Journal of the Korea Academia-Industrial cooperation Society
Vol. 17, No. 10 pp. 732-742, 2016

http://dx.doi.org/10.5762/KAIS.2016.17.10.732
ISSN 1975-4701 / eISSN 2288-4688

저대역 DDoS 공격 대응 시스템

이형수[1], 박재표[2*]
[1]숭실대학교 대학원 컴퓨터학과, [2]숭실대학교 정보과학대학원

Respond System for Low-Level DDoS Attack

Hyung-Su Lee[1], Jae-Pyo Park[2*]
[1]Department of Computer, Graduate School, SoongSil University
[2]Graduate School of Information Science, SoongSil University

요 약 본 논문에서는 향후에도 지속적으로 발생 가능성이 높은 저대역 DDoS 공격에 대비하여 TLF(Time Limit Factor)를 적용한 솔루션을 기존의 고대역 DDoS 방어 시스템에 추가함으로써 고대역의 DDoS 공격과 더불어 저대역 DDoS 공격에 대해서 방어 할 수 있도록 하였다. 저대역 DDoS 공격은 정상적인 서비스 연결을 가장하여 연결된 세션을 지속적으로 점유함으로써 정상적인 사용자들의 서비스 요청에 대한 장애를 유발시킨다는 점에 착안하여 각 세션별 일정시간 동안의 통신량을 체크하여 비정상적인 경우 저대역 DDoS 공격으로 간주하여 해당 세션을 종료시키는 방법이다. 그러나, 정상적인 연결 상태에서도 네트워크의 일시적인 장애들로 인해 통신에 장애를 가져오는 경우 저대역 DDoS 공격으로 오탐하여 서비스를 차단할 수 있다는 점 때문에 저대역 DDoS 공격으로 탐지되었다 할지라도 관련 정보에 대해 Blacklist를 통한 Drop이 아닌 일정시간동안만 Blocking 후 다시 재 접속이 가능하도록 하였다. 고대역 DDoS 방어시스템을 이용하여 저대역 DDoS 공격에 대한 테스트를 진행한 결과 고대역 DDoS 방어시스템은 저대역 DDoS 공격으로 단순 연결된 세션들에 대해 정상적인 통신으로 인지하여 세션에 대한 차단이 불가하였으며 이로 인해 저대역 DDoS 공격을 받은 시스템은 리소스 고갈로 서비스 불가 현상이 발생하였다. 본 논문에서 제안한 TLF 알고리즘을 고대역 DDoS 방어시스템에 적용하게되면 고대역 및 저대역 DDoS에 대한 방어가 가능할 뿐만 아니라, 서비스를 제공하는 시스템에 모듈형태로 추가 적용을 할 경우 저대역 DDoS 공격에 대한 대처가 가능하다.

Abstract This study suggests methods of defense against low-level high-bandwidth DDoS attacks by adding a solution with a time limit factor (TLF) to an existing high-bandwidth DDoS defense system. Low-level DDoS attacks cause faults to the service requests of normal users by acting as a normal service connection and continuously positioning the connected session. Considering this, the proposed method makes it possible for users to show a down-related session by considering it as a low-level DDoS attack if the abnormal flow is detected after checking the amount of traffic. However, the service might be blocked when misjudging a low-level DDoS attack in the case of a communication fault resulting from a network fault, even with a normal connection status. Thus, we made it possible to reaccess the related information through a certain period of blocking instead of a drop through blacklist. In a test of the system, it was unable to block the session because it recognized sessions that are simply connected with a low-level DDoS attack as a normal communication.

Keywords : DDoS, TLF Algorithm

1. 서론

국내 초고속 인터넷 통신망 인프라의 성장에 따라 대부분의 네트워크들은 기가비트 네트워크로 진화 되었으며, 일반 가정에도 100Mbps급의 초고속 인터넷이 보급되어 있다. 또한, 스마트폰과 같은 이동통신기기들의 사

*Corresponding Author : Jae-Pyo Park(Soongsil University)
Tel: +82-2-820-0270 email: pjerry@ssu.ac.kr
Received July 7, 2016 Revised August 18, 2016
Accepted October 7, 2016 Published October 31, 2016

용 확대에 따라 무선 인터넷 환경들이 전국으로 구축되어 있어 우리나라는 언제 어느 곳에서나 인터넷을 자유롭게 이용할 수 있는 유비쿼터스 환경이 구축되어 있다고 할 수 있다. 이러한 통신망 인프라의 발달은 사람들에게 언제 어디서나 필요한 서비스를 제공받을 수 있는 환경을 제공하고 있지만 이에 반하는 사이버 위협도 지속적으로 증가하고 있다. 사이버 위협은 대표적으로 해킹, 바이러스 및 DoS(Denial of Service)와 DDoS (Distributed Denial of Service) 공격이 있다. 해킹이나 바이러스의 공격은 특정된 시스템을 대상으로 하고 있는 반면, DoS 공격은 시스템을 포함한 서비스 및 네트워크 전체를 공격 대상으로 하고 있어서 피해의 정도는 훨씬 크다고 할 수 있다. 특히, 2009년에 발생한 7.7 DDoS공격은 인터넷 대란이라고 할 수 있는 인터넷 전체에 커다란 위협이 되었으며, 이러한 공격은 매년 지속적인 증가세를 보이고 있다. 또한, 자동화된 공격도구들의 인터넷 오픈에 따라 전문적인 지식을 가지지 않은 사람이라도 누구나 인터넷에 위협을 줄 수 있게 되었다. 이는 향후 더욱 더 많은 사이버 위협들이 발생할 수 있다는 의미이며, 이러한 결과는 선의의 사용자들까지 피해를 입게 되어 정상적인 서비스를 받을 수 없게 될 수 있다. 이렇듯 진화 및 확장되는 사이버 위협에 대한 공격을 막을 수 있는 기술적인 대응을 위한 연구 및 대응방안을 마련하여 보다 안전한 인터넷 환경을 제공하고 선의의 사용자들에 대한 정상적인 서비스를 보장할 수 있는 방안이 연구되어야 한다.

2. 관련연구

2.1 DDos 공격 및 유형

DDoS 공격은 DoS 공격을 짧은 시간에 여러 곳에서 발생시키도록 하는 공격으로 대부분의 공격이 자동화된 툴을 이용한다. 또한 DDoS 공격은 DoS 공격을 증폭시켜주는 중간자가 필요하며 이를 위해 악성봇을 이용한 수많은 좀비 PC를 만들어 사용한다[13].

대량의 트래픽을 발생시켜 대역폭을 소모시키기 위한 단순 Flooding 공격과 서버나 장비의 부하를 유발시키는 Connection 공격 그리고 특정 서비스의 방해를 목적으로 하는 Application 공격으로 구분할 수 있다[3]. 또한, DDoS 공격의 유형은 대량의 패킷을 발생시켜 네트워크 및 시스템의 리소스를 모두 마비시키는 고대역 DDoS 공격과 정상적인 통신 패킷을 이용하여 시스템의 취약점을 공격 리소스의 고갈을 통해 정상적인 서비스를 방해하는 저대역 DDoS 공격으로 나누어 볼 수 있다. 고대역 DDoS공격은 대표적으로 ICMP Flooding, SYN Flooding, UDP Flooding 공격들이 있으며, 저대역 DDoS의 경우 Slowrolis를 예로 들 수 있다.

Table 1. Type of DDoS Attack

	Flooding Attack	Load-induced attack of server / device	Denial-of-service attack specific
Attack methods	ICMP/UDP Flooding DNS Query Flooding BGP DRDoS	Syn Flooding Fragmented Packet Flooding	CC Attack Get Flooding Slowloris
Effect of the attack	To exceed the bandwidth capacity of N / W	Load-induced N / W device server, Load induction of security equipment	Induction of server load
Type of Attack	Traffic transmission to exceed the bandwidth	64byte following Packet processing induce beyond server Backlog queue	Depletion of the CPU resources than the number of sessions

2.2 고대역 DDos 공격 유형

2.2.1 ICMP Flooding 공격

ICMP(Internet Control Message Protocol)는 IETF RFC792에 정의된 프로토콜로써 호스트 간 혹은 상태 변화를 알려주고 요청에 응답을 하는 기능을 담당하는 네트워크 제어프로토콜로 활성화된 서비스나 포트가 필요하지 않는 유일한 프로토콜이다[1-2].

이러한 ICMP의 특징을 악용한 ICMP Flooding은 대량의 ICMP 패킷을 공격자가 직접 victim에게 전송하는 방법으로, 그 변종의 예로 Smurf, Welchia worm 등이 있다.

Smurf는 공격자가 source IP 주소를 victim의 IP 주소로 설정한 후, broadcast 주소로 ICMP echo request 패킷을 전송하면 그 하위 모든 시스템들이 ICMP echo reply 패킷을 victim으로 전송하게 되어 대량의 패킷들이 집중하여 네트워크 부하를 높이게 된다. Welchia worm은 감염 시스템에 대하여 IP 주소의 B 클래스를 고정시키고 C 클래스부터 증가시키며 ICMP 패킷을 전송하여 다른 감염 대상을 찾고 감염 시스템의 성능을 저하시키는 형태이다[4].

2.2.2 SYN Flooding 공격

SYN Flooding 공격은 특정 시스템에 대한 불법적인 권한을 얻는 적극적인 방법이 아니라 네트워크와 시스템의 자원을 공격대상으로 하는 공격 방법이다. 이것은 TCP가 데이터를 보내기 전에 연결을 먼저 맺어야 하는 연결지향성을 이용한 방법이다[8]. 즉, SYN Flooding은 TCP의 연결과정인 Three-way handshaking을 이용하여 공격자(Attacker)가 대상 시스템(Victim)에 source IP address를 spoofing하여 SYN 패킷을 특정 포트로 전송하게 되면 이 포트의 대기 큐(Back-log-Queue)를 가득차게 하여 이 포트에 들어오는 연결요청을 큐가 비울 때까지 무시하도록 하는 방법이다[4-5].

2.2.3 UDP Flooding 공격

UDP(User Datagram Protocol)를 이용한 패킷전달은 비연결형(connectionless) 서비스로서 포트 대 포트로 전송한다. 대표적인 응용 서비스로 TFTP, SNMP, 실시간 인터넷 방송 등을 들 수 있다[7].

UDP Flooding은 UDP의 비연결성 및 비신뢰성 때문에 공격이 용이한 방법이다. UDP는 source address와 source port를 spoofing하기 쉽다는 약점들을 이용해 과다한 트래픽을 victim에 전송함으로써 spoof되는 victim 간 네트워크를 마비시킨다. 이 공격은 주로 echo와 chargen 서비스에 이용하며 [Fig. 1]와 같이 동작한다.

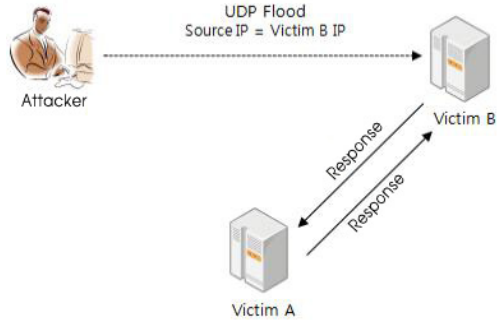

Fig. 1. UDP Flooding Attack

공격자가 victim A에게 source IP 주소를 victim B의 IP 주소로 spoofing하여 대량의 UDP 패킷을 전송하면 victim A와 victim B는 계속해서 서로 패킷을 주고 받게 되어 두 시스템 사이의 네트워크에 과부하가 초래된다[2][8].

2.3 저대역 DDos 공격

고대역 DDoS의 경우 공격자는 정상적인 통신에 필요한 세션의 연결성을 배제하고 초당 수만에서 수십만개 단위의 대량 패킷을 생성하여 victim으로 전송하여 시스템의 서비스를 고갈 시키지만, 저대역 DDoS 공격은 초당 약 2~3000개의 소량의 패킷으로 특정 시스템의 특정 어플리케이션과 TCP의 Three-way handshaking을 이용한 정상적인 방법으로 세션을 연결 하며 [Fig. 2]와 같이 동작 한다.

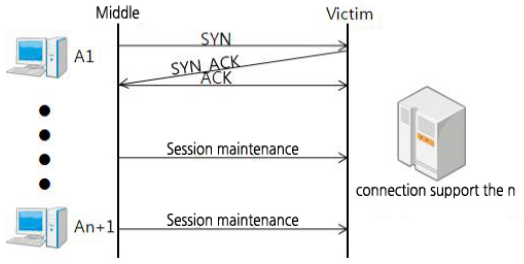

Fig. 2. Low-Level DDoS Attack

세션이 연결된 Agent에서는 일정 시간마다 Keep Alive 패킷을 전송하여 서버와의 연결을 지속적으로 유지하도록 요청하고 패킷을 전달 받은 서버는 세션을 계속 유지하면서 Agent들의 추가적인 서비스 요청을 기다리게 된다. Agent들의 추가적인 연결요청이 계속되면 서버는 연결가능한 세션의 수를 초과하게 되고 결국 추가적인 서비스에 대한 응답을 할 수 없게 된다.

2.3.1 Slowris

Slowloris는 가장 최근인 2009년 6월 17일 발표된 DoS 공격 툴로써 아파치 웹서버를 겨냥하여 만들어 졌다. 이 툴은 많은 HTTP Connection 을 연결하여 웹서버가 MaxClient에 도달하게 함으로써 HTTP 서비스가 중단되도록 하는 형태의 공격이다.

마치 TCP SYN Flooding 공격과 유사하게 일단 정상적인 GET 접속 요청을 한 후 [Fig. 3]의 패킷과 같이 마지막에 하나의 CRLF을 하지 않아 서버에서 대기를 하게 한 후 timeout 에 도달하게 하거나 또는 무의미한 헤더를 지속적으로 전송하게 된다[9][10].

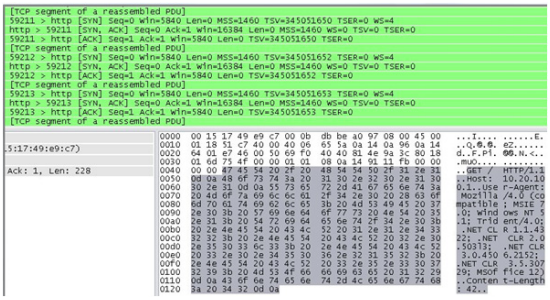

Fig. 3. Contents of the attack packets using Slowloris

이 공격은 아직 마지막의 CRLF 를 입력하지 않았기 때문에 서버는 마지막 CRLF를 받을 때까지 세션을 유지하면서 대기 상태가 되며 결국 MaxClient에 도달하여 서비스가 중단하게 된다.

2.4 기존 DDoS 방어시스템의 구성과 문제점

2.4.1 기존 DDos 방어시스템의 구성

현존하는 대부분의 DDoS 방어시스템은 Layer 3, Layer 4, Layer 7에 대한 고대역 DDoS 공격 방어가 가능하도록 구성되어 있으며 [Fig. 4]과 같이 각 Layer별 필터를 통해 고대역 DDoS 공격을 방어하도록 되어 있다.

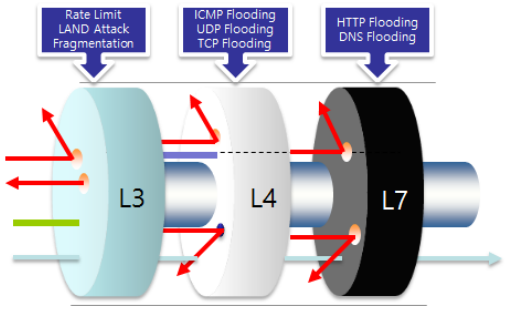

Fig. 4. Method of operation of the DDoS defense system

고대역 DDoS 공격의 탐지를 위해 방어시스템에 보안 정책을 설정하여 정해진 범위를 넘어가는 과도한 Traffic이 발행하는 경우 DDoS로 탐지하도록 하고 있다. 즉, 정해진 시간동안 일정량 이상의 통신이 발생하게 되면 이를 인지하도록 하는 Traffic Rate Limit 정책을 통해 대규모로 발생되는 Flooding 공격을 탐지 하는 것이다 [11]. Layer 7의 경우 HTTP Get Flooding 또는 CC(Cache Control) Attack 공격의 경우 모든 패킷의 데이터 부분까지 검사하여 유해성 여부를 검사 할 수 있도록 구성되어 있다[9]. 또한, 들어오는 모든 패킷을 분석하고 처리하여야 하는 부담을 줄이기 위해 특정 IP에서의 공격 빈도를 파악하여 공격자 또는 좀비 PC로 인지된 경우 Blacklist에 등록하여 전처리 과정 중에 패킷을 Drop 하도록 하여 패킷 처리에 대한 성능 향상을 꾀하고 있다[12].

2.4.2 기존 DDoS 공격 대응 방안 문제점

기존의 DDoS 방어시스템들이 대부분 2.3절에서 살펴보았던 DDoS 대응 방안을 기반으로 하여 개발되어진 시스템으로 다음과 같은 문제점들을 가지고 있다.

- 고대역 DDoS 공격에 초점이 맞추어져 있어 저대역 DDoS 공격에 대한 대응 방안 부재
- Layer 7의 어플리케이션 취약점을 이용하는 DDoS 공격에 대해 HTTP의 일부 공격에 대한 탐지만 지원하며, 다른 서비스에 대한 대응 방안 미비
- 공격으로 탐지된 IP 및 서비스 포트에 대해 패킷의 Drop 기능만 지원

고대역 DDoS 공격 경우 특성상 공격 IP의 대부분이 무작위로 Spoofing된 IP를 사용하게 되는데, 이때 방어시스템에 의해 탐지된 IP의 Backlist 등록과 Drop으로 인해 정상적인 IP를 사용하고 있는 사용자의 서비스 요청도 같이 Drop되는 문제점이 존재한다. 또한 일시적인 네트워크 장애에 의한 패킷의 비정상 전달로 인해 Layer 7단계에서 DDoS로 판단하게 되는 경우에도 동일한 문제점으로 인해 정상적인 서비스를 받을 수 없게 된다.

3. TLF를 이용한 저대역 DDoS 공격 대응

3.1 저대역 DDoS 공격 대응 개선 방향

지금까지 살펴본 DDoS 공격에 대한 대응 방안 및 고대역 DDoS 방어시스템에서 처리하지 못하고 있는 저대역 DDoS 공격에 대해 다음과 방법으로 대응 하도록 한다.

- 모든 세션에 대해 Time Limit Factor를 추가
- 각 세션별 특정시간 동안의 통신 상태를 모니터링 하여 정해진 용량 이하로 통신하는 경우 저대역 DDoS 공격으로 탐지

- 공격으로 탐지된 세션에 대해서는 Drop 또는 정해진 시간동안 Blocking
- Drop의 경우 동작 방법
① 해당되는 세션 테이블의 모든 정보 제거
② 해당 IP 및 정보를 Blacklist에 등록 후 차단
- Blocking의 경우 동작 방법
① 해당되는 세션 테이블 Time counter Enable
② Blocking 시간동안 해당 세션의 패킷 버림
③ Blocking 시간이 지나면 해당 세션 테이블의 모든 정보 제거
④ 세션 성립 과정부터 다시 시작

3.2 저대역 DDoS 공격 대응 시스템 설계

3.2.1 시스템의 구조

[Fig. 5]은 기존 DDoS 방어시스템에 저대역 DDoS 공격에 대한 방어 솔루션을 구현한 방어시스템의 구조를 나타내다.

기존 DDoS 방어시스템과의 차이점은 세션 테이블에 TLF를 처리하기 위해 여분의 필드들을 추가하였으며, 세션 테이블을 참조하여 각 필터를 통과하기 이전에 패킷의 전달 과정을 처리하는 전처리부의 일부 함수를 변경하였다. 그리고 저대역 DDoS를 실제적으로 탐지하는 Layer 7의 Low Level Attack 처리부와 패킷의 Drop이 아닌 Blocking을 위한 모듈이 추가되었다. 본 논문에서는 위의 방어시스템 구조에서 기존 고대역 DDoS를 위한 부분을 제외하고 저대역 DDoS공격을 위해 새로이 추가된 부분에 대해서만 기술 하였다.

3.2.2 주요 관리 구조 변경 설명

DDoS 방어시스템은 공격 탐지에 필요한 각 Layer 필터들을 가지고 있으며 각 필터들의 동작 방법을 결정하기 위한 방법으로 보안정책에 임계치 및 동작 방법등을 설정 하게 된다. 또한 Layer 4이상에서의 Rate Limit 적용 및 컨텐츠의 내용을 분석하기 위해서 세션 테이블을 관리하고 있으므로 이들 주요 관리 Factor에 대한 구조를 변경하여 저대역 DDoS 공격에 대응 할 수 있도록 한다.

(1) 세션 테이블에 TLF 추가

각 세션의 상태들을 관리하고 DDoS 탐지 시 해당 세션을 Blocking 하기 위해 [Fig. 6]와 같이 기존의 세션 테이블에 Time Rate Limit, Blocking Time 및 세션 상태관리를 위한 필드를 추가 하였다.

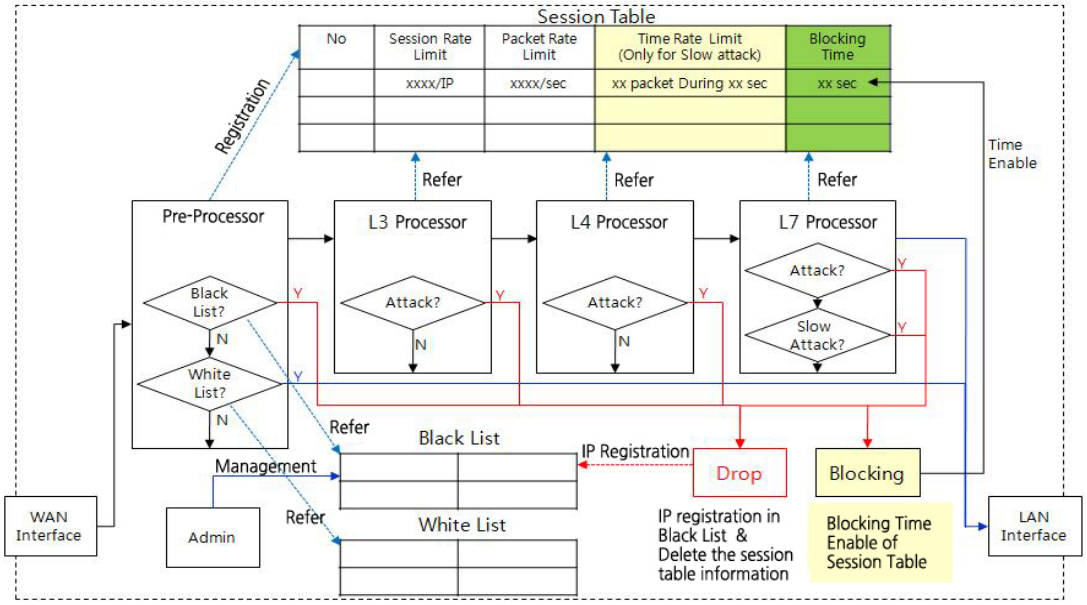

Fig. 5. Low-Level DDoS The structure of the defense system

Session No.	SIP	SPort	DIP	DPort

Session No.	SIP	SPort	DIP	DPort	TRL (Time Rate Limit)		BT (Blocking Time)		Check
					count	sec	sec	stat	

Fig. 6. Session table with added TLF

TRL(Time Rate Limit)은 세션이 연결되면 정해진 시간(Sec)동안 일정량(Count) 이상의 패킷 통신이 발생하지 않을 경우 저대역 DDoS로 간주하게 된다. BT(Blocking Time)는 TRL에 의해 저대역 DDoS로 탐지되어 Blocking이 진행되어야 하는 경우 상태(Stat)가 Enable 되고 정해진 시간(Sec)동안 해당 세션의 통신을 Blocking하게 된다. 그리고 세션의 상태를 관리하는 Check부분은 현재의 세션 정보가 정상적인 통신상태인지 DDoS 상태를 체크하고 있는지 판단하여 정상적인 통신 상태일 경우 각 DDoS 필터를 Bypass하여 바로 내부의 시스템으로 패킷을 전달할 수 있도록 한다.

세션 테이블의 필드 추가에 따라 방어시스템 구현에 필요한 테이블의 변경된 주요 구조체는 [Fig. 7]와 같다.

```
struct session_table
{
    unsigned long          sip;        // 서비스 요청 Client IP
    unsigned short int     sport;      // 서비스 요청 Client 서비스 포트
    unsigned long          dip;        // 내부 서버 IP
    unsigned short int     dport;      // 내부 서버 서비스 포트
    struct TRL             trl;        // Time Rate Limit 구조
    struct BT              bt;         // Blocking Time 구조
    unsigned char          check;      // 해당 세션에 대한 DDoS 공격 탐지 완료 후 정상통신 여부 알림
                                       // 1 : 정상통신 중, 0 : DDoS 공격 탐지 중(Blocking 포함)
};

struct TRL                             // session_table.check == 0일 경우만 사용
{
    unsigned long          count;      // 세션에 대한 통신량, session_table.trl.sec != 0 까지만 합산
    unsigned char          sec;        // 모니터링 시간, 매초 1씩 감소
};

struct BT                              // session_table.check == 0일 경우만 사용
{
    unsigned char          sec;        // 세션 Blocking 시간, session_table.bt.stat == 1 이면 매초 1씩 감소
    unsigned char          stat;       // 해당 세션에 대한 Blocking 여부, 1 : Blocking
                                       // session_table.bt.stat = 0 & session_table.bt.sec = 0이면 Drop
};

unsigned long              TRL_count;  // 보안정책에서 설정한 통신량
unsigned char              TRL_sec;    // 보안정책에서 설정한 세션의 모니터링 시간
unsigned char              BT_sec;     // 보안정책에서 설정한 Blocking 시간
```

Fig. 7. View of the main structure of the session table

(2) 보안정책 UI 수정

TLF를 적용하여 저대역 DDoS 공격을 타지하기 위한 방어시스템의 보안정책 설정 화면에 Low Level Attack UI를 [Fig. 8]와 같이 추가하였다.

추가된 UI부분은 본 논문에서 제안하고 있는 TLF를 사용하기 위해 필요로하는 TRL, BT과 관련된 임계치를 얻기 위한 부분으로 추가된 부분의 주요 내용은 다음과 같다.

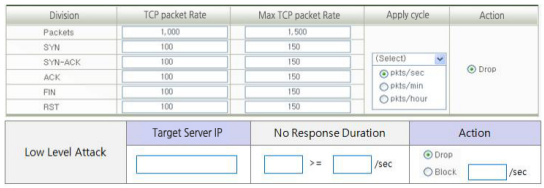

Fig. 8. Setting screen of security policy for the input of information TLF

- 세션에 대한 모니터링 시간
- 모니터링 시간 동안의 최소한의 통신량
- DDoS 공격 판단시 적용 방법(Drop, Blocking)
- Blocking의 경우 Blocking할 시간

3.3 알고리즘 동작 설명

3.3.1 세션 테이블 생성

패킷이 처음 도착하는 경우 방어시스템에서는 패킷을 처리하기 위한 세션 정보가 존재하지 않으므로 [Fig. 9]와 같은 과정을 거쳐 새로운 세션 테이블을 생성하게 된다.

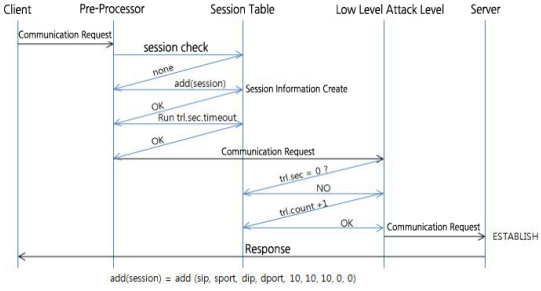

Fig. 9. The process of creating the session table

전처리부에서 세션 테이블 존재여부를 체크하여 세션 테이블이 존재하지 않는 경우 Null 값을 리턴하게 되며 전처리부에서는 필요한 정보들을 기반으로 새로운 세션 테이블을 생성한다. 이때, 세션 테이블의 TLF 필드들에 보안정책 화면에서 설정한 각각의 정보를 함께 전달하여 설정하게 된다. 세션 테이블이 정상적으로 생성되면 TRL 필드의 정해진 시간을 역카운트하는 타이머를 생성하여 종료가 될 때 까지 세션의 통신량을 체크하기 시작한다.

3.3.2 세션 모니터링 과정

세션 테이블이 정상적으로 생성된 이후에는 [Fig. 10]과 같은 과정을 통하여 세션의 통신량을 모니터링하게 된다.

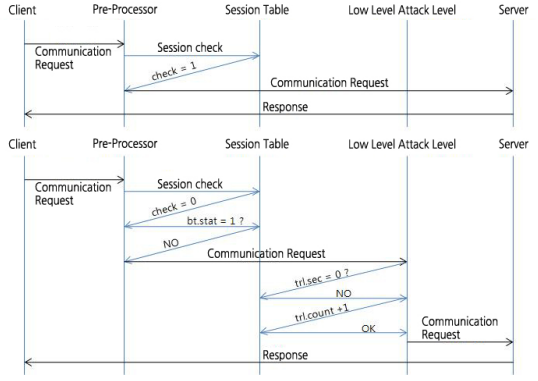

Fig. 10. Session traffic is monitored

전처리부에서는 패킷에 대한 세션 테이블 정보를 획득하고 해당 세션에 대한 상태를 파악한다. 세션의 상태가 정상적인 통신으로 인지되었다면 바로 내부 시스템으로 패킷을 전달하고, 세션의 상태가 비정상적인 경우 세션의 Blocking 설정 상태를 체크한다. 세션에 대한 모니터링이 계속 진행되고 있는 경우 Blocking 설정이 Disable 되어 있으며 패킷은 Layer 7에 있는 Low Level Attack 필터를 거치게 된다. Low Level Attack부에서는 모니터링 시간이 종료되었는지를 체크하고 패킷의 카운트를 증가 시킨다.

3.3.3 세션 모니터링이 완료된 경우

세션 모니터링 시간이 종료되면 Low Level Attack부는 [Fig. 11]과 같은 과정을 통하여 세션에 대한 상태를 변경하게 된다.

모니터링 시간이 종료되면 세션의 통신량이 보안정책에서 설정된 최소 통신량과를 비교하여 세션의 통신량이 크다면 해당 세션은 정상적인 세션으로 판단하여 세션의 상태 구분 필드를 Enable시키고 해당 세션은 모니터링 과정에서 전처리부에 의해 바로 내부 시스템으로 연결된다. 그러나, 최소 통신량보다 적거나 같은 경우 Low Level Attack부는 저대역 DDoS로 탐지하여 보안정책의 실행부에 체크된 동작을 하게 된다. 보안정책에서 Drop으로 체크되어 있는 경우(session_table.bt.stat 0 &

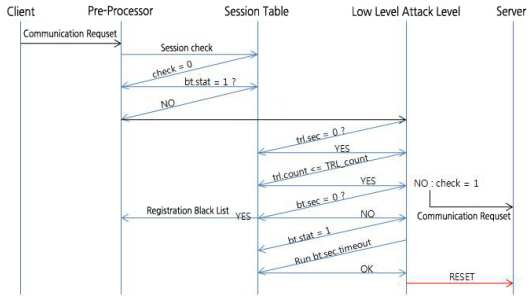

Fig. 11. Operation procedure of monitoring at the end

session_table.bt.sec =0) 세션의 정보를 Blacklist에 등록하고 세션 테이블에서 해당 세션의 모든 정보를 삭제한다. Blacklist로 등록된 세션은 일반 DDOS 방어시스템의 처리와 동일하게 전처리부에서 패킷을 Drop하여 더 이상의 통신을 할 수 없게 된다. 보안정책에서 DDoS 탐지시의 실행이 Blocking으로 되어 있는 경우 Blocking 필드를 Enable시키고 Blocking 할 시간을 역카운트하는 타이머를 실행하며, 내부 서비스에 RESET 패킷을 전달하여 연결되어있는 세션을 종료하도록 한다.

3.3.4 Blocking 동작 과정

[Fig. 12]은 DDoS 공격으로 탐지된 세션에 대해서 Blocking이 이루어지는 과정을 나타낸다.

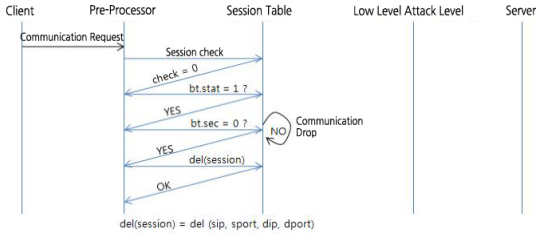

Fig. 12. Blocking Operation process

모니터링 완료 결과 DDoS 공격으로 탐지된 세션과 관련된 패킷은 정해진 Blocking 시간동안 전처리부에서 패킷을 Drop하게 된다. Blocking 시간이 종료된 경우 전처리부는 세션 테이블의 해당 정보를 모두 삭제 한다. 세션 테이블의 내용이 삭제된 이후에 수신되는 패킷에 대해서는 처음 세션 생성 과정부터 다시 시작하여 서비스를 받을 수 있다.

4. 구현 및 성능 평가

4.1 DDoS 대응 시스템 구현 환경

각 시험에 사용된 시스템의 환경은 서버PC는 OS Ubuntu Linux, ProFTP 1.3.2e, DDoS 방어시스템 OS는 전용 OS, 고대역 DDoS 방어시스템 TLF 모듈 탑재, Client PC는 MS Windows 7, 사용 Tool : Dos Command FTP, Wireshark 1.4.2, AttackPacket 2.6.1.2 을 사용하여 테스트 하였다.

4.2 시험 방법

기존 DDoS 방어시스템에서의 고대역 DDoS 공격에 대한 테스트를 진행하여 정상적으로 동작되는가를 확인한 다음 저대역 DDoS에 대한 방어가능 여부를 테스트 하였다. 이후 TLF가 적용된 모듈을 추가로 설치 적용 후 저대역 DDoS 공격에 대한 동일한 테스트 진행 하였다. 고대역 DDoS 는 공격 툴인 AttackPacket을 이용하였으며 저대역 DDoS는 DOS Command 상에서 FTP 서비스를 이용하여 다음과 같은 방법느오 테스트를 진행하였다.

- 고대역 DDoS 테스트
① DDoS 방어시스템에 탐지 및 차단 정책 설정
② 공격 툴을 이용하여 ICMP, UDP, SYN Flooding 공격 시도
③ 공격에 대한 결과 확인

- 저대역 DDoS 테스트
① DDoS 방어시스템에 탐지 및 차단 정책 설정
② FTP 서버의 MaxClient를 5개로 제한한다.
③ 3개의 저대역 DDoS 공격 Client, 4개의 정상통신 Client 준비
④ 저대역 DDoS 공격 Client를 연결 후 정상통신 Client를 순차적으로 연결
⑤ 일정 시간 경과 후 추가적인 Client의 FTP 접속 시도
⑥ FTP Client 정상 동작 확인
⑦ Client들의 연결되어 있는 모든 세션 종료
⑧ TLF를 지원하는 모듈 설치 및 적용
⑨ 저대역 DDoS 공격을 차단하기 위한 TLF 값적용
⑩ 저대역 DDoS 테스트의 ③~⑥번에 해당되는 테스트를 동일하게 진행

4.3 시험 결과

4.3.1 저대역 DDoS 테스트 결과

저대역 DDoS 공격이 TCP Three-way handshaking을 거쳐 세션을 정상적으로 연결한 상태에서 단순 세션연결 유지만 요청하며 MaxClient 상태에 도달한 서버에서 더 이상의 서비스를 제공할 수 없게 되는 상황을 만들기 위해 서버에서 접속 가능한 MaxClient의 수를 제한하여 테스트를 진행 하였다. 서버에 ProFTP 서비스를 설치하고 MaxClient는 5개로 제한하였으며 입력 기다림에 대한 Timeout은 5분(300초)으로 제한 하였다. Client는 7개의 FTP 접속을 시도한 결과 [Fig. 13]와 같이 5개 이상의 접속 시도 요청은 서버로부터 접속 거부로 이어져 정상적인 서비스를 받을 수 없었다.

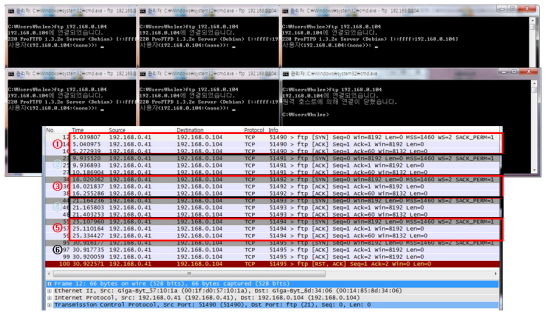

Fig. 13. Low-Level DDoS Attack Result

3개의 저대역 DDoS 공격용 Client의 경우 단순한 세션 연결만 지속하도록 하였고 이후의 Client들은 모두 정상적인 통신을 하기 위한 Client들로 세션이 연결된 이후에 디렉토리 이동 및 파일 다운로드 등에 관련된 명령어들을 실행하였다. 일정시간 경과 후 연결되어 있는 Client들과 접속이 거부되었던 Client들에서 다시 접속 시도 및 세션 연결 유지 테스트 진행 결과 모두 정상적으로 세션이 연결되어 있었다.

4.3.2 TLF 적용 후 테스트 결과

TLF는 저대역 DDoS 방어를 위해 세션의 모니터링을 위해 필요한 모니터링 시간과 최소한의 통신량에 대한 적절한 설정값을 필요로 한다. FTP 서비스 이용과 관련된 최소한의 통신량을 추출하기 위해 1개의 Client에 대

해 FTP control 포트인 서비스 21번 포트에 대해서 서버와 접속하는 각 단계별 패킷의 통신량을 [Table. 2]과 같이 테스트 하였다.

Table 2. The minimum traffic to use the FTP service

	Number of communication packets Client	Total number of packets
Communication request	3 EA	3 EA
USER Input	2 EA	5 EA
PASSWD Input	2 EA	7 EA
DIR Input	2 EA	9 EA
CD Command Input	1 EA	10 EA

파일 전송을 위해서는 세션 생성에 필요한 TCP 3-way handshaking을 시작으로 사용자의 아이디 및 패스워드입력과 필요한 파일의 위치나 이름을 확인하는 단계까지만 최소한 10개의 패킷이 Client에서 서버로 전달되어 지므로 파일을 전송하기 위해서는 이보다 많은 패킷이 전달되어야 FTP의 정상적인 통신이 이루어지고 있다고 판단 할 수 있다. 따라서 FTP 서비스에서의 모니터링시 필요한 최소 통신량의 Factor는 10으로 설정 하였다. 또한 적절한 모니터링 시간의 추출을 위해 일반적인 FTP 사용자의 경우 파일 전송 요청까지 소요되는 시간을 알아보기 위해 5명의 다른 사용자들로 테스트를 진행하여 [Table. 3]과 같은 결과를 얻었다.

Table 3. Time to file transfer requests

	Connection(sec)	File Request Start(sec)	Total
1 Time	4 sec	13 sec	17 sec
2 Time	7 sec	20 sec	27 sec
3 Time	6 sec	17 sec	23 sec
4 Time	10 sec	25 sec	35 sec
5 Time	7 sec	19 sec	26 sec
Average	7 sec	19 sec	26 sec

테스트 결과, 접속을 요청한 시간부터 파일 전송을 요청하는 시점까지 평균 약 26초 정도가 소요되고 있었다. 이는 파일전송이 필요해 FTP 서버로 접속하는 대부분의 경우 최소한 30초 이내에는 파일 전송에 관련된 모든 패킷이 지나간다고 할 수 있다. 이 결과를 바탕으로 모니터링에 필요한 시간 Factor로 30초를 추출할 수 있었다.

이러한 테스트 결과는 FTP 서비스를 이용하여 서버에 접속한 이후 바로 파일 전송을 요청하는 경우에 한하며, HTTP와 같이 세션을 연결함과 동시에 결과를 요구하는 서비스들의 경우 별도의 테스트를 통하여 적절한 Factor들을 추출하여야 한다. 테스트 결과에 따라 TLF에 필요한 Factor값으로 30초의 모니터링 시간과 10개의 최소 통신량을 TRL의 Factor값으로 설정하였으며 BT와 관련된 Factor로써 저대역 DDoS 공격으로 탐지된 경우 Drop이 아닌 Blocking을 진행하되 60초간의 Blocking Time을 적용하였다. TLF 와 관련된 Factor들을 모두 적용한 모듈을 기존 DDoS 방어시스템에 모듈 형태로 설치하고 4.3.2.1에서 진행하였던 저대역 DDoS 공격과 동일한 테스트를 진행하여 [Table. 4]와 같은 결과를 얻을 수 있었다.

Table 4. Result of the connection of test session of the waiting time for each of TLF after application

Waiting Time(sec)	Trying to connect	Normal Connection	Access denied
10	2	0	2
20	2	0	2
30	2	0	2
60	2	2	0
90	2	2	0
120	2	2	0
180	2	2	0
240	2	2	0
300	2	2	0
600	2	2	0

테스트 결과 TLF 모듈을 적용한 이후에도 DDoS 방어 시스템에서는 모니터링이 진행되는 30초 동안은 TLF를 적용하기 전의 고대역 DDoS 방어시스템과 동일하게 5개 이상의 Client들에 대해서는 서비스 거부 현상이 발생하였다. 그러나 모니터링 시간의 지난 이후부터 세션에 대한 연결 빛 통신상태를 체크한 결과 단순 연결만 되어 있던 저대역 DDoS 공격에 사용되었던 3개의 Client들에 대한 연결은 이루어지지 않고 있었으며 서비스 거부가 발생되었던 2개의 Client에서 FTP 서비스를

위한 세션 연결 및 통신이 정상적으로 이루어짐을 확인하였다.

FTP 서비스에 대한 TLF 모듈에 적용한 모니터링을 위해 필요한 시간과 최소 통신량에 대한 Factor로 30초와 10개의 설정값이 적절하게 동작되고 있음이 확인되었으며 TLF 모듈을 추가로 적용할 경우 기존의 고대역 DDoS 방어시스템으로도 저대역 DDoS 공격을 방어할 수 있음을 확인하였다.

4.3.3 테스트 결과 비교

[Table. 5]은 고대역 DDoS 방어시스템을 설치한 상태에서 TLF 모듈을 적용하기 전과 적용한 이후의 저대역 DDoS 공격에 대한 시간대별 대응 결과를 나타내고 있다.

Table 5. Results of time-of-day connection test of TLF before applying / after

Waiting Time(sec)	Attempts to connect	Before TLF Application		After TLF Application	
		Normal Connection	Access denied	Normal Connection	Access denied
10	2	0	2	0	2
20	2	0	2	0	2
30	2	0	2	0	2
60	2	0	2	2	0
90	2	0	2	2	0
120	2	0	2	2	0
180	2	0	2	2	0
240	2	0	2	2	0
300	2	2	0	2	0
600	2	2	0	2	0

TLF 모듈 적용 전에 고대역 DDoS 방어시스템으로는 저대역 DDoS 공격에 대해 어떠한 방어동작도 이루어지지 않았다. 이로 인해 서비스를 제공하는 시스템의 리소스 고갈로 서비스 불가 상태로 되었으며 서비스에서 설정된 입력 대기시간에 대한 Timeout이 설정된 경우 Timeout 이후에 저대역 DDoS 공격과 관련된 세션이 종료되고 새로운 서비스 요청을 받을 수 있었다. 반면 적절한 Factor값을 설정한 TLF 모듈을 적용하고 나 뒤에는 모니터링 시간이 종료된 후에 그동안의 통신량에따라 정상적인 통신으로 인지되지 않은 세션들을 Blocking 시켜 시스템의 리소스를 추가 확보함으로써 다른 정상적인 서비스를 요청하는 Client들의 통신을 받아들일 수 있었다. [Fig. 14]은 TLF 적용 전후에 따른 서비스를 제공하는 시스템에 접속되어 있는 Client들의 현황을 나타내고 있다.

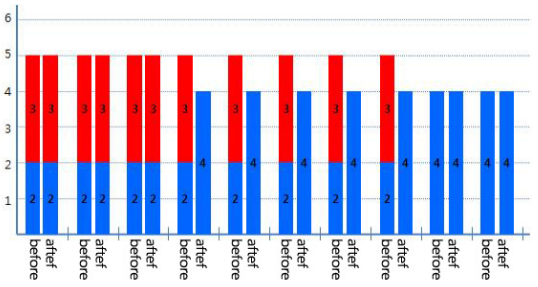

Fig. 14. Client online in another time zone of TLF application before / after

지금까지의 결과에서 볼 수 있듯이 기존의 고대역 DDoS 방어시스템에서 방어할 수 없었던 저대역 DDoS 공격에 대하여 TLF 모듈을 적용한 이후에 단순 접속되어 있는 Client들의 통신 상태를 모니터링하여 비정상적인 통신 상태에 있는 세션을 차단함으로써 서비스를 제공하는 시스템에서는 추가적인 리소스를 확보하여 정상적인 사용자들에게 서비스를 제공해 줄 수 있음을 확인하였다.

5. 결론

본 논문에서는 저대역 DDoS 공격에 대한 대응 방안으로 Time Limit Factor로서 Time Rate Limit와 Blocking Time 및 세션의 상태를 체크하는 필드들을 추가하여 저대역 DDoS 공격을 방어할 수 있는 알고리즘을 구현하였다. 또한 네트워크의 일시적인 장애로 인한 저대역 DDoS 공격으로 오탐 되었을 경우를 대비하여 공격으로 탐지된 정보에 대해 단순 Drop이 아닌 일정시간동안의 Blocking 기능을 제공하여 일정 시간이 지난 뒤 다시 재 접속이 가능하도록 하였다.

고대역 DDoS 방어시스템은 저대역 DDoS 공격으로 단순 연결된 세션들에 대해 정상적인 통신으로 인지하여 세션에 대한 차단이 불가하여 저대역 DDoS 공격을 받은 시스템은 리소스 고갈로 서비스 불가 현상이 발생하였다. 반면 TLF를 적용한 저대역 DDoS 방어시스템으로

동일한 테스트를 진행한 결과 정상적인 서비스를 요청하는 세션의 연결은 계속 유지되었지만 저대역 DDoS 공격에 의한 단순 연결만하고 있는 세션을 종료시켜 시스템의 리소스 고갈을 방지함으로써 정상적인 서비스를 제공할 수 있었다.

본 논문에서 제안한 TLF 알고리즘을 고대역 DDoS 방어시스템에 적용하게되면 고대역 및 저대역 DDoS에 대한 방어가 가능할 뿐만 아니라, 서비스를 제공하는 시스템에 모듈형태로 추가 적용을 할 경우 저대역 DDoS 공격에 대한 대처가 가능하다. 그러나, Time Limit Factor를 적용한 저대역 DDoS 알고리즘은 네트워크 환경 및 서비스의 제공 방법에 따라 많은 변수들을 가지고 있기 때문에 일률적으로 정해 질 수가 없다는 단점이 있다. 때문에 각 서비스별 운영 환경에 맞는 Factor들을 추출하여 적용하기 위해서는 각 실환경에서의 테스트 및 일정시간의 모니터링을 통하여 결정되어져야 하며 저대역 DDoS 방어를 위한 TLF 알고리즘의 원활한 동작을 위해서는 고대역 DDoS 방어시스템보다 높은 성능처리와 리소스를 필요로 한다. 시스템에 모듈형태로 적용하지 않고 전용 DDoS 방어시스템에 적용하기 위해서는 TLF를 처리하기 위한 리소스와 서비스별 관리방법등에 대한 별도의 추가적인 연구가 이루어져야 한다.

References

[1] Jong Yeop Lee, Mi sun Yoon, Hoon Lee, “Monitoring and Investigation of DoS Attack”, KNOM Review 2004

[2] Alefiya Hussain, "Experience with a Continuous Network Tracing Infrastructure", ACM SIGCOMM'05 Workshops, 2005.
DOI: http://dx.doi.org/10.1145/1080173.1080181

[3] David Dagon, CliZou, and Wenke Lee. "Modeling botnet propagation using time zones". In Proceedings of the 13th Annual Network and Distributed System Security Symposium, 2006

[4] David Dagon, Guofei Gu, Chris Lee, and Wenke Lee. “A taxonomy of botnet structures“. In Proceedings of the 23 Annual Computer Security Applications Conference (ACSAC'07), December 2007.
DOI: http://dx.doi.org/10.1109/acsac.2007.44

[5] E. Messmer. "Nugache worm kicking up a Storm", January 2008.

[6] Ki Hoon Kwon, Young Goo Han, Seok Bong Jeong, Se Hun Kim, Soo Hyung Lee, Joong Chan Na, “Fast Detection Scheme for Broadband Network Using Traffic Analysis”, KIISC, Vol14, No4, 2004

[7] Soon Hwa Hong, "Monitoring and analysis of network traffic using Load Balancing Method", Master's thesis, 2002.

[8] Les Cottrell and Connie Logg, "Throughput Time Series Patterns (Diurnal and Step Functions)", July 2004.

[9] Beak Do Woon, "Implementation for L7 DDoS Defense", August 2014.

[10] Lee Heon Jin, "A Study on The Complex Types DDoS Attacks and Protection", February 2014.

[11] Byeong-uk Lee, Cheol-woong Lee, Seung-hun Shin, Byeong-hee Roh, “Implementaion of Modeling and Simulation for DDoS Attack and Detection in Wired Tractical Netwotk Usong OPNET Cyber Effect Model”, KCI, Vol12, No2, 2016

[12] Dongwon Seo, "Probabilistic Filter Propagation and Scheduling against Distributed Denial-of-Service Attacks", February 2014.

[13] Sungmo Jung, "Global Network Security System to Prevent Cyber Attack", February 2014.

이 형 수(Hyung-Su Lee) **[정회원]**

- 1991년 2월 : 성균관대학교 전자공학과 (전자공학사)
- 2011년 2월 : 숭실대학교 정보과학대학원 정보보안학과 (공학석사)
- 2013년 3월 : 숭실대학교 대학원 컴퓨터학과 (박사수료)
- 2015년 4월 ~ 현재 : 소원네트웍스 연구소장
- 2016년 3월 ~ 현재 : 인하공업전문대학교 겸임교수

<관심분야>
네트워크 보안, 사이버보안, 정보보안

박 재 표(Jae-Pyo Park) **[종신회원]**

- 1998년 8월 : 숭실대학교 대학원 컴퓨터학과 (공학석사)
- 2004년 8월 : 숭실대학교 대학원 컴퓨터학과 (공학박사)
- 2010년 3월 ~ 현재 : 숭실대학교 정보과학대학원 교수

<관심분야>
네트워크 보안, 디지털포렌식, 정보보안