논문 18-43-11-25 The Journal of Korean Institute of Communications and Information Sciences '18-11 Vol.43 No.11
https://doi.org/10.7840/kics.2018.43.11.1959

SDN을 위한 DNS 증폭 공격 방어 기법

최 동 호◆, 박 민 호°, 주 양 익°°

DNS Amplification Attacks Defense System for Software-Defined Networks

Dong-ho Choi◆, Min-ho Park°, Yang-ick Joo°°

요 약

Public DNS 서버는 로컬네트워크 상의 DNS 서버보다 대용량의 컴퓨팅 자원을 사용하여 수 백 만개의 도메인 주소에 대한 IP를 저장해 응답 속도가 빠르고, 국가정책 또는 지리적 이유 등의 문제로 차단된 웹사이트를 우회 접속하는 등의 다양한 부가 기능을 제공해 많은 사용자가 이용하고 있다. 하지만, Public DNS 서버는 공개되어 있어 누구나 접속이 가능하여 공격자의 DNS 서버를 이용한 공격의 수단으로 활용되었다. DNS 서버를 이용한 대표적인 공격인 DNS 증폭 공격은 공격자가 제어 가능한 봇넷을 통한 공격대상의 IP를 도용하여 Public DNS 서버로 DNS 쿼리를 전송하면, 공격대상이 Public DNS 서버로부터 다수의 DNS 응답메시지를 받아 불능상태에 빠지게 되는 문제를 발생시킨다. 본 논문은 SDN 환경에서 요청되지 않은 DNS 쿼리의 응답메시지를 차단하는 방어 기법을 이용한 DNS 증폭 공격 방어 시스템을 제안한다.

Key Words : DNS(Domain Name System), Public DNS Server, DNS Amplification Attacks, SDN(Software-Defined Networks)

ABSTRACT

The public DNS (Domain Name Service) has been used more and more because of some benefits, such as the quick response time by using a large amount of computing resources, and the bypass of web site blocking according to national policies or geographic reasons. However, the public DNS servers are open to the public, which makes them accessible to anyone and even vulnerable to various attacks, especially the DNS amplification attack which is the most well-known attack. In this attack, an attacker steals a lot of IP addresses via a controllable botnet, and sends DNS queries that seems to be sent by the stolen IP addresses to public DNS servers. Because a victim receives and deals with a number of DNS response messages from the public DNS servers, it can be exhausted, which reaches Distributed Denial of Service Attack (DDoS). This paper proposes a DNS amplification attack defense system by using the features of Software Defined Networks, and shows how efficiently it defends the attacks.

※ 이 논문은 2018학년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2018-0-00254. SDN 보안 기술개발)

◆ First Author : (ORCID:0000-0001-6299-1696)Department of ICMC convergence technology, Soongsil University, dhc@ssu.ac.kr, 학생회원

° Corresponding Author : (ORCID:0000-0003-3033-192X)School of Electronic Engineering, Soongsil University, mhp@ssu.ac.kr, 종신회원

°° Corresponding Author : (ORCID:0000-0003-3125-5316)Division of Electrical and Electronics Engineering, Korea Maritime and Ocean University, yijoo@kmou.ac.kr, 종신회원

논문번호 : 201806-D-038-RE, Received June 12, 2018; Revised August 20, 2018; Accepted September 18, 2018

Ⅰ. 서 론

DNS 서버는 숫자 형태의 IP 주소를 쉽게 기억하기 위하여 부여된 도메인 이름을 관리하고 도메인 이름에 대한 IP 주소 요청이나 IP 주소에 대한 도메인 이름 요청에 대해 응답하는 역할을 하여 수많은 웹사이트가 존재하는 네트워크에서 필수적인 요소이다. 하지만 DNS의 통신 대부분이 빠른 전송을 위해 비연결 지향성 서비스인 UDP를 이용하여 신뢰성과 안정성이 떨어지고, 이러한 점을 악의적으로 이용한 Cache poisoning, Compromising, Denying 등과 같은 공격이 늘고 있다[1].

DNS를 이용한 공격 중 하나인 DNS 증폭 공격은 공개되어있어 누구나 접근이 가능한 Public DNS 서버를 이용하여 공격대상이 대량의 DNS 응답메시지를 받아 불능상태에 빠지게 하는 공격으로 일종에 DNS 서버를 이용한 DDoS 공격이다. 공격대상이 일반 Host일 경우에 네트워크가 혼잡해짐으로써 대역폭(Bandwidth)이 크게 감소하여 네트워크 이용이 불가능한 문제가 발생하고, 회사나 금융기관의 DNS 서버일 경우에 이용자들에게 원활한 서비스 제공이 힘들어져 경제적인 손실인 2차적 피해까지 초래할 수 있다.

위와 같은 DNS 증폭 공격을 방어하기 위하여 이전 연구[2-4]에서는 DNS 쿼리와 DNS 응답 메시지를 데이터베이스에 저장시키고 이를 서로 매핑시켜 Host가 요청한 DNS 쿼리에 대한 응답 메시지임을 확인해 공격을 탐지하거나[2], 스위치에 Bloom filter를 적용하여 DNS 쿼리의 해시값과 수신되는 DNS 응답 메시지의 해시값을 비교하여 공격을 탐지하는 방법[3]이 제안되었지만, 추가적인 통신비용이 요구되거나 오탐율이 상승하는 문제가 발생되었다. 통신비용을 줄이고 공격탐지율을 상승시키는 방법으로 SDN 환경에서 발생되는 DNS 쿼리 내 송신지 IP(Source IP), 수신지 IP(Destination IP) 등의 정보를 스위치 메모리나 컨트롤러 메모리에 저장시키고 수신되는 DNS 응답 메시지의 정보와 비교하여 공격을 탐지하는 방법[4]이 제안되었지만, 공격으로 인한 스위치와 컨트롤러 사이에 발생되는 과부하를 고려하지 않았다.

본 논문에서는 SDN의 장점 중 하나인 흐름제어(Flow_control)을 이용하여 요청된 DNS 쿼리에 대한 DNS 응답메시지를 컨트롤러에서의 추가적인 확인없이 스위치에서 확인하여 수신하고, 요청하지 않은 DNS 쿼리에 대한 DNS 응답메시지를 차단하는 방어기법을 제안한다.

Ⅱ. 본 론

2.1 배경지식 및 관련 연구

2.1.1 SDN(Software-Defined Networks)

SDN(Software-Defined Networks; 소프트웨어 정의 네트워크)는 데이터 전송을 위한 전송 장치가 구성되어있는 데이터 계층(Data plane 또는 인프라 구조계층(Infrastructure layer))과 전체 망의 동작을 제어하는 제어 계층(Control layer)사이에 표준화된 인터페이스를 제공하며, 소프트웨어 프로그래밍을 통해 네트워크 운용자가 여러 상황에 맞게 패킷 전송에 대해 단순화되어있는 데이터 계층에서의 네트워크 경로 설정 및 복잡한 구성에 대한 운용 관리를 효과적으로 제어할 수 있는 차세대 네트워크 기술이다[5].

기존 네트워크 구조에서는 동적인 트래픽을 예측하여 제어하기에 어려움이 따르고, 복잡한 네트워크 구조로 정체되는 구간에 대한 네트워크 토폴로지 변경 및 업데이트가 진행되기 힘들었다. 또한, 네트워크 장비는 제조사에 의해 적용되는 폐쇄적인 구조이고, 표준 API나 개방된 인터페이스가 존재하지 않기 때문에 사용자에 맞는 환경 구축 및 추가적인 기능 개발의 어려움이 야기되었다. 하지만, SDN에서의 인터페이스 기술 중 하나인 OpenFlow[6]을 통해 기존 네트워크에서는 구성하기 어려운 복잡한 경로 설정을 컨트롤러에서 프로그래밍하여 효과적으로 제어하여 대처할 수 있다.

기존 네트워크 구조에서는 하나의 네트워크 장비에 제어 기능과 전송 기능이 동시에 이뤄져 수신되는 패킷에 대한 전달 업무를 네트워크의 각 장비마다 반복적으로 수행하는 비효율적이고, 네트워크의 각 장비를 일일이 확인하고 감시해야하는 복잡함과 새로운 응용 프로그램 도입 또는 네트워크 구성의 변경이 생길 경우 네트워크가 일시적으로 다운되는 문제가 생겼다. 하지만, OpenFlow의 도입으로 제어 계층과 데이터 계층으로 역할이 분리되어 네트워크상에서 하나의 컨트롤러가 모든 스위치를 중앙 집중적으로 제어함으로써 반복적인 전달 업무를 한 번에 처리해 업무량과 시간비용을 크게 줄이고, 네트워크 전체적인 상황을 고려하여 flow의 흐름을 프로그래밍을 통해 유연하게 제어하여 관리할 수 있게 되었다. 또한, 컨트롤러와 스위치 간 보안채널(Secure channel)을 통해 안전한 통신을 할 수 있다.

기존 네트워크의 단점을 보완할 수 있는 SDN을 도입하기 위해 국내·외로 많은 연구가 진행되어 현재 국

내에선 상용화 단계를 진행 중에 있지만, 기존 네트워크에서의 정적인 보안 솔루션을 SDN에서 이용하기에는 컨트롤러의 설계 및 데이터 평면의 구현에 따라 동적으로 변하는 SDN 환경에 적당하지 않다. SDN 환경에서 발생될 수 있는 문제에 대한 보안 솔루션과 아직 발견되지 않은 보안 취약성에 대한 지속적인 연구와 기술 개발이 시급한 상황이다.

2.1.2 DNS(Domain Name System)

DNS란, 숫자 형태의 식별 번호로 되어있는 호스트의 IP 주소를 더 단순하고, 쉽게 기억하기 위한 도메인 이름을 관리하고 도메인 이름에 대한 호스트의 IP 주소 요청이나 반대로 호스트의 IP 주소에 대한 도메인 이름 요청을 수행하도록 개발된 시스템이다. DNS는 도메인 네임 스페이스와 리소스 레코드, 네임서버, 리졸버(Resolver; 변환기)로 구성되어있다. 도메인 네임 스페이스는 중복되지 않는 도메인 네임이 유일한 값으로 정의된 공간이고, 리소스 레코드는 해당 도메인에 대한 레코드 Type이 저장된 공간이다. 일반적으로 호스트에 대한 32비트의 IP 주소(IPv4)를 알려주는 레코드 A Type과 호스트에 대한 128비트의 IP 주소(IPv6)를 알려주는 레코드 AAAA Type(Quad-A type), 도메인 이름의 별칭을 알려주는 레코드 CNAME 등의 레코드 Type 등이 저장된다. 또한, DNS 서버마다 추가적으로 제공하는 다양한 레코드 Type을 지원하기도 한다.

네임서버는 도메인 네임에 대한 정보를 가지고, 외부로부터 DNS 쿼리 요청이 있을 때, 가지고 있는 도메인 네임에 대한 정보를 조회하여 응답을 보내는 역할을 한다. 리졸버는 분산 구조의 DNS를 조회하기 위해 호스트에 구현된 소프트웨어 라이브러리 형태의 시스템이다. 네트워크의 규모가 커지면서 늘어나는 DNS 쿼리 및 응답메시지로 전체 트래픽이 큰 폭으로 증가하는 문제가 발생하여 트래픽 절감 및 DNS 절차의 효율성을 해결하기 위해 도메인 네임에 대한 데이터를 캐싱하는 기능을 수행하는 캐싱 네임서버(Caching Name Server; Recursive Name Server)를 사용하게 되었다. 캐싱 네임서버 운용 시 재귀적(Recursive) 서비스 제공의 제한이 없을 경우, DNS 캐시 포이즈닝 공격이나 DDoS 공격에 악용될 위험이 있다.

2.1.3 Public DNS 서버와 DNS 증폭 공격

도메인 네임에 대한 IP 주소를 알기 위해 대부분 사용자들은 로컬 네트워크상에 위치한 DNS 서버나 ISP가 제공하는 DNS 서버에 DNS 쿼리를 전송하여 요청한 도메인 네임에 대한 IP 주소가 담긴 DNS 응답 메시지를 수신함으로써 웹사이트에 접근할 수 있다. 하지만, 해당 도메인 네임의 IP 주소가 DNS 서버 캐시 내에 존재하지 않을 때, 루트 DNS 서버 등의 다른 DNS 서버로 요청하게 돼 이로 인한 지연시간이 발생된다. 반면, Public DNS 서버는 수백만 개의 웹사이트 주소를 저장할 수 있는 대용량의 캐시를 이용하여 DNS 쿼리에 대한 응답을 즉시 받을 수 있고, 국가 정책상이나 지리적인 이유 등의 문제로 차단된 웹사이트를 우회하여 접근하거나 도메인 네임 작성 시 발생된 오타를 인식하고 관련 웹사이트를 제공하는 Typo correction 기능, 해당 도메인 주소를 줄여서 간단한 단어로도 IP 주소를 받을 수 있는 Shortcut 기능 등의 다양한 부가 기능을 사용할 수 있어 많은 사용자들이 Public DNS 서버를 이용하고 있다. 하지만, DNS 서버는 인증이나 확인 절차 없이 정보를 주고받는 UDP 통신으로 신뢰성이 떨어지는 문제가 있고 특히, Public DNS 서버는 공개되어 있어 누구나 접근이 강하기 때문에 공격자들이 DNS 서버를 이용한 공격의 수단으로 이용되고 있다.

DNS 서버를 이용한 DDoS 공격 중 하나로 DNS 증폭 공격(DNS Amplification attacks)이 있다. 그림 1과 같이 공격자는 제어가 가능한 Botnet(악성코드 Bot에 감염된 PC로 이뤄진 집합)을 이용하여 공격대상에 대한 명령 하달을 한다. Bot들은 다 수의 Open DNS 서버에 DNS 쿼리 메시지의 송신지 IP(Source IP)를 공격자가 도용한 공격대상의 IP로 바꿔 다량의 DNS 쿼리를 전송하는데 이 때, 도메인 주소의 모든 데이터가 담겨 크기가 큰 DNS 응답 메시지를 요청하는 ANY Type의 DNS 쿼리 메시지를 전송한다. DNS 쿼리를 수신한 Public DNS 서버들은 DNS 쿼리의 송신지 IP(Source IP)인 공격대상으로 대량의 DNS 응답 메시지를 전송하고, 공격대상이 과도한 트래픽으로

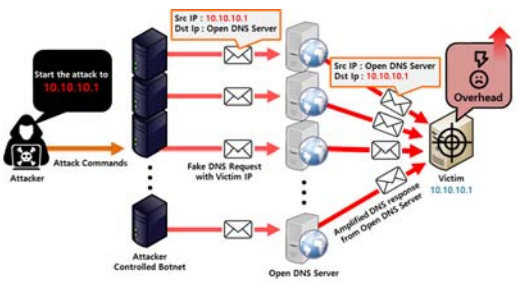

그림 1. DNS 증폭 공격 과정
Fig. 1. Process of DNS Amplification attacks

인한 불능 상태(Overhead; 과부하)에 빠지게 되는 공격이다. 일반적인 DDoS 공격은 공격자가 공격 시에 많은 트래픽을 발생시켜야했지만 DNS 증폭 공격에는 60Bytes의 DNS 쿼리로 도메인의 모든 정보가 담긴 약 4,000Bytes의 DNS 응답 메시지를 발생시키기 때문에 적은 양의 트래픽으로 많은 양의 공격 트래픽을 발생시킬 수 있다. 또한, 일반적인 DDoS 공격은 공격 대상이 Bot들의 IP를 차단해 공격을 방어할 수 있지만, DNS 증폭 공격은 도용된 공격대상의 IP로 생성된 DNS 쿼리에 대한 DNS 응답 메시지로 공격이 이뤄지기 때문에 기존 DDoS 공격 방법으로는 방어하기 어렵고 Public DNS 서버로부터 유입되는 모든 패킷을 차단하기에는 네트워크 내 Public DNS 서버를 이용하는 사용자에게 피해를 줄 수 있다.

국내에서는 2013년 6월 27일 주요 정부 기관을 대상으로 한 DNS 증폭 공격이 처음 발견[7]되었고, 세계적으로 2018년 1분기에 DNS 증폭 유형의 DDoS 공격이 지난 분기보다 두 배로 증가했으며 전년도 대비 거의 700% 증가하였다[8]. 이와 같이 전 세계 네트워크는 DNS 증폭 공격에 많이 노출되어 있으며, 점점 지능화되어가는 공격을 탐지하고 방어하기 위한 지속적인 연구가 필요한 상황이다.

2.1.4 관련 연구

DNS 증폭 공격에 대한 탐지 및 방어 관련 연구들이 많이 진행되어 왔다. 첫 번째 연구에서는 기존 네트워크에서 송신하는 DNS 쿼리 메시지와 이에 수신되는 DNS 응답 메시지를 1대 1 매핑하여 DNS 증폭 공격을 탐지하는 방법[2]을 제안하였다. 송신하는 DNS 쿼리의 정보와 이에 수신되는 DNS 응답메시지를 별도의 데이터베이스에 저장시키고, 이를 서로 매핑하여 Host가 요청한 DNS Query임을 확인한다. 이 때, 매핑되지 않은 DNS 응답 메시지는 공격으로 간주하고 제거(Drop)시키는 방법이다. 하지만 별도의 데이터베이스를 운용하여 추가적인 통신비용이 요구되는 문제가 발생되었다. 두 번째 연구에서는 추가적인 통신비용을 줄이고자 기존 네트워크에서 데이터베이스를 사용하지 않고 스위치의 내부 메모리에 Bloom filter를 적용하여 DNS 증폭 공격을 탐지하는 방법[3]을 제안하였다. DNS 쿼리 정보를 해시 값으로 저장하고, 수신되는 DNS 응답 메시지의 해시 값과 비교하여 해시 값이 다른 DNS 응답 메시지는 차단하는 방법이다. 하지만, Bloom filter를 사용할 경우, 단방향의 데이터만을 저장시키고 저장된 데이터를 삭제하기 어려워 오탐율(False Positive Ratio)이 증가하는 문제가 발생되기 때문에, DNS 증폭 공격을 제대로 탐지하기 어려웠다. 세 번째 연구에서는 SDN 환경을 이용하여 DNS 쿼리에 대한 송신지 IP(Source IP), 수신지 IP(Destination IP) 등의 정보를 스위치의 내부 메모리나 컨트롤러 내부 메모리에 저장시키고 이에 수신되는 응답 메시지의 정보와 비교하는 방법을 제안하였다[4]. 별도의 저장장치를 사용하지 않고 SDN 스위치와 SDN 컨트롤러 내부에 모든 DNS 쿼리의 정보를 저장하고 수신되는 DNS 응답 메시지의 정보와 저장된 DNS 쿼리 정보를 매핑하여 탐지하기에 정확도를 높일 수 있었다. 하지만, 수신된 DNS 응답 메시지의 정보는 응용 계층의 정보로써 전송 계층의 포트 번호까지만 확인할 수 있는 SDN 스위치에서는 불가능하기 때문에 오직 SDN 컨트롤러에서만 DNS 응답메시지임을 알 수 있다. DNS 증폭 공격이 발생되었을 때, 스위치로 유입되는 패킷들을 저장된 DNS 쿼리와 매핑하기 위해서는 우선적으로 DNS 메시지임을 확인하기 위해 컨트롤러로 Packet_In 메시지가 전송되는데, 많은 양의 패킷이 유입될 경우에는 다량의 Packet_In 메시지로 인하여 컨트롤러의 과부하(Overhead)가 발생되는 문제가 야기된다.

Ⅲ. 제안하는 SDN 환경에서의 DNS 증폭 공격 방어 시스템

3.1 제안하는 방어 시스템 구현

그림 2는 제안하는 방어 시스템의 논리적 구조이다. 제안하는 방어 시스템의 구현 및 실험에서는 SDN 컨트롤러 중 하나인 POX 컨트롤러 내에 제안하는 방어시스템을 구축하고, 컨트롤러와 SDN 스위치 중 하나인 OVS(Open VSwitch)의 스위치포트 S1과 연결하여 OpenFlow를 통한 Flow rule 생성 및 제어를 할

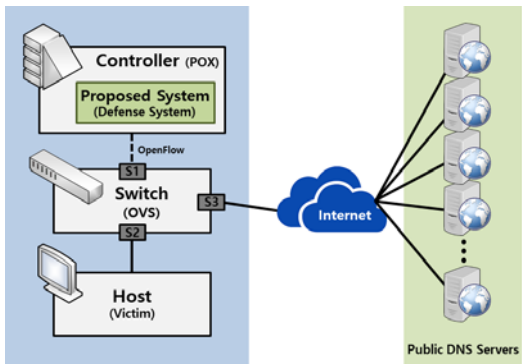

그림 2. 제안하는 방어 시스템의 논리적 구조
Fig. 2. Logical structure of proposed defense system

수 있도록 하였다.

OVS 안에 스위치포트 S2, S3를 생성하여 OVS 스위치포트 S2와 Host를 연결하고, OVS 스위치포트 S3과 외부 망을 연결해 Host가 Public DNS 서버를 이용할 수 있도록 하였다. 제안하는 방어 시스템은 크게 3 개의 구조로 이뤄져있다.

3.1.1 Malicious DNS Defender

Malicious DNS Defender는 공격자가 공격대상의 IP를 도용하여 Public DNS 서버로부터 발생시킨 대량의 응답메시지를 SDN 스위치(OVS)에서 차단하는 Rule을 생성한다.

그림 3의 ①과 같이 초기 상태에서 SDN 컨트롤러와 OVS가 서로 연결될 시, Malicious DNS Defender는 외부 망과 연결된 OVS 스위치포트 S2로 유입되는 모든 DNS 관련 Flow를 차단하는 Rule을 생성하기 위해 그림 3의 ②와 같이 Flow rule Installer에게 차단 Rule 생성정보(In_port=S2, Destination UDP Port=53(DNS))를 전달한다. Flow rule Installer는 그림 3의 ③과 같이 OVS로 Flow_mod message를 전송하여 OVS 내 Flow table에 Rule이 생성되도록 한다. Host가 요청하지 않은 DNS 응답 메시지가 OVS 스위치포트 S2를 통해 유입될 때, 그림 3의 ④와 같이 OVS 내 Flow table에 생성된 Rule에 의해 스위치포트 S2으로 수신되고 UDP Source port가 53인 DNS Packet은 차단된다. 제안하는 시스템의 네트워크 환경은 별도의 DNS 서버운용 없이 Public DNS 서버를 통해 DNS를 제공받고 있기 때문에, Public DNS 서버로부터 수신되는 모든 DNS 응답메시지는 DNS 쿼리가 Public DNS 서버에 전송된 후 받게 된다. 그렇기 때문에 외부로부터 요청되지 않은 DNS 응답메시지가 유입될 경우에는 모두 공격이라 간주할 수 있다.

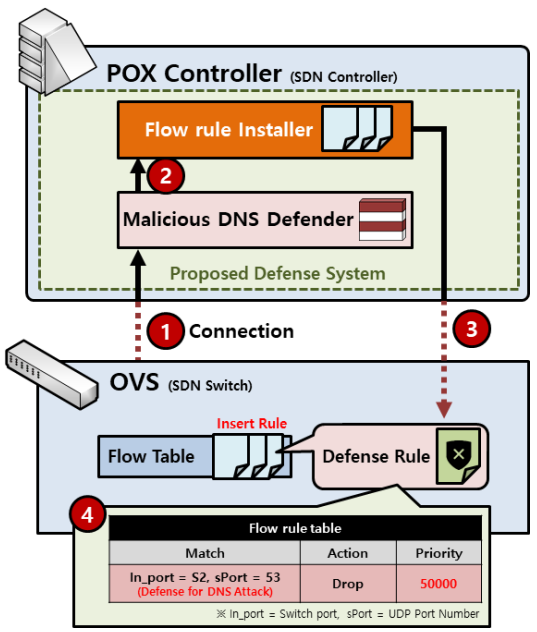

그림 3. Malicious DNS Defender 동작 과정
Fig. 3. Process of Malicious DNS Defender

3.1.2 DNS Checker

DNS Checker는 OVS에서 컨트롤러로 발생된 Packet_In 메시지 중 Host로부터 Public DNS 서버로 송신되는 DNS 쿼리에 대한 Rule과 Public DNS 서버로부터 Host로 수신되는 DNS 응답메시지에 대한 Rule을 생성하는 역할을 한다.

그림 4의 ①과 같이 Host가 DNS 쿼리를 전송하였을 때, OVS 내 Flow rule table에 해당하는 rule이 없기 때문에 그림 4의 ②와 같이 OVS에서 컨트롤러로 OpenFlow를 통해 Packet_In 메시지를 전송한다. 컨트롤러 내 제안하는 방어 시스템의 DNS Checker에서 Packet_In Handler를 통해 Packet_In 메시지를 수집하고, 수집된 Packet_In 메시지 중 수신지 IP(Destination IP)가 Public DNS 서버의 IP이고 수신지 UDP 포트번호가 53인 DNS Packet을 확인한다. DNS Packet일 경우, 해당 Packet_In 메시지에서의 송신지 IP(Source IP; Host IP), 수신지 IP(Destination IP; Public DNS 서버 IP), 스위치 포트번호(Port Number), 송신지 UDP 포트번호(Source UDP port Number(Random Number; 10000~65536)), 수신지 UDP 포트번호(Destination UDP port Number(53;

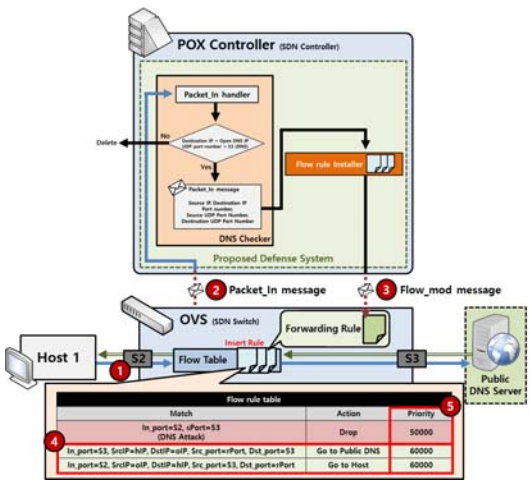

그림 4. DNS Checker 동작 과정
Fig. 4. Process of DNS Checker

DNS)) 등의 정보를 이용하여 Flow rule Installer을 통해 생성된 Flow_mod 메시지를 그림 4의 ③과 같이 OVS로 전송시켜 OVS 내 Flow table에 rule이 삽입되도록 한다.

이 때, 그림 4의 ④와 같이 외부 망에 위치해 있는 Public DNS 서버로 DNS 쿼리가 송신되는 Rule과 Public DNS 서버로부터 DNS 쿼리에 대한 DNS 응답 메시지가 수신되는 Rule을 같이 삽입시킴으로써 요청한 DNS 쿼리에 대한 DNS 응답메시지가 컨트롤러를 거치지 않고 생성된 Rule에 의해 바로 Host에게 전달될 수 있도록 한다.

3.1.3 Flow rule Installer

Flow rule Installer는 Malicious DNS Defender와 DNS Checker로부터 전달된 정보에 따라 OVS 내 Flow table에 rule을 생성시켜주기 위한 Flow_mod 메시지를 생성한다. 그림 4의 ⑤와 같이 요청하지 않은 DNS 쿼리에 대한 DNS 응답메시지를 차단하는 Rule보다 요청한 DNS 쿼리와 이에 대한 DNS 응답 메시지를 송·수신하는 Rule의 우선순위(Priority) 값을 높게 설정하여 처리하기 때문에 요청한 DNS 쿼리와 이에 대한 DNS 응답메시지는 수신하고, 요청하지 않은 DNS 쿼리에 대한 DNS 응답메시지는 차단한다.

3.2 DNS 증폭 공격에 대한 방어 과정

그림 5와 같이 Host가 외부에 위치해 있는 Public DNS로부터 DNS 쿼리를 전송할 때, 컨트롤러 내의 제안하는 방어 시스템으로부터 OVS에 송·수신되는 DNS 쿼리와 DNS 응답메시지에 대한 Flow rule를 삽입시킨다. DNS 쿼리에 대한 응답메시지를 수신할 때 삽입된 Rule에 의해 컨트롤러를 거치지 않고 Host에게 바로 전달된다. 외부 네트워크에 위치한 공격자가 Host의 IP주소를 도용한 DNS 증폭 공격을 진행하였을 때, Public DNS로부터 생성된 다량의 응답 메시지가 OVS로 전송되게 된다. 하지만, OVS Flow rule table 내에 요청하지 않은 DNS 쿼리에 대한 DNS 응답메시지는 Drop 시키는 rule이 삽입되어 있어 공격자의 DNS 증폭 공격을 방어할 수 있다. 또한, Host가 요청한 DNS 쿼리에 대한 DNS 응답 메시지는 Drop rule보다 우선순위가 높기 때문에 Host에게 전달된다.

Ⅳ. 실 험

본 실험에서는 Public DNS 서버를 이용한 DNS 증폭 공격에서 야기되는 문제점 증명 및 제안하는 방어 시스템의 성능 평가 실험을 위하여 테스트 환경이 아닌 실제 SDN 환경을 구축하였다. 실험 환경은 그림 6과 같이 Python 기반의 SDN 컨트롤러인 POX를 사용하여 OVS를 통해 Host(Victim)에서 외부 네트워크로 송·수신 되는 Flow를 제어할 수 있도록 하고, 컨트롤러 내부에 DNS 증폭 공격에 대한 방어 시스템을 구현하였다. 외부 네트워크에 위치한 공격자에서 Packet Generator Tool 중 하나인 Scapy[9]를 이용해 수신지 IP가 Host(Victim) IP로 변경된 DNS 쿼리를 초당 1,000~1,200개씩 Public DNS 서버로 전송시켰다. DNS 쿼리에서 요청하고자 하는 도메인 주소는 전 세계적으로 많이 알려진 500개의 웹사이트[10]와 국가기관 및 금융기관의 홈페이지를 사용하였고, 활발히 운영 중인 20개의 Public DNS 서버를 이용하였다[11].

그림 7과 같이 공격자가 다른 두 레코드 Type(A Type, ANY Type)의 DNS 쿼리를 200개 단위로 조절하여 Public DNS 서버에 전송하였다. 이에 Public DNS 서버로부터 수신하는 DNS 응답 메시지에 대하여 Host(Victim)와 외부 Host 사이의 대역폭(Bandwidth)의 변화량을 제안하는 시스템 유·무에 따라 각각 측정하여 비교하였다.

그림 8과 같이 Host와 외부 Host간의 대역폭

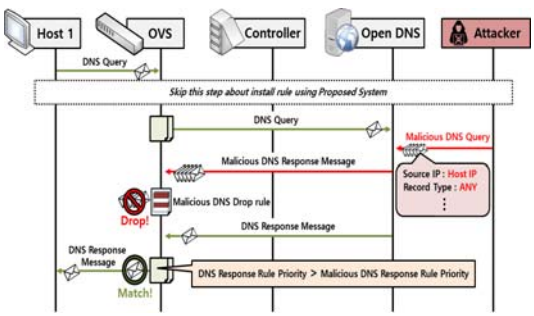

그림 5. DNS 증폭 공격에 대한 방어 절차
Fig. 5. Defense procedure of DNS Amplification Attacks

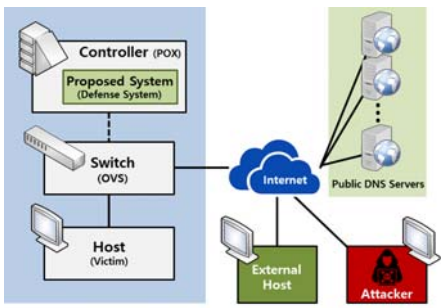

그림 6. SDN을 이용한 실험 환경
Fig. 6. Experimental environment with SDN

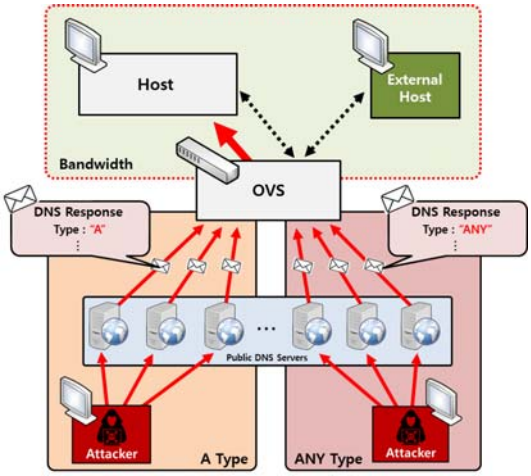

그림 7. DNS 증폭 공격 시 Host와 외부 Host 간의 대역폭(Bandwidth) 측정
Fig. 7. Bandwidth measurement between host and external host in DNS Amplification attacks

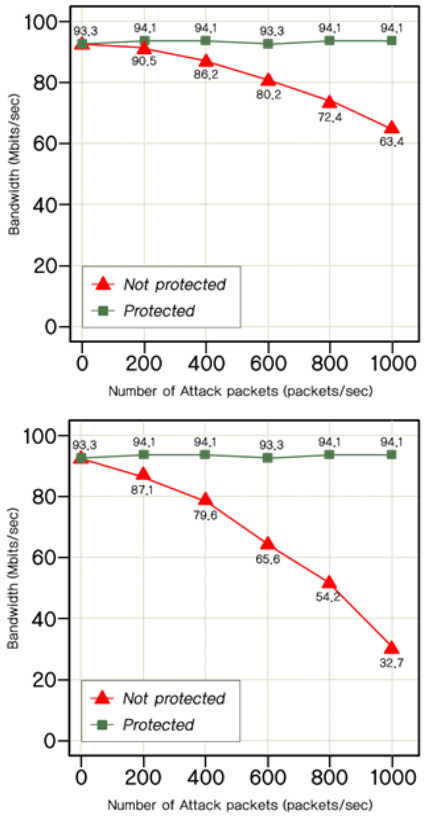

그림 8. 대역폭(Bandwidth) 측정 결과(위: A Type, 아래: ANY Type)
Fig. 8. Bandwidth measurement result (Top: A Type, Down: ANY Type)

(Bandwidth)이 A Type의 DNS 증폭 공격 시 63.4 Mbits/s로 공격 전과 약 20 Mbits/s의 차이를 보였고, ANY Type의 DNS 증폭 공격 시 32.7 Mbits/s로 대역폭이 크게 감소되어 ANY Type의 DNS 증폭 공격이 네트워크에 큰 영향을 미친다는 것을 확인하였다. 또한, 제안하는 방어 시스템을 사용하였을 경우에 발생되는 DNS 증폭 공격을 차단함으로써 평균 94.1 Mbits/s로 안정적인 대역폭이 유지되었다.

V. 결 론

본 논문에서는 SDN의 장점 중 하나인 흐름 제어(Flow_control)기능을 이용하여 Host에서 요청한 DNS 쿼리와 이에 수신될 DNS 응답 메시지에 대한 Flow rule을 생성하여 컨트롤러의 추가적인 확인 없이 스위치에서 바로 Host에게 전달하고, Public DNS를 이용한 DNS 증폭 공격에 의해 요청되지 않은 DNS 응답 메시지를 차단하는 기술에 대해 연구하였다. 시뮬레이션이 아닌 실제 SDN 환경을 구현한 실험을 통해 Public DNS 서버를 이용한 DNS 증폭 공격으로 초당 1,000개의 DNS 응답메시지가 유입되었을 때, ANY Type의 DNS 쿼리에 대한 응답메시지로 대역폭(Bandwidth)이 32.7 Mbits/s로 크게 감소해 ANY Type의 DNS 증폭 공격이 네트워크에 영향을 끼침을 나타내었다. 또한, 제안하는 방어 시스템을 사용하였을 경우, 공격을 차단해 안정적인 대역폭을 나타내어 우수성을 입증하였다. 제안하는 방어 기법이 DNS 증폭 공격 방어와 더불어 네트워크 보안에 큰 발전을 기여할 것으로 기대된다.

References

[1] J.-W. Choi, M.-J. Chun, D.-W. Hong, and C.-H. Seo, “A proposal countermeasure to DDoS attacks targered DNS,” *JKIISC,* vol. 23, no. 4, pp. 729-735. Aug. 2013.

[2] G. Kambourakis, T. Moschos, D. Geneiatakis, and S. Gritzalis, “Detecting DNS amplification attacks,” *CRITIS*, pp. 185-196, Málaga, Spain, Oct. 2007.

[3] S. Di Paola and D. Lombardo, “Protecting against DNS reflection attacks with Bloom filters,” *DIMVA 2011*, pp. 1-16, Amsterdam. The Netherlands, Jul. 2011.

[4] S. Kim, “Preventing DNS amplification

attacks using the history of DNS queries with SDN," *ESORICS 2017*, pp. 135-152, Oslo, Norway, Sep. 2017.

[5] S. Pack, I. Jang, D. Seo, and J. Lee, "New paradigm of future network About SDN/NFV," *KICS Inf. and Commun. Mag.*, vol. 32, no. 7, pp. 82-92, Jun. 2015.

[6] Open Networking Foundation, *OpenFlow Switch Specification*(2009), Retrieved Apr. 10, 2018, from http://www.opennetworking.org

[7] AhnLab ASEC, *Major government agencies DNS Amplification for DNS server DDoS attack,* Retrieved Apr. 15. 2018. from http://asec.ahnlab.com/951

[8] NEXUSGUARD, "*DDoS Threat Report 2018 Q1*," Retrieved Jul. 28, 2018, from http://www.nexusguard.com/threat-report-q1-2018

[9] Scapy, "*Welcome to Scapy's documentation*," Retrieved Aug. 12, 2018, from http://scapy.readthedocs.io/en/latest

[10] Alexa (Amazon AI), "*The top 500 sites on the web*," Retrieved Aug. 13, 2018, from http://www.alexa.com/topsites

[11] Lifewire, "*Free and Public DNS Servers*," Retrieved Aug. 13, 2018, from http://www.lifewire.com/free-and-public-dns-servers-2626062

최 동 호 (Dong-ho Choi)

2016년 2월 : 청운대학교 컴퓨터학과 졸업

2016년 9월~현재 : 숭실대학교 정보통신소재융합학과 석사과정

<관심분야> 컴퓨터공학, 통신공학

박 민 호 (Min-ho Park)

2000년 2월 : 고려대학교 전자공학과 학사 졸업

2002년 2월 : 고려대학교 전자공학과 석사 졸업

2010년 2월 : 서울대학교 전기컴퓨터공학과 박사 졸업

2013년 3월~현재 : 숭실대학교 전자정보공학부 교수

<관심분야> 전자공학, 컴퓨터공학, 통신공학

주 양 익 (Yang-ick Joo)

1998년 2월 : 고려대학교 전자공학과 학사 졸업

2000년 8월 : 고려대학교 전자공학과 석사 졸업

2004년 8월 : 고려대학교 전자공학과 박사 졸업

2004년 9월~2012년 2월 : 삼성전자 DMC 연구소 책임연구원

2012년 3월~현재 : 한국해양대학교 전자전기정보공학부 교수

<관심분야> 통신 보안, 무선자원관리