JKIICE Journal of the Korea Institute of Information and Communication Engineering

한국정보통신학회논문지(J. Korea Inst. Inf. Commun. Eng.) Vol. 21, No. 3 : 549~557 Mar. 2017

ICMP 공격 방지를 위한 outbound traffic controller의 설계 및 구현

유권정 · 김은기*

Design and implementation of outbound traffic controller for the prevention of ICMP attacks

Kwon-jeong Yoo · Eun-gi Kim*

Department of Information and Communication Engineering, Hanbat National University, Daejeon 34158, Korea

요 약

ICMP(Internet Control Message Protocol)는 TCP/IP 프로토콜 스택 중의 주요 프로토콜로, 오류 보고 기능을 지원하지 않는 IP의 단점을 보완하는 프로토콜이다. 데이터 전송 과정에서 문제가 발생하면 라우터 또는 수신 호스트는 오류 원인을 포함한 ICMP 메시지를 송신 호스트에게 전송한다. 하지만 이러한 과정에서 공격자가 위조된 ICMP 메시지를 호스트들에게 전송함으로써 호스트 간 통신을 비정상적으로 종료시킬 수 있다. 또는 대량의 메시지를 빠른 속도로 피해자 호스트에게 전송하여 피해자 호스트의 시스템을 마비시키기도 한다. 이러한 문제점을 해결하기 위해 본 논문에서는 여러 유형의 ICMP 공격들을 방지할 수 있는 아웃바운드 트래픽 컨트롤러(outbound traffic controller)를 설계 및 구현하였다. 각 경우에 따라 다른 방법으로 공격 메시지의 전송을 방지함으로서 다양한 네트워크 공격을 예방 할 수 있다. 또한 불필요한 네트워크 트래픽의 발생을 방지 할 수 있다.

ABSTRACT

ICMP(Internet Control Message Protocol) is a main protocol in TCP/IP protocol stack. ICMP compensates the disadvantages of the IP that does not support error reporting. If any transmission problem occurred, a router or receiving host sends ICMP message containing the error cause to sending host. However, in this process, an attacker sends a fake ICMP messages to the host so that the communication can be terminated abnormally. An attacker host can paralyzes system of victim host by sending a large number of messages to the victim host at a high rate of speed. To solve this problem, we have designed and implemented outbound traffic controller that prevents various ICMP attacks. By preventing the transmission of attack messages in different ways according to each case, various network attacks can be prevented. In addition, unnecessary network traffic can be filtered before transmitted.

키워드 : ICMP, 트래픽, 컨트롤러, 네트워크 공격, 보안

Key word : ICMP, traffic, controller, network attack, security

Received 07 December 2016, **Revised** 08 December 2016, **Accepted** 06 January 2017

* **Corresponding Author** Eun-Gi Kim(E-mail:egkim@hanbat.ac.kr, Tel:+82-42-821-1215)
Department of Information and Communication Engineering, Hanbat National University, Daejeon 34158, Korea

Open Access http://doi.org/10.6109/jkiice.2017.21.3.549 print ISSN: 2234-4772 online ISSN: 2288-4165

ⓒThis is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License(http://creativecommons.org/li-censes/ by-nc/3.0/) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.
Copyright Ⓒ The Korea Institute of Information and Communication Engineering.

Ⅰ. 서 론

ICMP(Internet Control Message Protocol)는 인터넷 상에서 발생할 수 있는 여러 가지 예외 상황의 원인을 파악하고 문제를 해결할 수 있도록 지원한다[1,2].

이러한 점을 이용하여 공격자 호스트는 위조된 메시지를 피해자 호스트에게 전송함으로써 호스트 간 통신을 비정상적으로 종료 시킬 수 있다. 또는 공격자 호스트가 ICMP 메시지의 내용을 위조하여 피해자 호스트에게 대량으로 전송함으로서 피해자의 시스템을 마비시키는 DoS 공격을 수행 할 수 있다[3]. 하지만 현재 ICMP 공격에 대한 근본적인 해결 방안은 제시되지 못하고 있다. 이러한 이유로 기존의 많은 운영체제들이 ICMP 메시지의 수신을 거부함으로써 공격을 방어한다[1,3]. 이와 달리 본 논문에서 제안하는 아웃바운드 트래픽 컨트롤러는 전송 측에서 ICMP 공격용 패킷의 전송을 차단함으로써 ICMP 공격을 원천적으로 불가능하게 한다. 결과적으로 다양한 ICMP 공격과 네트워크 내에 발생할 수 있는 불필요한 트래픽을 방지 할 수 있다.

본 논문에서는 ICMP 메시지를 이용한 land 공격, smurf 공격, ping of death 공격, ping flood 공격, blind connection-reset 공격을 방지할 수 있는 방안을 제시한다. 이를 위해 아웃바운드 트래픽 컨트롤러는 리눅스 커널 소스 코드를 수정하여 구현되었다. 그리고 구현된 아웃바운드 트래픽 컨트롤러를 실제 임베디드 보드에 포팅 하여 테스트함으로써 ICMP 공격을 예방 할 수 있고 불필요한 트래픽이 발생되지 않는 것을 확인하였다.

본 논문의 순서는 다음과 같다. 2장에서는 관련 연구와 ICMP 공격의 종류 및 유형에 대해 설명하고 3장에서는 제안하는 아웃바운드 트래픽 컨트롤러의 설계를 기술한다. 4장에서는 기존 시스템에서 ICMP 공격을 수행한 결과와 아웃바운드 트래픽 컨트롤러의 동작 검증에 대해 설명하고 마지막으로 5장에서는 결론을 다룬다.

Ⅱ. 관련 연구

2.1. ICMP

일반적으로 TCP/IP 프로토콜에서 데이터는 네트워크 계층에 존재하는 IP를 통해 데이터 링크 계층으로 전송된다. 하지만 IP는 신뢰성이 없고 비연결형 전송 서비스를 제공하며 오류 보고 및 정보 수집 기능이 없다[4]. 이러한 이유 때문에 IP 패킷이 전송되는 중에 네트워크 문제가 발생한다면 호스트는 해당 문제를 해결 할 수 없다. ICMP는 이를 보완하기 위한 프로토콜로, 네트워크 문제가 발생하면 현재 통신 중인 상대 호스트에게 오류 원인을 포함한 메시지를 전송한다. 이를 수신한 상대 호스트는 현재 연결되어 있는 통신을 종료하거나 문제를 해결하기 위한 방안을 모색 할 수 있다. ICMP는 논리적으로 네트워크 계층에 존재하며 ICMP 메시지는 바로 데이터 링크 계층으로 전달되지 않고 IP 패킷에 캡슐화 되어 전송된다[1].

ICMP 메시지는 오류 보고 메시지와 질의 메시지로 나뉜다. 오류 보고 메시지는 라우터 또는 호스트가 통신 중에 문제가 발생 했을 때 사용된다. 질의 메시지는 호스트 또는 네트워크 관리자가 라우터 또는 다른 호스트로부터 특정 정보를 얻어오기 위해 사용된다[2-5]. ICMP 메시지의 구조는 다음 그림 1과 같다.

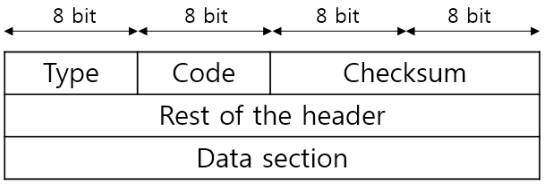

Fig. 1 ICMP message format

타입(type)은 ICMP 메시지의 용도를 나타내며, 코드(code)는 특정 타입의 원인을 나타낸다. 체크섬(checksum)은 ICMP 전체 메시지에 대한 오류 검출 기능을 지원한다[1,4,5].

2.2. 기존의 ICMP 공격 유형

2.2.1. 위조된 메시지를 전송하여 호스트 간 연결 종료를 유발하는 공격

위조된 메시지를 전송하는 유형의 공격들은 상대방의 동작 여부를 확인하는 메시지를 통해 이루어진다. 이때 사용되는 메시지의 타입은 8(echo request)과 0(echo reply)이다. Echo request 메시지는 상대 호스트의 동작 여부를 확인하기 위해 응답을 요청하는 메시지로, 송신 호스트가 수신 호스트에게 전송한다. 이를 수신한 호스트는 echo reply 메시지를 통해 응답함으로써

현재 자신이 동작하고 있음을 나타낸다[1,2].

이러한 유형의 공격에는 land 공격, smurf 공격, ping of death 공격, blind connection-reset 공격이 있다.

Land 공격이란 공격자가 echo request 메시지의 출발지 IP 주소(source IP address)와 목적지 IP 주소(destination IP address)를 피해자 호스트의 IP 주소로 설정한 뒤 피해자 호스트에게 전송하는 것이다. 이를 수신한 피해자 호스트는 응답하기 위해 echo reply 메시지를 전송한다. 이때 출발지 IP 주소가 피해자 호스트의 IP 주소이기 때문에 echo reply 메시지는 피해자 호스트에게 돌아온다. 이를 빠른 속도로 반복하게 되면 시스템은 과부하에 걸리게 된다[6,7].

Smurf 공격은 공격자 호스트가 echo request 메시지의 출발지 IP 주소를 피해자 호스트의 IP 주소로 설정하고, 목적지 IP 주소를 망 내에 있는 전체 호스트들로 설정하여 전송하는 것이다. 이를 수신한 호스트들은 응답하기 위해 피해자 호스트의 IP 주소로 echo reply 메시지를 전송한다. 피해자 호스트는 무방비 상태에서 수많은 echo reply 메시지를 수신하게 되고 피해자 호스트의 시스템은 과부하 상태가 된다[3, 6-8]. Ping of death 공격은 공격자 호스트가 echo request 메시지의 데이터 크기를 최대치로 설정한 뒤 피해자 호스트에게 전송하는 것이다. 이때 데이터는 MTU (maximum transmission unit)에 맞춰 단편화(fragmentation)되어 전송된다. 이를 수신한 피해자 호스트는 단편화 된 데이터의 조각들을 재조립한다. 이 과정을 빠른 속도로 반복하면 피해자 호스트는 단편화 된 데이터들의 재조립을 반복하게 되면서 과부하에 걸리게 된다[3,7,8].

Blind connection-reset 공격은 공격자 호스트가 피해자 호스트에게 통신 연결이 불가함을 나타내는 메시지를 전송함으로써 이루어진다. 이때 사용되는 메시지의 타입은 3(Destination unreachable)이고, 코드는 2(Protocol unreachable), 3(Port unreachable), 4(Fragmentation needed and DF bit set)인 것으로 전체 3개가 존재한다. 타입 3(Destination unreachable)은 전송한 메시지가 수신 호스트에게 전달 될 수 없음을 나타낸다. 코드 2(Protocol unreachable)는 송신 호스트가 요청한 프로토콜을 수신 호스트가 지원 할 수 없을 때 전송된다. 코드 3(Port unreachable)은 송신 호스트가 요청한 포트를 수신 호스트가 지원 할 수 없을 때 전송된다. 코드 4(Fragmentation needed and DF bit set)는 호스트가 수신한 데이터에 대해서 단편화(fragmentation)가 필요한 경우에 수신한 패킷의 DF(Don’t fragment) 비트가 설정되어 있어서 단편화를 수행 할 수 없음을 나타낸다. 이와 같은 메시지들을 수신한 송신 호스트는 수신 호스트와의 연결을 종료한다[4,5].

2.2.2. 상대방의 네트워크 취약점을 스캔하는 공격

이 유형의 공격은 공격자 호스트가 실질적인 공격을 수행하기 전에 피해자 호스트에 대한 정보를 획득하기 위한 과정으로, 기본적으로 포트 스캔 방식을 사용한다[7,9]. 따라서 별다른 피해를 끼치지 않으며 사전 정보 획득을 목적으로 한다. 그리고 수집된 정보를 가지고 DDoS 공격과 같은 실질적인 공격을 수행한다. 이러한 유형의 공격 중 ICMP를 이용한 공격은 sweep 공격이 있다.

Sweep 공격은 공격자 호스트가 네트워크 내에 존재하는 모든 호스트들에게 echo request 메시지를 전송하는 것으로, 응답을 전송하는 호스트가 현재 동작하고 있음을 알 수 있다[7,9].

2.2.3. DoS(Denial of Service) 유형의 공격

DoS 공격이란 대량의 데이터를 빠르게 전송하여 상대방의 시스템을 마비시키는 공격으로, 아직까지 정확한 해결 방안이 제시되지 못하고 있다[10,11]. 이러한 유형의 공격 중 ICMP를 이용한 공격은 ping flood 공격이 있다.

Ping flood 공격은 공격자 호스트가 엄청난 속도로 echo request 메시지를 피해자에게 전송하여 피해자의 시스템을 마비시키는 공격이다. 이때 상대 호스트의 응답을 기다리지 않고 가능한 빨리 전송한다. 공격자 호스트가 전송한 echo request 메시지를 수신한 피해자는 응답하기 위해 echo reply 메시지를 전송하게 되고, 이 과정이 반복되면 피해자 호스트의 수신 대역폭뿐만 아니라 송신 대역폭까지 모두 소비되어 피해자 호스트의 시스템은 접속 불능의 상태가 된다[3,10,11].

Ⅲ. 아웃바운드 트래픽 컨트롤러의 설계

본 논문의 아웃바운드 트래픽 컨트롤러는 2절에서 기술한 공격 메시지의 전송을 제한하기 위해 다음과 같

이 설계되었다. 또한 운영체제로 리눅스(linux)를 이용하였고 3.2.78 버전의 커널을 수정하여 구현되었다[5].

3.1. sk_buff 구조체와 rtable 구조체

본 논문에서는 송수신되는 패킷의 IP 주소에 대한 진위 여부를 확인하기 위해 sk_buff 구조체와 rtable 구조체를 이용한다.

sk_buff 구조체는 소켓 버퍼(socket buffer)로 사용되는 구조체로, 호스트와 프로토콜의 정보를 포함하고 리눅스 내부에서 이동하며, 실제 송수신되는 데이터를 관리한다. 따라서 sk_buff 구조체의 값은 유저 영역에서 입력한 값으로 변경 될 수 있다. 이 구조체는 include/linux/skbuff.h 파일에 정의되어 있다[6,12].

rtable 구조체는 라우팅 정보를 담고 있으며, 목적지에 해당하는 하드웨어 주소를 알아내는데 사용된다. 또한 데이터에 대한 송수신 처리를 임시 저장해 두는 dst_entry 구조체를 포함하고, IP 고유의 정보를 포함한다. 따라서 유저 영역에서 rtable 구조체의 값을 변경할 수 없다. 이러한 이유로 전송되는 패킷의 IP 주소에 대한 위조 여부를 확인 할 수 있다. 이 구조체는 include/net/route.h 파일에 정의되어 있다[6,12].

3.2. 커널 내부에서의 아웃바운드 트래픽 컨트롤러

해킹 툴은 일반적으로 RAW_SOCKET을 사용한다. RAW_SOCKET이란 호스트가 데이터를 조작하여 응용프로그램 계층(Application layer)에서부터 데이터링크 계층(Data-link layer)까지 직접 송수신 할 수 있게 하는 소켓이다[13]. 따라서 사용자 영역에서 해킹 툴을 사용하여 생성된 패킷은 일반적인 IP 패킷의 송신 과정과 다르게 데이터링크 계층으로 직접 전송된다. 이러한 점을 이용하여 본 논문에서 제안하는 아웃바운드 트래픽 컨트롤러는 위조된 패킷을 감지하기 위해 데이터링크 최하위 계층이면서 데이터의 단편화가 이루어지기 전에 동작한다. 그림 2는 리눅스 커널의 IP 계층에서 패킷이 외부로 나가는 경우에 호출되는 함수들을 보여주고 있으며, 본 연구에서 제안하는 아웃바운드 트래픽 컨트롤러는 이러한 함수들을 수정하여 구현되었다[12]. 함수 ip_finish_output()은 전송할 IP 패킷을 검사하고 필요에 따라서 ip_fragment()나 ip_finish_output2() 함수를 호출한다. ip_fragment 함수는 패킷이 MTU 이상의 크기를 갖는 경우 호출되며, 패킷의 단편화(fragmentation) 작업을 수행한다. ip_finish_output2()는 인라인(inline) 함수로 패킷의 MAC 헤더 부분을 조사하여 패킷이 올바른 링크 계층으로 전달한다.

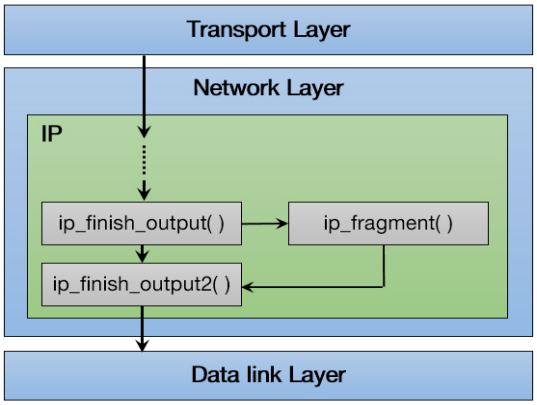

Fig. 2 Outbound traffic controller in network layer

3.3. 본 논문에서 제안하는 ICMP 공격 방지 방법

Land 공격의 경우, 전송되는 echo request 메시지의 출발지 IP 주소와 목적지 IP 주소가 동일하다. 두 값의 동일 여부를 확인하기 위해 sk_buff 구조체의 출발지 IP 주소와 목적지 IP 주소를 비교한다. 두 값이 동일한 경우, 루프백(loopback) 여부를 확인하기 위해 sk_buff 구조체의 출발지 IP 주소와 rtable 구조체의 출발지 IP 주소를 비교한다. 동일하지 않을 경우, 해당 메시지의 출발지 IP 주소는 위조되었음을 의미한다. 따라서 해당 패킷을 land 공격으로 판단하고 폐기한다. 이와 다르게 sk_buff 구조체의 출발지 IP 주소와 rtable 구조체의 출발지 IP 주소가 동일할 경우, 루프백(loopback)을 의미하므로 해당 패킷의 전송을 허용한다. 이미 여러 운영체제에서는 수신한 echo request 메시지의 출발지 IP 주소와 목적지 IP 주소가 일치할 경우에 land 공격으로 판단하고 해당 패킷을 폐기하기 때문에 land 공격이 가능한 경우는 드물다[3]. 하지만 본 논문에서 제안하는 아웃바운드 트래픽 컨트롤러는 공격 방지뿐만 아니라 네트워크 내에 불필요한 트래픽이 발생하지 않도록 공격용 ICMP 메시지로 판단된 패킷을 네트워크로 전송되기 전에 폐기한다.

Smurf 공격의 경우, 전송되는 echo request 메시지의 출발지 IP 주소와 실제 자신의 IP 주소가 다르다. 두 값을 확인하기 위해 전송되는 패킷의 정보를 담고 있는

sk_buff 구조체와 라우팅 정보를 담고 있는 rtable의 출발지 IP 주소를 비교한다. 그 결과, 다를 경우에 해당 패킷을 smurf 공격으로 판단하고 폐기한다.

Ping of death 공격의 경우, 전송되는 echo request 메시지의 데이터 크기를 최대치로 설정한다. 일반적으로 인터넷에서 많이 쓰이는 이더넷의 MTU로는 1500 byte 이며[14], 본 논문에서는 echo request 메시지의 데이터 크기가 1500 byte보다 클 경우는 없을 것으로 판단하였다. 이러한 이유로 본 논문에서는 전송되는 echo request 메시지의 데이터 크기가 1500 byte를 초과하면 ping of death 공격으로 판단하고 해당 패킷을 폐기한다. Sweep 공격의 경우, 망 내의 전체 호스트들에게 echo request 메시지를 전송하기 위해 패킷의 목적지 IP 주소를 broadcast로 설정한다. 따라서 본 논문에서는 전송되는 echo request 메시지의 목적지 IP 주소가 broadcast인 경우에 해당 패킷을 폐기한다.

Blind connection-reset 공격은 특정 원인으로 인해 통신 연결이 불가함을 나타내는 destination unreachable 메시지를 이용한다. Destination unreachable 메시지 중, 코드 2(Protocol unreachable), 코드 3(Port unreachable), 코드 4(Fragmentation needed and DF bit set) 메시지들은 상대 호스트로부터 연결 요청 패킷을 수신 한 뒤, 경우에 따라 전송된다[4]. 때문에 본 논문에서는 해당 메시지가 정상적으로 수신한 패킷에 대해 발생한 것임을 확인하기 위해 sk_buff 구조체에 수신 플래그인 'incoming flag'를 추가하였으며, 초기 값은 0이다. 일반적으로 상대 호스트로부터 정상적으로 패킷을 수신할 경우, 패킷은 ip_rcv_finish() 를 거치게 된다[12]. 이 점을 이용하여 본 논문에서는 ip_rcv_finish() 로 수신된 패킷 중 패킷의 수신 타입이 unicast인 경우에만 해당 패킷의 incoming_flag 값을 1로 설정한다. 이와 달리 해킹 툴을 사용하여 생성한 ICMP 메시지는 해당 함수를 거치지 않으므로 플래그 값이 1로 변경되지 않는다. 따라서 호스트가 destination unreachable 메시지를 전송하려고 할 때, sk_buff 구조체 내의 incoming flag 값이 '1'이면 정상적으로 수신 된 패킷에 대해 발생된 ICMP 오류 메시지임을 알 수 있다. 이와 달리, incoming flag 값이 '1'이 아니면 해당 메시지는 위조된 패킷임을 알 수 있다. ip_rcv_finish()는 net/ipv4/ip_input.c 파일에 정의되어 있다[12].

Ping flood 공격의 요점은 빠른 속도로 패킷을 전송한다는 것이다. 때문에 본 논문에서는 1초 동안 echo request 메시지가 최대 전송 값만큼 전송되면 ping flood 공격으로 판단하고 전송 지연 시간만큼 echo request 메시지의 전송을 제한한다. 본 논문에서는 사용자가 유저 영역에서 전송 지연 시간과 최대 전송 개수를 본인이 원하는 값으로 설정 할 수 있도록 하였다.

Ⅳ. 동작 검증

아웃바운드 트래픽 컨트롤러의 동작을 검증하기 위해 와이어샤크(wireshark) 프로그램을 이용하여 전송되는 공격용 패킷을 캡처하였다. 그리고 ICMP 공격을 수행하기 위해 hping2 해킹 툴을 이용하였다.

4.1. 기존 커널에서 ICMP 공격을 수행한 결과

공격자 호스트인 pc와 피해자 호스트인 pc는 허브를 통해 유선으로 연결되어 있으며, 둘 다 기존 커널을 사용하는 환경에서 ICMP 공격을 수행하였다. 본 논문에서 공격자 호스트의 IP 주소는 (x.x.x.185)이고, 피해자 호스트의 IP 주소는 (x.x.x.107)이다.

Land 공격을 수행하기 위해서 목적지 호스트의 IP 주소와 출발지 호스트의 IP 주소를 모두 피해자 호스트의 IP 주소로 설정하여 전송한다. 그 결과, 그림 3과 같이 land 공격용 패킷이 전송됨을 볼 수 있다. 그림 3을 보면 피해자 호스트가 자신에게 echo request 메시지를 전송하는 것처럼 보이고 있다. 하지만 현재 대부분의 운영체제에서 수신되는 패킷의 출발지 IP 주소와 목적지 IP 주소가 같으면 해당 패킷을 폐기한다. 본 논문에서 이용하는 운영체제도 그 중 하나이기 때문에 echo reply 메시지는 전송되지 않는다.

Source	Destination	Protocol	Length	Info
[illegible].107	[illegible].107	ICMP	42	Echo (ping) request
[illegible].107	[illegible].107	ICMP	42	Echo (ping) request
[illegible].107	[illegible].107	ICMP	42	Echo (ping) request
[illegible].107	[illegible].107	ICMP	42	Echo (ping) request
[illegible].107	[illegible].107	ICMP	42	Echo (ping) request

Fig. 3 Land attack packets

Smurf 공격을 수행하기 위해서 목적지 IP 주소를 broadcast로 설정하고 출발지 IP 주소를 피해자 호스트의 IP 주소로 설정하여 전송한다. 그 결과, 그림 4와 같이 피해자 호스트가 echo request 메시지를 broadcast로

전송하고 다른 호스트들로부터 echo reply 메시지를 수신한 모습을 볼 수 있다. 하지만 실제로 피해자 호스트는 echo request 메시지를 전송하지 않았음에도 불구하고 그림 4와 같은 결과가 발생한다.

Source	Destination	Protocol	Length	Info
203.230.102.107	185.230.102.255	ICMP	60	Echo (ping) request
203.230.102.16	185.230.102.107	ICMP	60	Echo (ping) reply
203.230.102.59	185.230.102.107	ICMP	60	Echo (ping) reply
203.230.102.18	185.230.102.107	ICMP	60	Echo (ping) reply
203.230.102.85	185.230.102.107	ICMP	60	Echo (ping) reply

Fig. 4 Smurf attack packets

Ping of death 공격을 수행하기 위해서 echo request 메시지의 데이터의 크기를 65000 byte로 설정하였다. 그 결과, 그림 5와 같이 공격자 호스트가 보낸 패킷이 단편화되어 전송되는 모습을 볼 수 있다. 그림 6과 그림 7은 단편화 된 패킷의 내부구조이다. 전송되는 패킷은 그림 6의 total length 값을 통해 이더넷의 MTU에 맞춰져 단편화 됐음을 알 수 있다. 또한 그림 7을 통해 단편화 된 패킷의 데이터는 의미 없는 값으로 채워져 있음을 알 수 있다.

Source	Destination	Protocol	Length	Info
185.230.102.185	203.230.102.107	IPv4	1514	Fragmented IP
185.230.102.185	203.230.102.107	IPv4	1514	Fragmented IP
185.230.102.185	203.230.102.107	IPv4	1514	Fragmented IP
185.230.102.185	203.230.102.107	IPv4	1514	Fragmented IP
185.230.102.185	203.230.102.107	IPv4	1514	Fragmented IP
185.230.102.185	203.230.102.107	IPv4	1514	Fragmented IP
185.230.102.185	203.230.102.107	IPv4	1514	Fragmented IP

Fig. 5 Ping of death attack packets

```
⊟ Internet Protocol Version 4, Src
    Version: 4
    Header Length: 20 bytes
  ⊞ Differentiated Services Field:
    Total Length: 1500
```

Fig. 6 Total length of fragmented packets

```
⊟ Data (1480 bytes)
    Data: 0800dcfd6b52000058585858585858585858
    [Length: 1480]

0010  05 dc 00 6b 20 00 40 01  ef c4 cb e6 66 b9
0020  66 6b 08 00 dc fd 6b 52  00 00 58 58 58 58
0030  58 58 58 58 58 58 58 58  58 58 58 58 58 58
0040  58 58 58 58 58 58 58 58  58 58 58 58 58 58
0050  58 58 58 58 58 58 58 58  58 58 58 58 58 58
0060  58 58 58 58 58 58 58 58  58 58 58 58 58 58
0070  58 58 58 58 58 58 58 58  58 58 58 58 58 58
```

Fig. 7 Data of Fragmented packet

Ping flood 공격을 수행하기 위해 공격 메시지 전송 간격을 1 μs로 설정한 뒤, echo request 메시지를 전송한다. 그 결과, 그림 8과 같이 빠른 속도로 echo request 메시지가 전송되는 것을 볼 수 있다.

Time	Source	Destination	Protocol	Length	Info
1.972417000	185.230.102.185	203.230.102.107	ICMP	42	Echo (ping) request
1.972458000	185.230.102.185	203.230.102.107	ICMP	42	Echo (ping) request
1.972485000	185.230.102.185	203.230.102.107	ICMP	42	Echo (ping) request
1.972503000	185.230.102.185	203.230.102.107	ICMP	42	Echo (ping) request
1.972523000	185.230.102.185	203.230.102.107	ICMP	42	Echo (ping) request
1.972544000	185.230.102.185	203.230.102.107	ICMP	42	Echo (ping) request
1.972561000	185.230.102.185	203.230.102.107	ICMP	42	Echo (ping) request
1.972578000	185.230.102.185	203.230.102.107	ICMP	42	Echo (ping) request
1.972595000	185.230.102.185	203.230.102.107	ICMP	42	Echo (ping) request
1.972611000	185.230.102.185	203.230.102.107	ICMP	42	Echo (ping) request
1.972629000	185.230.102.185	203.230.102.107	ICMP	42	Echo (ping) request

Fig. 8 Ping flood attack packets

Sweep 공격을 수행하기 위해 echo request 메시지의 목적지 IP 주소를 broadcast로 설정하여 전송한다. 그 결과, 그림 9와 같이 망 내의 호스트들로부터 응답이 수신되는 것을 볼 수 있다.

Source	Destination	Protocol	Length	Info
203.230.102.185	185.230.102.255	ICMP	42	Echo (ping) request
203.230.102.16	185.230.102.185	ICMP	60	Echo (ping) reply
203.230.102.59	185.230.102.185	ICMP	60	Echo (ping) reply
203.230.102.18	185.230.102.185	ICMP	60	Echo (ping) reply
203.230.102.85	185.230.102.185	ICMP	60	Echo (ping) reply

Fig. 9 Sweep attack packets

Blind connection-reset 공격을 수행하기 위해 ICMP 메시지의 각 타입과 코드를 설정한 뒤 피해자 호스트에게 전송한다. 다음 그림 10부터 그림 12까지 그림은 와이어샤크 프로그램을 통해 피해자 호스트가 수신한 공격용 패킷을 캡처한 것이다.

```
⊟ Internet Control Message Protocol
    Type: 3 (Destination unreachable)
    Code: 2 (Protocol unreachable)
```

Fig. 10 Protocol unreachable message

```
⊟ Internet Control Message Protocol
    Type: 3 (Destination unreachable)
    Code: 3 (Port unreachable)
```

Fig. 11 Port unreachable message

```
⊟ Internet Control Message Protocol
    Type: 3 (Destination unreachable)
    Code: 4 (Fragmentation needed)
```

Fig. 12 Fragmentation needed and DF bit set message

4.2. 아웃바운드 트래픽 컨트롤러의 동작

공격자 호스트인 임베디드 보드와 피해자 호스트인 pc는 허브를 통해 유선 랜으로 연결되어 있으며, 구현한 아웃바운드 트래픽 컨트롤러는 임베디드 보드에 포팅 하였다. 임베디드 보드는 falinux의 EZ-S3C6410 (ARM S3C6410 chip, 128 MB RAM)을 사용하였다. 다음 그림 13은 아웃바운드 트래픽 컨트롤러의 동작을 검증하기 위한 실험 환경 사진이다. 노트북은 임베디드 보드에 명령어를 입력하기 위해 사용되고, 데스크톱은 피해자 호스트로써 사용된다.

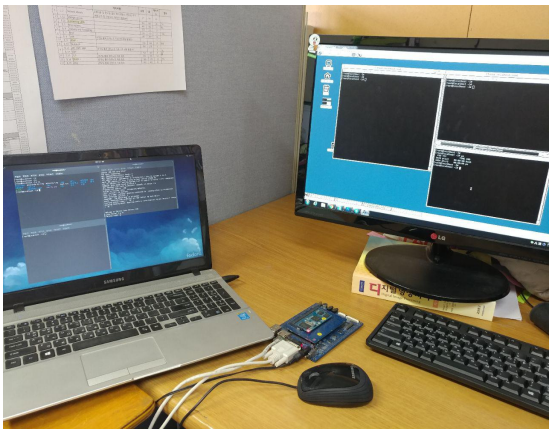

Fig. 13 Test environment

동작 검증 중에 전송되는 공격용 패킷은 아웃바운드 트래픽 컨트롤러에 의해 폐기되기 때문에 캡처될 패킷은 존재하지 않는다. 따라서 아웃바운드 트래픽 컨트롤러의 동작 검증에서는 wireshark 프로그램이 사용되지 않는다.

공격 수행 시, 공격자 호스트의 IP 주소는 x.x.x.200이고, 피해자 호스트 IP 주소는 x.x.x.107로 설정하였다. 또한 공격 유형을 확인하기 위해 공격이 감지되면 유저 영역에서 해당 공격 이름 또는 공격 유형이 출력되도록 하였다. 그림 14-17은 각 공격 명령어와 공격이 수행된 모습이다.

```
[root@falinux ~]$ hping2 --icmp 203.230.102.107 -a 203.230.102.107
HPING 203.230.102.107 (eth0 203.230.102.107): icmp mode set, 28 hsl
*** This is LAND attack! ***
```

Fig. 14 Land attack command and capture

```
[root@falinux ~]$ hping2 --icmp 203.230.102.255 -a 203.230.102.107
HPING 203.230.102.255 (eth0 203.230.102.255): icmp mode set, 28 hsl
*** This is smurf attack ***
```

Fig. 15 Smurf attack command and capture

```
[root@falinux ~]$ hping2 --icmp 203.230.102.107 -d 65000
HPING 203.230.102.107 (eth0 203.230.102.107): icmp mode s
*** This is ping of death! ***
```

Fig. 16 Ping of death attack command and capture

```
[root@falinux ~]$ hping2 --icmp 203.230.102.255
HPING 203.230.102.255 (eth0 203.230.102.255): i
*** This is sweep attack! ***
```

Fig. 17 Sweep command and capture

그림 18은 ping flood 공격 명령어와 공격을 수행했을 때의 모습이다.

```
[root@falinux ~]$ hping2 --icmp 203.230.102.107 -i u1
HPING 203.230.102.107 (eth0 203.230.102.107): icmp mode set,
eaders + 0 data bytes
```

Fig. 18 Ping flood command and capture

```
count: 95... delay timer init[ 946690270.70086 ]
delay timer exit[ 946690275.80053 ] ######
```

Fig. 19 Echo request count timer

그림 19에서 count 값은 1초 동안에 전송된 echo request 메시지의 개수이다. 또한 그림 19에서 보이는 숫자는 시스템이 부팅 된 후로부터 지난 시간을 초(sec)로 나타낸 값이다. 그림 19를 보면 1초 동안에 95개의 echo request 메시지가 전송됐음을 알 수 있다. 그 결과 echo request 메시지의 전송을 제한하는 타이머가 동작됐고, 5초 뒤 타이머가 해제됐음을 알 수 있다.

그림 20-22는 blind connection-reset 공격 명령어와 공격이 수행된 모습이다. 그 결과, 해당 공격 유형이 출력되는 것을 확인 할 수 있다.

```
[root@falinux ~]$ hping2 --icmp -C 3 -K 2 203.230.102.107
HPING 203.230.102.107 (eth0 203.230.102.107): icmp mode set,
        ******** This packet is FAKE! *******
```

Fig. 20 Destination unreachable (protocol unreachable) command and capture

```
[root@falinux ~]$ hping2 --icmp -C 3 -K 3 203.230.102.107
HPING 203.230.102.107 (eth0 203.230.102.107): icmp mode set,
        ******** This packet is FAKE! *******
```

Fig. 21 Destination unreachable (port unreachable) command and capture

```
[root@falinux ~]$ hping2 --icmp -C 3 -K 4 203.230.102.107
HPING 203.230.102.107 (eth0 203.230.102.107): icmp mode set,
        ******** This packet is FAKE! *******
```

Fig. 22 Destination unreachable (fragmentation needed and DF bit set) command and capture

hping2를 사용하여 ICMP 공격을 수행한 결과, 공격용 패킷은 아웃바운드 트래픽 컨트롤러에 의해 전송되지 않고 폐기된다. 결과적으로 위조된 패킷은 전송되지 않는다. 따라서 캡처될 패킷은 존재하지 않는다.

Ⅴ. 결 론

본 논문에서는 ICMP를 이용한 네트워크 공격을 방지하기 위해 다양한 공격용 ICMP 메시지가 네트워크로 전송되지 않게 하는 아웃바운드 트래픽 컨트롤러(outbound traffic controller)를 제안하였다.

일반적으로, 수신되는 메시지를 차단하거나 거부함으로써 공격을 방어하는 방법과는 다르게 본 논문에서 제안하는 아웃바운드 트래픽 컨트롤러는 공격 메시지가 전송되는 것을 방지함으로써 네트워크상의 불필요한 트래픽이 발생되는 것을 방지 할 수 있다. 따라서 다양한 유형의 ICMP 공격을 방지하는 것은 물론 호스트가 DDOS 공격에 노출되어도 좀비 pc가 되는 것을 예방할 수 있을 것으로 예상된다.

추후에는 본 논문의 아웃바운드 트래픽 컨트롤러를 이용하여 더욱 다양한 네트워크 공격을 방지하는 할 수 있는지에 대한 연구를 수행할 예정이다.

ACKNOWLEDGMENTS

This research was financially supported by the Ministry of Trade, Industry and Energy (MOTIE) through the Promoting Regional specialized Industry (No. R0003847)

REFERENCES

[1] B. A. Forouzan, *TCP/IP Protocol Suite*, Fourth Edition. New York, NY: McGRAW HILL INTERNATIONAL EDITION, p. 246, 2010.

[2] IETF Std. RFC 1122, *Requirements for Internet Hosts -- Communication Layers*, IETF, R. Braden, October 1989.

[3] D. Yang, *Introduction to information security*, Seoul, Hanbit Academy. Inc, pp. 102-107, 2008.

[4] IETF Std. RFC 5927, *ICMP Attacks against TCP*, IETF, F. Gont, July 2010.

[5] K. J. Yoo, E. G. Kim, “A study on the outbound traffic controller for prevention of ICMP attacks,” *in The 40 th Conference of KIICE*, Daejeon, pp. 1-3, 2016.

[6] T. hirokazu, O. iturou, Y. isaku, *Linux kernel 2.6 structure and principles*, Seoul, Hanbit Media Inc., pp. 460-503, 2007.

[7] M. Tulloch, *Microsoft Encyclopedia of Security*, Redmond, Microsoft Press Pub., pp. 138-247, 2003.

[8] J. Erickson, *Hacking: the art of exploitation*, 2nd Edition, Seoul, Acorn Pub, pp. 256-319, 2010.

[9] L. Teo. (2000, December). Network Probes Explained: Understanding Port Scans and Ping Sweeps [Internet]. Available: http://www.linuxjournal.com/article/4234.

[10] S. H. Kim, C. W. Yoon, *Hacking and security*, Seoul, youngjin.com Pub., pp. 50-251, 2003.

[11] R. A. Grimes, *Malicious mobile code*, Sebastopol, O'Reilly Media, p. 312, 2001.

[12] R. Rosen, *Linux Kernel Networking-Implementation and Theory*, Berkley, Apress Pub., 2014.

[13] W. W. Gay, *Linux socket programming by example*, Indianapolis, Que Pub., pp. 118-120, 2000.

[14] IETF Std. RFC 894, *A Standard for the Transmission of IP Datagrams over Ethernet Networks*, IETF, Charles Hornig, April 1984.

유권정(Kwon-Jeong Yoo)

2015년 2월 : 한밭대학교 정보통신공학과 졸업 (정보통신공학 학사)
2017년 2월 한밭대학교 정보통신전문대학원 졸업 (정보통신공학 석사)
※관심분야 : 네트워크 보안, 컴퓨터 네트워크, 암호화, 시스템 프로그래밍, 임베디드 S/W

김은기(Eun-Gi Kim)

1989년 2월 : 고려대학교 대학원 전자공학과 (전자공학 석사)
1994년 2월 : 고려대학교 대학원 전자공학과 (전자공학 박사)
1995년 2월 ~ 현재 : 한밭대학교 정보통신공학과 교수
※관심분야 : 컴퓨터 네트워크, 임베디드 S/W, 암호화, 네트워크 보안