5%만 만든다. 비정상 dataset에서는 6% 미만의 작은 비율이 TCP packet이고 대다수의 93%의 ICMP packet이 DDoS 공격에 사용되는 주요 protocol이 되고, 다른 protocol과 단 1%만 일치한다. dataset에서, TCP와 ICMP protocol이 가장 높은 비율을 차지한다. 따라서 이 실험에서는, TCP와 ICMP protocol을 사용하는 DDoS 공격에 초점을 둔다. ICMP flooding은 Type 1 이고, TCP SYN flooding은 Type 2이기 때문에 본 논문에서 제안하는 메커니즘이 Software-Defined Networking 환경에서 DDoS 공격 type을 확인 할 수 있다.

4.2 DDoS Attack Tool

실험에서는 BoNeSi[30] 라는 이름의 DDoS공격 툴을 사용한다. 이 툴은 botnet을 사용하여 DDoS 공격을 시뮬레이션 할 수 있고, 정의된 botnet으로부터 목표 네트워크에 TCP, ICMP, UDP packet을 발생시킬 수 있다. 제안된 메커니즘의 구현을 위해 사용한 BoNeSi 툴은 실제 DDoS공격 flow를 시뮬레이션 할 수 있는 강력한 tool로 평가 되어졌다.

4.3 Testing Process

test를 실행하기 전에, Table 2에 나와 있는 것처럼 DT와 SVM-TCP, SVM-ICMP,과 기존에 제안되어졌던 메커니즘인 DT와 SVM-DT의 결합 방법의 학습을 위한 dataset을 정교하게 준비했다. 다섯 가지 경우 모두에서, 각 SVM과 DT은 CAIDA dataset으로부터 추출되어진 flow 4000개를 학습했다. POX controller는 기본적으로 flow의 hard timeout 시간 값을 최대 30초로 설정하기 때문에 실제 SVM 평면에서 수평 축은 시간을 나타내고 0에서 30사이 이고, 수직 축은 패킷의 수 이고 무한대 이다. 왜냐하면 수직축은 flow-entry의 packet 수를 나타내는데 이는 계속 증가할 수 있기 때문이다. ICMP protocol에 준수에 관련하여, 일반 ICMP 사용자는 해당 네트워크에 많지 않은 패킷을 가끔 보내는 경우가 있지만, 공격 ICMP 사

www.dbpia.co.kr

용자는 오랜 시간 동안 자주 많은 수의 패킷을 피해 네트워크에 보낸다. HTTP와 FTP를 포함하는 TCP protocol의 조건에서, 일반 유저와 서버는 보통 각 섹션에 대해 방대한 패킷을 교환한다. 하지만, TCP Flooding 공격에서 좀비들은 단지 1~2개의 패킷으로 flow나 섹션을 연다. 테스트 절차는 두 개의 시나리오로 시작한다. 첫 번째는 System이 보통 flow만 포함하고, 두 번째 시나리오에서는 System이 ICMP flooding 과 TCP SYN flooding에 공격을 받는다. 첫 번째 시나리오에서 정당한 유저는 20000개 flow가 발생되는 인터넷에 있는 다양한 실제 웹 사이트에 접근한다. 그 동안에, 공격 시나리오에서, 악의적인 사용자에 의해 설치된 BoNeSi tool은 20000개의 비정상 flow를 만들어서 웹 사이트에 공격을 시행한다. 두 시나리오는 Table 2처럼 3가지 경우를 모두 반복한다.

표 2. 학습 flow 수와 Test flow
Table 2. Number of Training and Testing Flows.

Algorithm	Training	Testing
SOM	4000	40000
SVM	4000:4000	40000
SVM-SOM	4000:4000:4000	40000
DT	4000	40000
SVM-DT	4000:4000:4000	40000

V. 성능 평가

5.1 탐지 비율, 정확도, FAR 향상

세 가지 실험을 했고, 4개의 parameter를 계산했다. True Positive(TP)는 불법적인 flow으로 분류된 flow가 비정상 flow일 확률이고, True Negative(TN)는 신뢰할 수 있는 정상 flow로 분류된 flow가 정상 flow일 확률이고, False Positive (FP)는 불법적인 flow로 분류된 flow가 정상 flow일 확률이고, False Negative(FN)는 신뢰할 수 있는 정상 flow로 분류된 flow가 공격 flow일 확률 이다. Table 3은 모든 parameter의 값을 나타낸다.

표 3. 실험 결과
Table 3. Experiment Results.

Algorithm	TP (%)	TN (%)	FP (%)	FN (%)
SOM	93.49	94.24	6.51	5.76
SVM	92.33	93.45	7.67	6.55
SVM-SOM	96.03	98.17	3.97	1.83
DT	96.81	95.55	3.19	4.45
SVM-DT	98.41	97.81	1.59	2.19

위 Table 3에서 나타난 결과로, 기존 SOM과 SVM과 DT 그리고 SOM-SVM과 비교하여 SVM-DT 분류 성능을 평가하기 위해 결정적인 기준(탐지 비율, 정확도, FAR)을 계산한다. 그림 3은 다섯 가지 메커니즘 비율의 비교를 보여준다. 탐지 비율과 정확도면에서, 제안된 방법 SVM-DT은 가장 높은 비율을 차지할 때, 98.57% 및 98.11%로 기존 SVM, SOM, DT 그리고 SVM-SOM보다 더 나은 결과를 보였다. 반면에, 단일 SOM은 94,0%인데 탐지율과 정확도 면에서 다중 SVM 보다 대략 1% 우수하고, 오탐율에서, SVM-DT는 잘못된 경고 발생에서 2.19%만 발생시키므로 기존 방법 중에 가장 우수하다. 반면에 기존 SOM과 SVM에 의해 발생된 잘못된 경보율은 7.0%이고, SVM-SOM의 경우에는 3.84%로 SVM-DT가 약 2% 우수하다. 결론적으로, SVM-DT 결합 메커니즘이 모든 평가 기준에서 기존 방법보다 더 효과적이다.

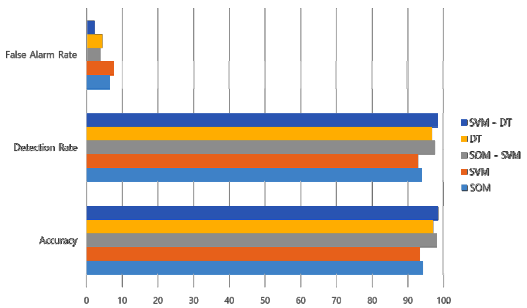


그림 3. 탐지율, 정확도, FAR 비교
Fig. 3. The Detection Rate, Accuracy and FAR comparison.

요약하자면 본 논문에서 제안하는 SVM과DT을 결합한 메커니즘 DDoS 공격을 받을 때 정확도와 탐지율 그리고 FAR모두 기존의 방법보다 우수하다.

VI. 결 론

본 논문에서 Software-Defined Networking에서 DDoS 공격을 다루기위해 SVM-DT 결합을 사용하는 메커니즘을 제안했다. 이 연구에서 제안한 기술은 이전 연구와 다르게 기계학습을 활용하고 결합한 후 사용하여, 기존의 방법과는 다르게 Machin Learning을 사용하여 더 정확하게 탐지가 가능하고, 각 Machin Learning의 이점을 활용하여 더 좋은 성능을 가져오

www.dbpia.co.kr

는데 있어서 획기적이다. 본 논문에서 제안한 기계학습 기반 DDoS 탐지 시스템은 세 가지 주요 기준 탐지율, 정확도, False Alarm Rate 에서 기존 방법보다 더 좋은 결과를 낸다. 향후 연구에 관해서는 더 많은 유형의 공격을 지원하기 위해 SVM-DT를 확장 할 계획이다.

References

[1] G. Lee, I. Jang, W. Kim, S. Joo, M. Kim, S. Pack, and C.-H. Kang, "SDN-Based middlebox management framework in integrated wired and wireless networks," *J. KICS*, vol. 39, no. 6, pp. 379-386, Jun. 2014.

[2] Y. Kyung, K. Hong, S. Park, and J. Park, "Load distribution method over multiple controllers in SDN," *J. KICS*, vol. 40, no. 6, pp. 1114-1116, Jun. 2015.

[3] P. Goransson and C. Black, *Chuck Black: Software Defined Networks: A Comprehensive Approach*, Elsevier, 2014.

[4] The Open Networking Foundation, *OpenFlow Switch Specification Version 1.4.0*, Retrieved 2016 [Online] from https://www.opennetworking.org

[5] S. Zargar, J. Joshi, and D. Tipper, "A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks," *IEEE Commun. Surv. Tuts.*, vol. 15, no. 4, pp. 2046-2069, Nov. 2013.

[6] J. Mirkovic and P. Reiher, "A taxonomy of DDoS attack and DDoS defense mechanisms," *ACM SIGCOMM Comp. Commun. Rev.*, vol. 34, no. 2, pp. 39-53, Apr. 2004.

[7] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DoS and DDoS problems," *ACM Comput. Surv.*, vol. 39, no. 1, pp. 1-42, Apr. 2007.

[8] R. K. C. Chang, "Defending against flooding-based distributed denial of service attacks," *A Tutorial, Comput. J. IEEE Commun. Mag.*, vol. 40, no. 10, pp. 42-51, 2002.

[9] D. H. Shin, K. K. An, S. C. Choi, and H.-K. Choi, "Malicious traffic detection using K-means," *J. KICS*, vol. 41, no 2, pp. 277-284, Feb. 2016.

[10] S. Shin, V. Yegneswaran, P. Porras, and G. Gu, "AVANT-GUARD: scalable and vigilant switch flow management in software-defined networks," in *Proc. ACM SIGSAC Conf. Com. Commun. Sec.*, pp. 413-424, 2013.

[11] P. Van Trung, et al., "A Multi-criteria-based DDoS attack prevention solution using software defined networking," in *IEEE Int. Conf. Advan. Tech. for Commun.*, pp. 308-313, 2015.

[12] J. Naous, R. Stutsman, D. Mazieres, N. McKeown, and N. Zeldovich, "Delegating network security with more information," in *Proc. 1st ACM workshop*, pp. 19-26, NY, USA, 2009.

[13] R. Braga, E. Mota, and A. Passito, "Lightweight DDoS flooding attack detection using NOX/OpenFlow," in *35th IEEE Conf. Lo. Com. Net.*, pp. 408-415, Denver, Colorado, 2010.

[14] S. Lim, J. Ha, H. Kim, Y. Kim, and S. Yang, "A SDN-Oriented DDoS blocking scheme for botnet-based attacks," in *ICUFN*, pp. 63-68, China, 2014.

[15] T. V. Phan, N. K. Bao, and M. Park, "A novel hybrid flow-based handler with DDoS attacks in software-defined networking," in *Proc. 13th IEEE Int. Conf. Advanced and Trusted Comput.*, Toulouse, France, Jul. 2016.

[16] J. Shawe-Taylor and N. Cristianini, *Support Vector Machines and other kernel-based learning methods,* Cambridge Univ. Press, UK, 2000.

[17] CAIDA Datasets, *Anonymized Internet Traces 2015*, Retrieved 2016 [Online]. from https://data.caida.org/datasets/passive-2015/

[18] CAIDA Datasets, *DDoS Attack 2007*, Retrieved 2016 [Online]. from https://data.caida.org/datasets/security/ddos-20070804/

[19] J. Shawe-Taylor and N. Cristianini, *Support Vector Machines and other kernel-based learning methods*, Cambridge Univ. Press, UK, 2000.

www.dbpia.co.kr

[20] D. J. Sebald and J. A. Bucklew, “Support vector machine techniques for nonlinear equalization,” *IEEE Trans. Sign. Process.*, vol. 48, no. 11, pp. 3217-3226, Nov. 2000.

[21] K. Lin and C. Lin, “A study on reduced support vector machines,” *IEEE Trans. Neu. Net.*, vol. 14, no. 6, Nov. 2003.

[22] S. R. Safavian and D. Landgrebe, “A survey of decision tree classifier methodology,” *IEEE Trans. Systems, Man, and Cybernetics*, vol. 21, no. 3, pp. 660-674, 1991.

[23] M. A. Friedl and C. E. Brodley, “Decision tree classification of land cover from remotely sensed data,” *Remote Sensing of Environ.*, vol. 61, no. 3, pp. 399-409, Sept. 1997.

[24] L. Rokach and O. Maimon, “Top-down induction of decision trees classifiers-a survey,” *IEEE Trans. Systems, Man, and Cybernetics, Part C (Appl. and Rev.)*, vol. 35, no. 4, pp. 476-487, Nov. 2005.

[25] E. Osuna, R. Freund, and F. Girosi, *Support vector machine : Training and applications*, MIT Artificial Intelligence Laboratory and Center for Biological and Computational Learning, 1997.

[26] P. S. Jun, I. W. Kang, J. E. Choi, J. W. Jeong, K. H. Kim, and S. R. Lee, “A method for efficient energy of selecting the cluster-head using SVM algorithm in wireless sensor networks,” in *Proc. KICS ICC*, pp. 21-22, Korea, Jun. 2011.

[27] Message Layer Definition, *OpenFlow Messages* Retrieved 2016 [Online]. from http://flowgrammable.org/sdn/openflow/message-layer/

[28] CAIDA Datasets, *Anonymized Internet Traces 2015*, Retrieved 2016 [Online]. from https://data.caida.org/datasets/passive-2015/

[29] CAIDA Datasets, *DDoS Attack 2007,* Retrieved 2016 [Online]. from https://data.caida.org/datasets/security/ddos-20070804/

[30] BoNeSi, *The DDoS Botnet Simulator*, Retrieved 2016 [Online]. from https://github.com/markus-go/bonesi

김 영 빈 (Young-pin Kim)

2016년 7월 : 숭실대학교 컴퓨터공학과 졸업

2016년 9월~현재 : 숭실대학교 정보통신소재융합학과 석사과정

<관심분야> 컴퓨터공학, 통신공학

최 동 호 (Dong-ho Choi)

2016년 2월 : 청운대학교 컴퓨터공학과 졸업

2016년 9월~현재 : 숭실대학교 정보통신소재융합학과 석사과정

<관심분야> 컴퓨터공학, 통신공학

판 반 트렁 (Trung P.Van)

2015년 7월 : Hanoi University of Science and Technology School of Electronics and Telecommunications 졸업

2017년 7월 : 숭실대학교 정보통신소재융합학과 석사 졸업

<관심분야> 컴퓨터공학, 통신공학

마이 렁 (Mai Tieu Long)

2012년 2월 : Hochiminh City University of Technology Department of Electronic and Telecommunication 졸업

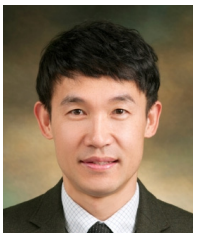
2017년 7월 : 숭실대학교 정보통신소재융합학과 석사 졸업

2017년 9월~현재 : 숭실대학교 정보통신소재융합학과 박사과정

<관심분야> 컴퓨터공학, 통신공학

www.dbpia.co.kr

박 민 호 (Min-ho Park)



2000년 2월 : 고려대학교 전자공학과 졸업
2002년 2월 : 고려대학교 전자공학과 석사 졸업
2010년 2월 : 서울대학교 전기컴퓨터 공학과 박사 졸업
2013년 3월~현재 : 숭실대학교 전자정보공학부 교수

<관심분야> 전자공학, 통신공학, 컴퓨터공학

www.dbpia.co.kr