韓國컴퓨터情報學會 論文誌
第14卷 第9號, 2009. 9.

2009-14-9-3-4

DDoS TCP Syn Flooding Backscatter 분석 알고리즘

최 희 식*, 전 문 석**

DDoS TCP Syn Flooding Backscatter Analysis Algorithm

Hee-Sik Choi *, Moon-Seog Jun**

요 약

본 논문에서는 인터넷 보급으로 우리 생활 속에 급성장하여 널리 이용되고 있는 대형 포털 및 소셜 네트워크 서비스를 공격하여 개인 고객의 데이터베이스를 가로채고 웹 사이트의 정상적인 서비스를 방해하는 DDoS(Distribute Denial of Service Attack) 분산 서비스 공격에 대해 알아보고자 한다. 공격 유형중에 TCP SYN Flooding 공격은 많은 트래픽을 유발시키지 않으면서도 공격 형태가 정상적인 트랜잭션의 형태를 가지고 있으므로 공격에 대한 탐지가 쉽지 않다. 이에 대해 본 논문에서는 기존의 탐지방법은 False Alarm을 유발할 가능성을 많이 가지고 있으므로 이를 보다 정확하게 탐지하기위한 방안을 모색하고 제안하고자 하며, Backscatter 현상을 탐지하여 TCP SYN Flooding 공격을 감지하는 알고리즘을 제안하고자 한다.

Abstract

In this paper, I will discuss how the Internet has spread rapidly in our lives. Large portals and social networks experience service attacks that access personal customers' databases. This interferes with normal service through DDoS(Distribute Denial of Service Attack), which is the topic I want to discuss. Among the types of DDoS, TCP SYN Flooding attacks are rarely found because they use few traffics and its attacking type is regular transaction. The purpose of this study is to find and suggest the method for accurate detection of the attacks. Through the analysis of TCP SYN Flooding attacks, we find that these attacks cause Backscatter effect. This study is about the algorithm which detects the attacks of TCP SYN Flooding by the study of Backscatter effect.

▸ Keyword : 분산서비스공격(DDoS), 침입탐지(Intrusion Detection), 공격(Attack), Backscatter

• 제1저자 : 최희식 교신저자 : 전문석
• 투고일 : 2009. 09. 03, 심사일 : 2009. 09. 09, 게재확정일 : 2009. 09. 13.
* 숭실대학교 대학원 컴퓨터학과 ** 숭실대학교 컴퓨터학부

Ⅰ. 서 론

인터넷의 눈부신 발달은 사용자에게 네트워크를 통한 다양하고 빠른 서비스를 제공받을 수 있는 환경을 제공하였다. 이와 같이 초고속 통신망 네트워크 기반 서비스에 대한 의존도가 증가하면서 개인 및 기업에 업무적 효율성이 또한 향상되었다. 반면에 인터넷의 역기능인 외부인의 시스템 불법 침입과 개인정보 유출 및 훼손, 악성 바이러스 유포등은 날로 심각한 수준에 이르고 있다[3]. 최근에도 소셜 네트워크 사이트에 대한 피싱 공격자들이 계정을 만들고 이를 이용하여 다른 사용자 계정 정보를 탈취하는 사례와 소셜 네트워크 사이트에 침입하여 가짜 초대장을 만들어 대량 메일발송 웜을 첨부하여 개인 정보를 탈취하는등 보안 위협은 앞으로도 더욱 증가할 것이다[5].

이렇듯 DDoS 공격은 대표적인 해킹 방법으로 지금까지 여러차례 비슷한 유형의 피해들이 속출하였다. 따라서 본 논문에서는 분산 환경에서의 DDoS 공격도구인 TCP SYN Flooding의 침입패턴을 분석하여 침입탐지시스템의 Alarm들 중에서 DDoS에 대한 False Alarm을 감소시키기 위한 방안으로 Backscatter 알고리즘을 제안하여 DDoS 공격 여부를 탐지할수 있는 방법에 대해 제안하고자 한다.

Ⅱ. 관련 연구

본 장에서는 DDos 공격이 이루어지는 동작 원리와 공격의 종류 및 침입 시도 기술에 대해서 살펴본다.

2.1 DDoS 동작 원리

DDoS 공격은 인터넷의 구조적인 취약성을 악용하여 정상적인 서비스의 지연 및 마비상황을 일으키는 공격이다[24].

DDoS의 공격의 동작 원리는 TCP/IP 프로토콜의 구조적인 보안 취약점을 이용하여, 공격대상 서버로부터 접속 연결 SYN 요청에 대한 응답을 처리하지 않아, 공격하는 호스트들에게 응답을 빗나가게 하여 종속 호스트의 서버가 네트워크 트래픽의 출발지인 것처럼 공격을 한다[19]. 이 때 IP 주소에 대한 특별한 인증 절차 없이 무제한적으로 대량의 데이터 패킷을 전송시켜 네트워크를 마비시킨다. [그림 1]은 DDoS 공격의 동작 원리를 나타낸다.

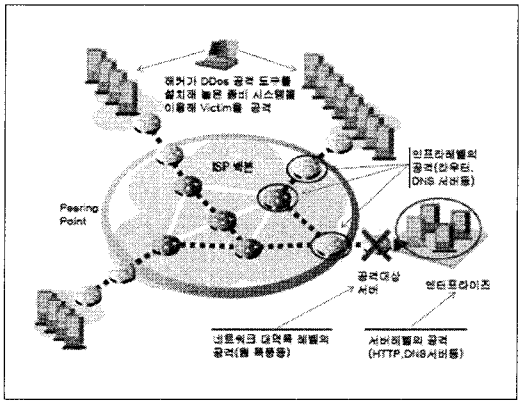

그림 1. DDoS 동작원리
Fig 1. The principles of action in DDoS

2.2 공격 Layer Structure

DDoS 공격적 구조적 특징에는 아래 [그림 2]과 같이 4개의 계층적 구조 특성을 보이고 있다.

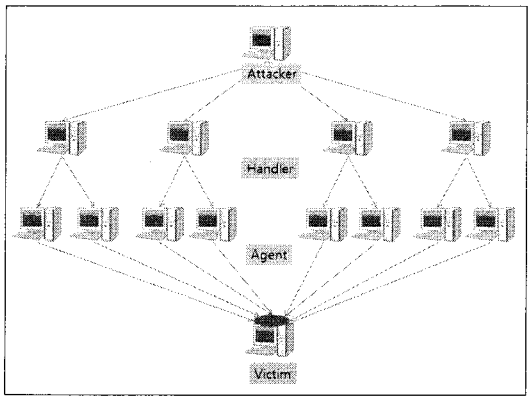

그림 2. DDoS 공격 계층 구조도
Fig 2. The structure of attack in DDos

위 [그림 2]에서 도시된 대로 Attacker는 Handler에게 DDoS 공격 툴을 설치한다. 그러면 Handler는 각 Agent들에게 공격을 수행하도록 요구한다. 명령을 전달 받은 수천~수만 개의 Agent들은 동시에 수천~수만 개의 패킷을 대상으로 대량의 데이터를 동시에 시스템에 전송하기 때문에 Victim 시스템은 과부하가 걸려 시스템이 마비되게 된다. 이 때 Attacker은 다양한 DDoS 공격 도구 중에서 하나를 선택하여 Handler에 설치하게 되며, 각 공격 도구들은 Handler와 Agent간의 독특한 통신 특성을 가지게 된다.

아래 [표 1]은 DDoS 공격 계층을 구성하고 있는 용어에 대한 설명이다.

표 1. 공격 프로세서 용어 설명 [22]

용어	설 명
Attacker	공격을 주도하는 해커 컴퓨터 (공격자)
Handler	• 공격자에서 직접 명령을 전달받아 공격을 수행하는 시스템 • 여러대의 Agent를 관리하는 Main System
Agent	공격 대상에 직접 공격을 실행시키는 시스템
Victim	공격을 당하는 대상 시스템

2.3 DDoS 공격 유형

DDoS 공격은 여러 Agent를 이용하여 동시에 피해 호스트로 Dos 공격을 행함으로써 일반적인 Dos 공격보다 강력한 파괴력을 가진 공격 형태로 다음과 같다.

2.3.1 TCP SYN Flooding 공격

TCP SYN Flooding 공격은 특정 시스템에 대한 불법적 인권한을 얻는 적극적인 방법이 아니라 네트워크와 시스템의 자원을 공격 대상으로 하는 공격 방법으로 TCP가 데이터를 보내기 전에 연결을 맺어야 하는 연결 지향형(Connection Oriented) 프로토콜을 이용한 TCP 연결의 결점을 이용한 공격으로 볼 수 있다[4]. 이 공격은 서버별로 한정되어 있는 동시 사용자를 존재하지 않는 클라이언트가 접속한 것 처럼 속이고, 또한 3-Way Hand Shaking 절차를 이용하여 연결하는 과정에서 Attacker가 Victim에 source IP address를 spoofing하여 SYN 패킷을 특정 포트로 전송하게 되면 이 포트의 대기 큐(Backlog Queue)에 오버플로우를 발생시켜 다른 사용자가 서버에서 제공하는 서비스를 이용하지 못하도록한다[15].[그림 3]

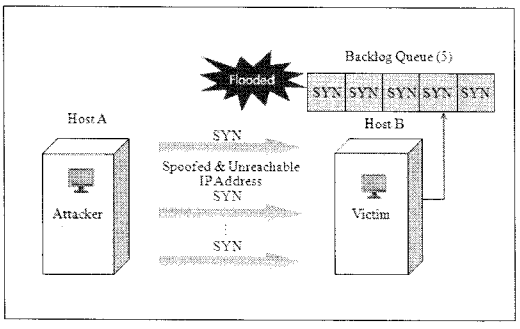

그림 3. TCP SYN Flooding
Fig 3. TCP SYN Flooding

2.3.2 TFN 공격

TFN 공격은 Trinoo 공격과 유사한 분산 툴로 많은 소스에서 하나 또는 여러 개의 목표 시스템에 대해 서비스 거부 공격을 수행하며[2], 사용하는 통신에 특정한 포트가 별도로 지정 되어 있지 않고 사용되며 항상 암호화가 되어 있다. 그렇기 때문에 포트 사용은 프로그램으로 처리되어 UDP, TCP, ICMP가 복합적으로 랜덤하게 처리하는 방식으로 결정되어 진다. 특히 TCP SYN Flooding과 UDP Flooding, ICMP Flooding, Smurf 공격을 사용하고 있으며 모든 명령은 CAST-256 알고리즘을 사용하여 암호화한다. 데몬은 인스톨시 자신의 프로세스 이름을 변경함으로써, 프로세스 모니터링을 회피하며 UDP 패킷의 헤더가 실제 UDP 패킷보다 3바이트만큼 더 크다. TCP 패킷의 헤더 길이는 항상 0으로 유지된 상태이고, Attacker가 Handler로 접근하기 위해서는 별도의 원격 접속 프로그램인 Telent등을 이용하여 Handler를 제어하여야 한다.

2.3.3 Trinoo 공격

Trinoo 공격은 많은 소스로부터 통합된 UDP Flooding 서비스 거부 공격을 유발하는데 사용되는 공격 툴이다[2]. Trinoo 공격은 몇 개의 Handler들과 많은 수의 클라이언트들로 이루어진다. 공격자는 Trinoo Handler에 접속하여 Handler에게 하나 혹은 여러 개의 IP 주소를 대상으로 서비스 거부 공격을 수행하라고 명령을 내린다. 그러면 Trinoo Master는 특정한 기간으로 하나 혹은 여러 개의 IP 주소를 공격하도록 데몬들과 통신한다. 공격을 당하는 Victim의 경우에서 볼 때에는, 이러한 공격이 어디에서 오는지 예측하기 힘들 뿐만 아니라 예방 및 추적하기도 매우 어려운 상황이다. Attacker는 Handler를 TCP 포트는 1524, 27665, 27444번을 이용하여 제어하고 Handler와 Agent간의 통신은 UDP 포트 27444번과 31335번을 이용하는 것이 특징이다[6].

2.3.4 UDP Flooding

UDP Flooding 공격은 UDP 프로토콜을 이용하여 클라이언트가 서버에 가상의 데이터를 연속적으로 보내어 용량이 초과 되도록 폭탄 공격을 시도하여 서버의 과부하 및 Network Overload를 발생시켜 정상적인 서비스를 하지 못하도록 공격한다. UDP는 비 연결성 및 비 신뢰성 프로토콜로 비교적 공격이 쉬울 뿐만 아니라 UDP의 Source address와 Source port를 Spoofing하기가 순조로워 과다한 트래픽을 Victim에 전송함으로써 Spoof되는 Victim간 네트워크를 순식간에 마비시킬 수 있게 된다[4]. 공격자가

Victim A에게 source IP address를 Victim B의 IP address로 spoofing하여 대량의 UDP 패킷을 전송하면 Victim A와 Victim B는 계속해서 서로 패킷을 주고받게 되어 두 시스템 네트워크 사이에 심한 과부하가 걸리게 된다. 이 공격은 주로 echo와 chargen 서비스를 이용한다[1].

특히 UDP Flooding 공격의 경우 MS Windows 계열의 운영체제를 사용하는 클라이언트와 비연결형(Connectionless) 패킷 전달 방식의 서비스를 제공하는 TFTP, SNMP, 실시간 인터넷 방송사들이 공격 Target 대상에 속한다[4].

2.3.5 ICMP Flooding

ICMP(Internet Control Message Protocol)는 IETF RFC792에 정의된 프로토콜로써 호스트간 혹은 호스트와 라우터간의 에러 상태 혹은 상태 변화를 알려주고 요청에 응답을 하는 기능을 담당하는 네트워크 제어 프로토콜이다[1][4].

ICMP Flooding 공격은 공격의 방향을 주로 500~1500 Byte의 패킷을 공격 Target으로 잡아서 공격 대상을 찾은 후에는 공격 대상 서버에 대용량 ICMP echo 폭탄 데이터를 전송하여 네트워크 회선 대역폭을 고갈시키고 시스템 성능을 저하시키는 공격 수법이다[4].

또한 ICMP Flooding은 유일하게 활성화된 서비스나 포트가 필요하지 않으므로 ICMP 패킷을 공격자가 비교적 쉽게 접근할수 있을 뿐만 아니라 대량의 패킷을 직접 Victim에게 전송한다. ICMP를 이용한 신종 변형된 공격 방법으로는 Smurf, Welchia worm 등이 있다.

Smurf는 공격자가 source IP address를 Victim의 IP address로 설정한 후, broadcast address로 ICMP echo request패킷을 전송하면 그 하위 모든 시스템들이 ICMP echo reply 패킷을 Victim으로 전송하게 되어 대량의 패킷들이 집중하여 네트워크 과부하를 불러 일으키는데 그 양이 너무도 엄청나서 Amplifier 공격이라고도 한다. 최근 발견된 Welchia worm은 감염 시스템에 대하여 IP address의 B클래스를 고정시키고 C클래스부터 증가시키며 ICMP 패킷을 전송하여 감염된 대상의 시스템을 찾아서 시스템의 성능을 최저로 떨어뜨리는 공격 형태이다. 공격 대상은 서버뿐만 아니라 같은 네트워크를 사용하는 대역 전체에도 통신 장애를 유발시킬 수 있다[4].

2.3.6 CC 공격

2008년 하반기에 발표된 CC(Cache-Control) 공격은 웹 부하 공격이라고도 하며, 웹 서버와 DB 서버에 과부하를 발생시키는 공격이다. 이 공격은 많은 대역폭을 전송시키는 것이 아니라 좀비 PC의 양을 늘리고 대역폭을 줄여서 GET 메시지를 서버로 전송한다. 어떻게 보면 대역폭 증가는 많이 늘지 않지만 보면 CPU가 순식간에 올라가게 된다[1].

2.4 DDoS 공격 침입 시도 기술

2.3에서 공격 유형을 살펴보았는데 대부분의 공격에는 아래와 같은 세 가지 분류의 침투 기술이 있다.

2.4.1 Foot Printing

Foot Printing은 인터넷을 이용하여 합법적으로 자유롭게 접근 할 수 있는 고객등의 정보를 수집하여 분석하는 기법을 말하며. 이를 통해 고객이름, 주민등록번호, e-Mail 주소, 전화번호, IP주소 범위, DNS Server, Mail Server, 인터넷 서비스 등을 수집한다[18].

2.4.2 Scanning

Scanning은 ping sweeps, Port scan, 운영체제 식별 등을 통해 작동중인 시스템의 종류와 인터넷을 통해 접근 가능한 시스템을 말하며, 제공하고 있는 서비스에는 어떤 것이 있는지를 탐지하는 기법이다. 이를 통해 접근 가능한 IP 주소, 침입 가능한 운영체제 종류와 유형, TCP 스택 구조 등을 수집한다[18].

2.4.3 Enumerating

Enumerating은 시스템에 연결하여 정보를 수집하는 것으로 대상 시스템의 자원 Resource, Account 정보 등을 추출하는 기법을 말하며, 이를 통해 사용자와 그룹정보, Routing table, SNMP 정보 등을 수집한다[7].

2.5 DDos 공격 분석

DDoS(Distributed Denial of Service) 공격의 두드러진 특징은 트래픽이 점차적으로 증가하여 xx초 이후에 포화상태에 도달한다는 것과 패킷 길이의 편차가 거의 없거나 동일하다는 것이다.

2.5.1 DDos 공격 단계

DDos 공격은 1시간 이내에 수천~수만대의 호스트들을 공격할 수 있는데 아래와 같은 단계로 이루어진다.

1) Scan단계 : 취약점을 찾기 위해 호스트를 검사한다.

2) Access권한 획득 단계 : Access권한을 얻을 수 있는 취약한 호스트들을 공략 한다.

3) Host에 Agent 설치 : 공격자를 은닉하고, 분산화된 공격을 위해 1차 Host들에 Agent를 설치한다.

4) 공격 진행 : 공격 명령이 Agent에게 전달되면 대상 호스트들에게 공격을 진행한다[18].

2.5.2 DDoS 공격 침입 탐지 기술

침입 탐지 기술은 시스템 로그를 분석하여 탐지하는 방법과 패킷 분석 툴을 통하여 탐지하는 방법, 포트 검색 공격을 탐지하는 방법, 알려진 침입 시도에 대한 Signature를 통해 탐지하는 오용 탐지 방법[17], 그리고 네트워크 트래픽 분석 및 패킷 내용을 검사하여 공격을 탐지하는 방법이 있다[18]. 아래 [표 2]는 침입 탐지 기술을 도표로 요약한 설명이다.[1][18]

표 2. 침입 탐지 도구

구 분	탐 지 도 구	특징 및 탐지 방법
시스템 로그 분석	syslog, message 분석	시스템 로그 파일 분석
패킷모니터링	tcpdump, snoop, netfind	패킷 모니터링 로그 분석
scan 공격 탐지	courtney, gabriel, Natas	모든 포트에 대한 접속 감시 및 기록 호스트 기반의 탐지 방법
오용 탐지	일반적인 침입 탐지시스템	알려진 침입시도 공격에 대해 탐지
Advanced 탐지	tcplogd, snort, RTSD, libnids	네트워크 패킷 분석, 접근 차단 등

2.5.3 DDoS 공격의 분석 범위

DDoS 공격의 분석 범위[표 3]에 대한 내용이다.

표 3. DDoS 공격의 분석 범위

구 분	내 용	정 보
기간 (duration)	공격이 시작된 후 종료 까지의 시간	공격 시작 시간, 종료 시간
attacks over time	일상 기간 중에서 공격이 집중하는 시간대	
inter packet delay	공격 패킷이 도착하는 패킷간 시간 간격	평균 시간, 편차
rate	단위 시간당 전송되는 공격 패킷의 수	pps (IP, port, protocol 별)
IP address	공격자 또는 공격대상 호스트의 주소	공격자주소(범위), 목적지 주소
port number	공격에 사용되는 포트 번호	포트 번호, pps
프로토콜 특성	공격시 주로 사용되는 프로토콜	protocol, pps
packet size	공격에 사용되는 패킷의 길이 분포	평균길이, 길이분포

David Moore, Geoffre M. Voelker, Stefan Savage의 "Inferring Internet Denial-of-Activity" DDoS 공격에 대한 분석 보고서[12]에 따르면, DDoS의 공격 대상의 대부분이 상용 사이트 및 대형 포털사이트(네이버,야후, 네이트온, 페이스북, MSN등)이며, 20~30%는 웹 해킹의 Target이 개인으로 조사되었다. 또한 약 5%의 공격은 라우터, Name Server와 같은 Infrastructure를 Target으로 하고 있다. 이러한 DDoS의 공격은 약 80% 정도가 사용하는 포트번호를 복수개로 하고 있으며, 일부 HTTP나 IRC의 경우가 단일 포트를 사용하였다. 이를 볼 때, 공격으로 사용되는 포트는 특정 포트에 제한하지 않고 있으므로 결국 모든 포트가 공격 대상임을 말해주고 있다.

2.5.4 DDoS 공격 프로토콜 특성 분석

조사된 DDoS 분석 보고서에 의하면 전세계 인터넷에 흐르는 DoS공격을 3주 동안 탐지한 결과, 12,000번 이상의 공격과 5,000개 이상의 공격대상이 있었음을 밝혀냈다. 이 보고서에서 밝힌, 공격에 사용된 프로토콜의 특성을 보면, TCP SYN Flooding, UDP Flooding, ICMP echo request, ICMP broadcasting(smurf)등의 형태이며, 공격의 대부분(약 90 ~ 94%)은 TCP이고, 패킷을 기준으로 할 경우 ICMP flood가 43%를 차지하였다. Flooding 공격은 주로 TCP SYN flood와 ICMP flood로 나타났으며, smurf공격이나 ping Flooding 공격에는 ICMP 트래픽의 과도한 증가가 수치로 감지되었다. Trinoo, tfn2k, UDP Flooding공격 등은 UDP 트래픽의 과도한 증가가 감지되었으며, Code Red 바이러스나 님다 바이러스의 경우에는 HTTP 트래픽의 과도한 증가가 나타났다. DDoS 공격의 프로토콜의 분포는 다음 [표 4]와 같다.

표 4. DDoS 공격 프로토콜 분포

공격 프로토콜	공격 분포		Backscatter Packet (1000단위)	
	횟 수	비율(%)	Packet 수	비율 (%)
TCP	3,902	93.5%	28,705	56.5%
UDP	99	2.4%	66	0.13%
ICMP	88	2.1%	22,020	43.3%
기 타	84	2.0%	37	0.07%

위의 [표 4]의 프로토콜 분포에서 DDoS 공격의 대부분이 TCP SYN 공격임을 알 수 있다. 다음은 공격 프로토콜에서 주로 사용되는 포트[표 5]대한 분석 자료이다.

표 5. PORT별 공격 횟수와 패킷 비교

공격 포트	공격 분포		Packet (1,000단위)	
	횟 수	비율 (%)	Packet 수	비율 (%)
multiple ports	2,740	66%	24,996	49%
Random	655	16%	1,584	31%
Other	267	6.4%	994	2%
unknown	91	2.2%	44	0.1%
HTTP (80)	94	2.3%	334	0.7
0	78	1.9%	22,007	43
IRC(6667)	114	2.7%	526	1.0
Authd(113)	34	0.8%	49	0.1
Telnet(23)	67	1.6%	252	0.5
DNS(53)	30	0.7%	39	0.1
SSH(22)	4	0.1%	2	0.0

위의 [표 5]에서 알 수 있듯이 특정 port에 집중되지 않고 여러 가지 다양한 port를 사용함을 알 수 있다. 또한 port를 0으로 공격하는 비율은 그리 많지 않지만, 전체 공격 패킷에서 차지하는 비율은 40%를 넘고 있다. 이는 ICMP flood 공격의 일종으로 network layer에서 이루어지므로 Transport layer의 port를 사용하지 않기 때문이다. 본 논문에서는 DDoS 공격의 대부분을 차지하는 TCP SYN 공격에 대한 False Alarm을 줄일 수 있는 방안을 모색하고자 한다.

III. Backscatter 검출 방안 제안

본 장에서는 이러한 TCP SYN공격에 대한 False Alarm을 줄이기 위해서 TCP SYN 공격에서 나타나는 Backscatter를 소개하고, 이러한 False Alarm을 줄일 수 있는 것으로 Backscatter에 대한 알고리즘을 제안하고자 한다.

3.1 Backscatter의 검출 방안 제안

'Backscatter'는 Vern Paxson이 만든 용어로, 그는 어떤 침입이 source address를 global multicast address로 조작함으로 multicast 연결이 끊어졌을 때 나타나는 현상을 'Backscatter 효과'라 하였다[11]. 이것을 David Moore와 Geoffre M. Voelker, Stefan Savage의 'Inferring Internet Denial-of-Service Activity'에서 네트워크 침입에 대한 분석의 기술로 사용하였다[12]. 일반적으로 네트워크 공격시에 침입자는 자신의 위치를 숨기기고 역 추적을 하지 못하도록, 패킷 source address를 바꾼 후에 공격을 시도한다. 공격에 사용되는 공격 툴에는 Shaft, TFN, TFN2k, Trinoo등이 있는데, 이들은 모두 source address를 바꾸는 방법들을 사용한다. 이렇기 때문에 공격자의 IP 주소가 Random하게 수시로 바뀌어서 전송되므로 공격을 당한 Victim 시스템들에게는 바뀌어진 IP주소로 패킷이 전송되게 된다. 이것이 바로 'Backscatter'라는 현상[그림 4] 이다.

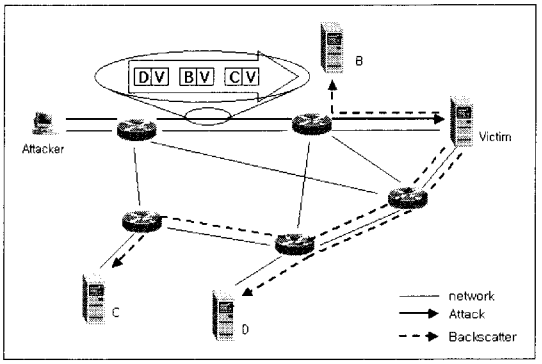

그림 4. Backscatter 현상
Fig 4. The actual state of Backscatter

[그림 4]에서 Attacker는 C, B, D라고 이름 붙여진 조작된 주소(source address)를 사용해서 희생자 V를 향해 일련의 SYN 패킷들을 보낸다. 이 패킷을 받자마자 시스템 V는 SYN/ACK 패킷을 조작된 IP 주소들(C, B, D)로 보낸다. 이렇게 네트워크 침입은 시스템 V를 목표로 공격이 이루어졌으나, 시스템 V의 정상적인 응답 메커니즘에 의해 네트워크 침입과 관련 없는 C, B, D 시스템들에게 일종의 파편처럼 패킷이 전송되는 현상을 Backscatter라 한다.[8] 또한, 패킷마다 불규칙적인 source address와 신뢰할 만한 전송과 공격시에 모든 패킷에 대해서 하나의 응답이 발생한다는 것을 고려해 볼 때, 특정 시스템이 인터넷에서 Victim으로 부터 적어도 하나의 응답 패킷을 받을 가능성은 m패킷의 공격 동안에 $E(X) = \frac{nm}{2^{32}}$ 이다. [8]

최근에는 KISA의 RTSD와 같은 침입탐지 시스템에서도 TCP SYN 공격을 탐지하는 방법으로 이러한 Backscatter가 발생하는 지를 포함하고 있다[16]. 그러나 탐지오류는 Backscatter 탐지 부분에도 발생할 수 있다. 즉, Backscatter가 아닌데 Backscatter로 여기는 경우와 Backscatter인데 탐지하Back는 경우이다.

본 논문에서는 이러한 Backscatter가 발생하였을 때, 감시를 위한 시스템 부하를 줄이고, 탐지오류율이 적은 Backscatter 탐지 방안을 제안한다.

3.2 Backscatter 알고리즘

기존의 탐지 tool들이 적용하는 침입 시도 탐지 알고리즘에 TCP SYN Attack 탐지 알고리즘을 추가한 것이다. TCP 패킷이고 flag bit가 SYN이라면, Destination Host의 IP address가 동일한 패킷이 일정시간에 임계치 이상 발생할 경우 SYN 공격으로 판단한다. 그러나 모든 패킷에 대해 SYN flag를 검사하고, 이들의 host 정보를 모두 HostList에 저장하여, connect한 값을 탐지 기준으로 하기에 다음 알고리즘의 line 30~36와 같이 처리용량이 넘어설 경우의 대응 체계가 필요하며, 이는 감시를 위해 많은 자원을 필요로 하게 된다. 또한 탐지 기준이 정상 트랜잭션 여부가 아니라 대상 호스트에 대한 SYN 패킷의 횟수를 기준으로 정하고 있어서 정상 트랜잭션에 대해서는 False Alarm이 발생할 수 있다.

3.3.1 Backscatter 탐지 기존 알고리즘[18]

```
1:  DetectSatus CheckNormalAttack(CPacket
*crntPacket)
2:  {
3:  Host* destHost;
4:  Time crntTime;
5:
6:  if (IsTcpSynPacket(crntPacket)
7:  { // 패킷이 Tcp 프로토콜인지와 flag bit가 SYN 인
가?
8:  if (IsThereInSourceHostList(crntPacket))
9:  { // 패킷의 호스트 리스트에서 현재 패킷을 받는 목적
     호스트 정보를 가져온다.
10:                          destHost          =
getHostFromDestHostList(crntPacket);
11:  crntTime = GetCurrentTime();
12:  // 현재 패킷이 전송된 시간이 시간범위내에 존재하는
       가?
13:  if ((crntTime - destHost->theLastUpdateTime)
14:       〈 Normal_Time_Threshold)
15: { // 목적 호스트가 현재 패킷과 동일한 목적 호스트를
     갖는가?
1                    6                    :
if(!IsThereSamePacketInHost(destHost,crntPacket))
17: { // 목적 호스트에게로 패키전송 횟수가 임계치를 넘
       어섰는가?
18:  if (destHost->Connection >SCAN_MAX)
19:  return DetecNormalAttack; // 공격 탐지
20:  destHost->Connection++; // 패킷전송횟수 증가
21:  }
22:  // 호스트 정보 수정
23:   UpdateHost(destHost, crntTime, crntPacket);

24:  }
25:  else
26:  { // 호스트 정보 초기화
27:  InitHost(destHost, crntPacket);
28:  }
29:  }
30:  else
31:  { // 패킷의 호스트 리스트 용량이 그 한계를 초과하
     였다면
32:  // 가장 오래된 호스트 정보를 제거함
33:  if (IsDestHostListFull())
     RemoveTheOldestFromDestHostList();
34:  destHost = MakeNewHost(crntPacket); // 새로
     운 호스트 정보생성
35:  AddHostToDestHostList(destHost); // 생성된 호
   스트  정보추가
36:  }
37:  }
38:  return Normal;
39:  }
```

3.3.2 Backscatter 탐지 수정 알고리즘

```
1: DetectSatus CheckTcpSynAttack(CPacket
*crntPacket
2:  {
3:  Host* pHost;
4:  Time crntTime;
5:  if (IsTcpSynAckPacket(crntPacket)
6:   { // 패킷이 Tcp 프로토콜인지와 flag bit가 SYN
     ACK 인가?
7:
8:   // 패킷의 srurce host, destination host, port
       number가 종일한 host 정보를 가져온다.
9:  pHost = getHostFromHostList(crntPacket);
10: if (pHost == NULL)
11: { // 일치하는 host정보가 없다면, host list에 추가한
다.
12:  addHostToHostList(crntPacket);
13:  } else
14:  { // 일치하는 host정보가 있다면, 패킷 발생 횟수가
       임계치를 넘어서는가?
15:  if (pHost->nSynAckCount 〉 MAC_SYNACK)
16:  {
17:  // 공격 탐지
18:  // 다음 탐지를 위해 해당 값을 초기화 한다.
19:  pHost->nSynAckCount = 0;
```

```
20:  return DetectBacksactter;
21:  }
22:  pHost->nSynAckCount++;
23:  }
24:  } else if ((IsTcpAckPacket(crntPacket)
25:  { // 패킷이 Tcp 프로토콜인지와 flag bit가 ACK 인가?
26:  // ACK 패킷이므로 역산한 srurce host, destination host, port number가
27: // 동일한 host 정보를 가져온다.
28: pHost = getHostFromHostList(crntPacket, DEF_REVERSE);
29: if (pHost != NULL)
30: { // SYN ACK에 상응하는 ACK가 왔으므로 정상 트랜잭션이다.
31:  // 따라서 count값을 초기화 한다.
32:  pHost->nSynAckCount = 0;
33:  }
34:  } // end of if (IsTcpSynAck, IsTcpAck)
35:
36:  crntTime = GetCurrentTime();
37:  // 현재 패킷이 전송된 시간이 시간범위를 초과하는가?
38:  if ((crntTime - oldTime) > Normal_Time_Threshold)
39:  { // 모든 호스트 정보 초기화
40:  InitHost();
41:  }
42:  oldTime = crntTime;
43:  return Normal;
44:  }
```

다음은 위에서 살펴 본 TCP SYN Attack 탐지 알고리즘을 수정하여 Backscatter 탐지를 위한 알고리즘이다. 패킷 중에서 SYN ACK와 ACK인것 만을 분석함으로 모든 패킷에 대해서 IP Address를 처리하지 않는다. 먼저 SYN ACK 패킷에 대해 src_ip, dest_ip, port number를 얻어서 HostList에 저장하고 이에 상응하는 ACK 패킷이 오면 SynAckCount를 0으로 초기화하고, 오지 않으면 SYN ACK 패킷의 srcHost를 기준으로 한 SynAckCount를 증가시키고 이것이 일정시간 이내에서 임계값을 초과하게 되면, Backscatter 발생으로 판단한다. Line 36~41에서와 같이 일정시간이 지나게 되면 SynAckCount 값을 0으로 초기화 한다. 일반적인 Ethernet local segment의 특성으로 SYN ACK 패킷의 src_ip의 개수는 보통 최대 255까지 이므로 이에 대한 처리모듈을 간단하게 할 수 있다. 즉, line 38~41과 같이 srcHostIP는 그대로 두고 DesIP, port, count만을 초기화 시킨다.

3.3 시스템 구조

시스템은 Agent, Server, Client로 구성되는 Three Tier Architecture(3단계 아키텍처)를 가진다. Agent는 네트워크 패킷을 수집하여 1차 분석하는 기능을 담당하고, Server는 여러 Agent들로부터 수집/분석된 데이터를 2차 분석하여 종합적인 판단을 하는 일과 Client로부터 요청 받은 사항들을 처리하는 일을 한다. 마지막으로 Client는 GUI를 통해 사용자와의 Interface를 담당한다. 다음 [그림 5]은 전체 시스템의 구조이다.

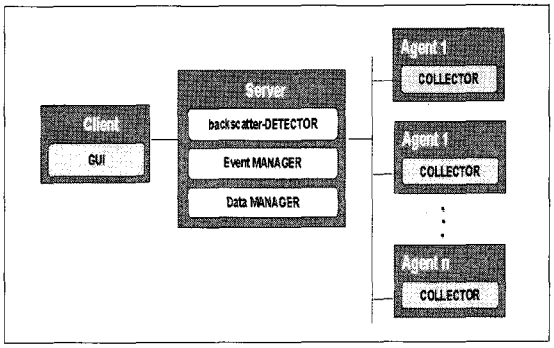

그림 5. 전체 시스템 구조
Fig 5. The structure of the whole system

이러한 구조를 통해 본 논문에서 제안한 Backscatter 탐지 알고리즘으로 실시간 트래픽 상황[그림 6]과 TCP, UDP, 포트, MAC과 IP 정보 등을 분석하게 된다. 일반적인 IP 정보와 침입탐지 후에 치명적인 공격에 대해서는 각기 다른 색으로 분류하여 차단된 패킷의 IP 자료를 실시간으로 모니터링하여 그 자료의 MAC 주소까지 파악[23]하여 보여준다.

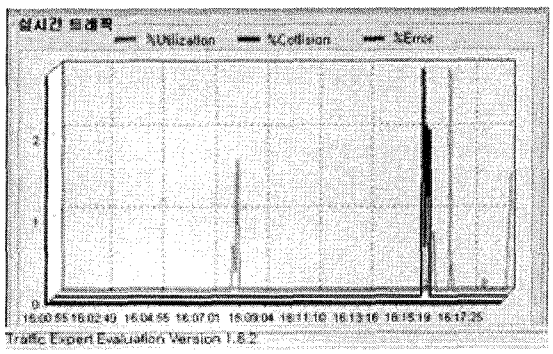

그림 6. 실시간 트래픽 상황
Fig 6. Traffic situation in real time

[그림 7]은 패킷에 대한 상세 분석으로 Backscatter 현상이 어느 시스템에서 발생하였는지를 알려주고 있다. 3초동안 발생한 SYN ACK와 상승하는 ACK가 발생한 비율을 나타내고 있으며 해당 Segment의 N/W health의 한 요소인 Packet error p%)을 보여주고 있다. 임계치값 이하인 경우 Backscatter로 판정하게 된다.

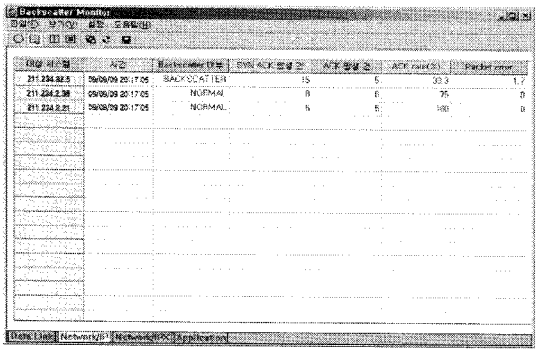

그림 7. Backscatter 현상 탐지
Fig 7. The present state detection of Backscatter

IV. 실험 및 분석

4.1 실험 방법

TCP SYN Flooding 공격으로 발생하는 Backscatter 현상을 탐지하는 실험을 위해 DAPRA에서의 IA(Information Assurance) Program에서의 실험 방법[9]을 참조하였으며 테스트 환경은 다음과 같다.

- 공격 호스트 : wowlinux 8.x
- 대상 호스트 : windows XP Home edition
- Backscatter 탐지 시스템 : windows 2003 professional
- Network Monitoring Tool : T.E. (Traffic Expert)
- 공격 Tool : Synk 4, sniffer
- 침입 시스템 (A)
- 대상 호스트 (V)
- 이웃 시스템 (O1, O2, O3)
- Backscatter 탐지 시스템 (M)

침입시스템(A)는 TCP SYN Flooding 공격 tool이 설치되어 동작하게 된다. 대상 호스트(V)로 SYN 패킷을 보내게 되며, 이 때 source IP address는 이웃 시스템(O1, O2, O3) 으로 spoofing되어 가addres 대상 호스트(V) 는 A렉 spoo의해 DoS상태로 향하게 되며 이러는 도중 Backscatter 현상이 발생하여 SYN ACK패킷을 이웃 시스템(O1, O2, O3)로 보내게 dres탐지 시스템(M)은 이러한 일련렉 sp 과정äe 며 이러는고 있다가 SYN ACKoo상응하는 ACK가 발생하지 않는 비율이 임계치과정넘게 되면 '침입 탐지'로 판단 addres 이러한 실험 방법e 적용하 가위해 실험 구성도[그림 8]를 다음과 같이 구축하였다.

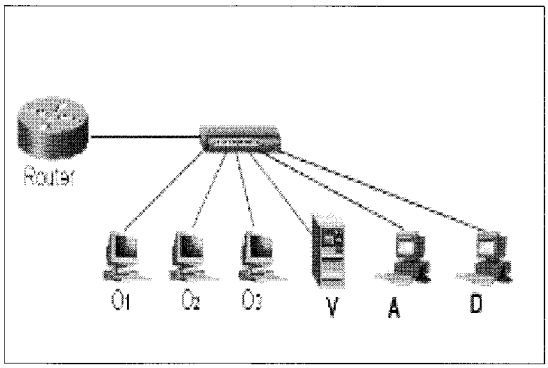

그림 8. 실험 구성도
Fig 8. The Figure of experimentation

먼저 정상 트랜잭션이 이루어지고 있는 network를 분석하여 TCP SYN 공격이 이루어지고 있을 때의 결과와 비교 자료로 사용한다. 다음으로 TCP SYN 공격이 이루어지고 있을 때의 Backscatter를 탐지하는 것이다.

4.2 실험 결과 및 분석

Interval time을 3초로 하였으며, 임계치는 ACK rate 70%로 하였다.

다음은 일반적인 상태에서의 트래픽 분석(Network Layer)[표 6]과 Backscatter 탐지 내용[표 7]이다.

표 6. 일반적인 상태에서의 트래픽 분석

Protocol	Packets	Packets(%)	Bytes	Bytes(%)
UDP	18	64.3	1228	70.2
ICMP(PING)	2	7.1	140	8
TCP	4	14.3	238	13.6
IGMP	3	10.7	84	4.8
IGRP	1	3.6	60	3.4
합 계	28	100	1750	100

표 7. Backscatter 탐지

대상시스템	시 간	Backscatter 여부	SYN ACK 발생	ACK 발생 건	ACK Rate (%)	Packet Error
150.23.25.5	02:53:44	NORMAL	1	1	100	0

특별한 데이터 전송이 거의 없는 일반적인 상태에서의 프로토콜 분포이다. 패킷 수집은 3초 동안에 이루어진 것이며, 총 28개의 패킷이 발생했다. 이 중에서 64.3%가 UDP를 사용하고 있고, TCP는 14.3%가 사용되고 있다. 이러한 상태에서 TCP SYN ACK와 ACK가 발생하였고 정상적인 TCP connection임을 알 수 있다.

다음은 TCP SYN 공격이 이루어지고 있을 때의 트래픽 분석[표 8]과 Backscatter 탐지 내용[표 9]이다.

총 76의 패킷 중에서 TCP 패킷은 80%로 정상적인 상태에서는 14.3%를 사용하던 것과 현저한 차이[그림 9]를 보이고 있다. 또한 공격 대상 시스템인 150.23.25.5의 경우 ACK Rate가 1.8%로 SYN ACK에 대한 ACK가 발생하지 않은 비율이 98.2%나 됨을 알 수 있다. 이 경우, ACK 발생률이 70%이하이므로 Backscatter Alarm[그림 10]이 발생된다.

표 8. TCP SYN 공격이 이루어지고 있을 때 트래픽 분석

Protocol	Packets	Packets(%)	Bytes	Bytes(%)
TCP	61	80.3	3770	80.3
UDP	10	13.2	606	12.9
ICMP(PING)	3	4.0	196	4.2
IGRP	2	2.6	120	2.6
합 계	76	100	4692	100

표 9. Backscatter 탐지 내용

대상시스템	시 간	Backscatter 여부	SYN ACK 발생	ACK 발생 건	ACK Rate(%)	Packet Error
150.23.25.5	02:53:44	Backscatter	54	1	1.8	0
150.23.25.89	02:57:05	NORMAL	1	1	100	0

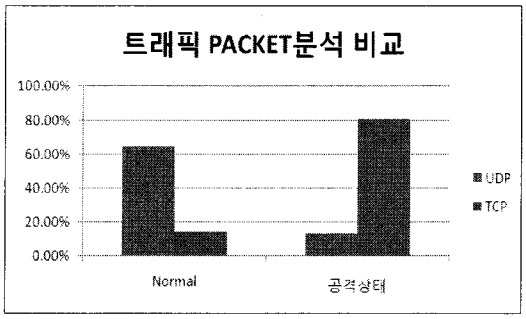

그림 9. 일반/공격 트래픽 packet 비교
Fig 9. Comparison of normal and attacked packet detection

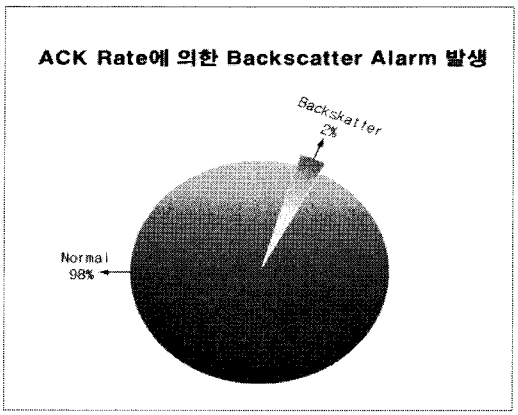

그림 10. Backscatter Alarm 발생
Fig 10. The occurrence of Backscatter Alarm

위의 내용에서 볼 수 있듯이 Source IP가 동일한 SYN ACK 패킷 발생과 이에 대한 ACK의 발생이 임계치 (70% 이하)에 도달하지 않을 경우 Backscatter로 판정 지을 수 있다는 조건을 적용하였다. 그러나 Backscatter로 판단할 임계값을 설정하는 부분에서 SYN ACK의 발생 횟수에 대한 임계값을 단지 0 이상일 경우로 하였다. 이 경우라면 1개의 SYN ACK가 발생하였고 이에 대한 ACK가 없으면 Backscatter로 판정할 것이다. 그러나 이것은 Backscatter로 하기엔 너무 적은 값이다. 따라서 실제 SYN Flooding공격에 대한 좀 더 심도 깊은 분석이 필요하며, 적절한 SYN ACK 발생 건을 임계치 값으로 적용해야 할 것이다. 또한 ACK 발생 비율에 대한 임계값을 너무 작게 하거나 크게 설정하면, Backscatter 탐지를 제대로 하지 못하게 된다. 따라서 실제 네트워크에서의 패킷 에러와 작은 오차등을 감안하여 ACK 발생 비율에 대한 임계값을 설정해야 할 것이다.

V. 결 론

본 논문에서 제안하는 방법은 SYN 패킷의 발생을 점검하지 않고 SYN ACK에 대한 ACK의 발생 비율을 감시하는 것으로, 기존의 방식보다 시스템 부하를 적게 하고 비정상 트랜잭션 여부에 초점을 주기에 탐지오류가 보다 적을 것으로 예상된다. 기존의 방법은 일정한 시간 동안에 동일 ip가 source address로 되어서 여러 destination address로 전송되는 패킷이 일정 개수 이상 발생하면 Backscatter로 간주하였다. 좀 더 개선된 방안으로 이러한 패킷의 TCP flag 설정이 SYN ACK인지를 확인하는 것이다. 그러나 모든 패킷에 대해 SYN flag를 검사하고, 이들의 host 정보를 모두 HostList에 저장하여 connect한 값을 탐지 기준으로 하기 때문에 처리용량이 넘어설 경우의 대응 체계가 필요하며, 이 같은 감시를 위해 많은 자원이 필요하게 된다. 또한 탐지 기준이 정상 트랜잭션 여부가 아니라 대상 호스트에 대한 SYN 패킷의 횟수를 기준으로 하기에 정상 트랜잭션에 대해서 False Alarm이 발생할 수 있다. 이러한 문제점을 최소화하고 TCP SYN Flooding 공격을 탐지하는데 있어 False Alarm을 줄일 수 있는 방안으로, 네트워크 패킷 분석을 통해 TCP 3-way handshaking의 SYN ACK와 ACK를 감시하는 방법을 제안한다.

본 논문에서 제시한 TCP SYN Flooding Backscatter 알고리즘을 적용하게 되면 탐지시스템의 리소스부하 및 탐지오류를 어느 정도 줄일 수 있게 될 것이다. 그러나, SYN 공격에 의해 ACK가 발생하지 않는 경우와 네트워크상에서의 일반적인 패킷 유실에 의해 ACK가 발생하지 않는 두 가지의 경우가 발생할 수 있으므로 실제 트래픽을 분석하여 적절한 임계치 값을 찾는 연구와, Backscatter으로 판정할 또 다른 방안을 찾는 연구가 계속 진행되어야 할 것이다.

참고문헌

[1] SK Inforsec, "UDP Flooding Attack 공격과 방어", *http://mss.skinfosec.co.kr/docs/wp-content/uploads/2007/04/udpflood_attack_and_defense_pub.pdf*

[2] ITKA, "DDos 공격툴", *httP://http://itka.kr/zbxe/?mid=secprot&sort_index=readed_count&order_type=des&category=906&document_srl=2114*

[3] ITFIND, "2009년 상반기 보안 위협", *http://www.itfind.or.kr//WZIN/jugidong/1412/file26183-141205.pdf*

[4] 보안뉴스, "최신 DDoS 공격 동향과 대응 방안", *http://www.boannews.com/media/view.asp?page=1&idx=16133*

[5] Holik in everything, *http://holik.org/?p=252*

[6] CERT, "Distributed Denial of Service Tools", *http://www.cert.org/incident_notes/IN-99-07.html*

[7] Joel Scambay, Stuart Mcclure, George Kurtz, *"Hacking Exposed : Network Security Secrets & Solutions, Second Edition"*, Osborne, 2001.

[8] Karanjit Siyan, "Inside TCP/IP", 1997.

[9] Sara Kaufman, Stephen Ying, "DARPA Information Assurance Program Experimental Confirmation-DDoS"

[10] Fydor, "The Art of Port Scanning", Phrack Magazine, Volume 7 Issue 51, 1997

[11] Vern Paxson. "Personal Communication", January 2001

[12] David Moore, Geoffre M. Voelker, Stefan Savage, "Inferring Internet Denial-of-Service Activity", 2001

[13] 박재홍, "네트워크 해킹과 보안", 글로벌출판사, 2003

[14] 양대일, "정보보안 개론과 실습", 한빛미디어(주), 2005

[15] 구민정, 오창석, "IPv6환경에서 DDoS 침입탐지", 한국컴퓨터정보학회, 제 11권, 제 6호, 186쪽, 2006년 12월.

[16] 한국정보진흥원, "악성 Bot 특성 분석을 통한 탐지 및 대응책", 2004년

[17] 오상현, 이원석, "패킷간 연관 관계를 이용한 네트워크 비정상행위 탐지", 정보보호학회논문지, 제12권, 제 5호, 64쪽, 2002년 10월.

[18] 유일선, "네트워크 취약점 검색공격에 대한 개선된 탐지 시스템", 건국대학교 박사학위논문, 2001년 12월.

[19] 박대우, 서정만, "TCP/IP 공격데 대안 보안방법 연구", 한국컴퓨터정보학회", 제 10권, 제 5호, 219쪽, 2005년 11월.

[20] 유대성, 오창석, "공격 탐지를 위한 트래픽 수집 및 분석 알고리즘", 한국콘텐츠학회논문지, 제 4권, 제 4호, 38쪽, 2004년 11월.

[21] 김미혜, "MIB 정보화 패킷 분석을 통한 DDoS 공격 탐지", 한국콘텐츠학회 논문지, 제 4권, 제 1호, 49-50쪽, 2004년 1월.

[22] 이근수, 박지현, 장진용, 송주석, 유동영, "DDoS 공격에 대한 탐지 및 추적 시스템 제안", 한국정보보호학회", 제11권, 제 1호, 39-40쪽, 2001년.
[23] 천우성, 박대우, "DoS공격에 대한 N-IDS 탐지 및 패킷 분석 연구", 한국컴퓨터정보학회", 제 13권, 제 5호, 222쪽, 2008년 11월.
[24] 이원호, 한군희, 서정택, "네트워크 패킷 기반 DDoS 공격 탐지 시스템 설계", 한국산학기술학회 춘계학술발표논문집, 155쪽, 2004년.

저 자 소 개

최 희 식

2006년 숭실대학교 산업정보대학원 (공학석사)
2009년 숭실대학교 대학원 컴퓨터학과 박사과정
2007년 숭실대학교 전자계산원 출강
2007년 삼육대학교 출강
2008년 경원대학교, 경민대학교 출강
관심분야: DRM, 유비쿼터스, RFID, USN, 인터넷보안

전 문 석

1981년 숭실대학교 전자계산학과 (공학사)
1986년 University of Maryland Computer Science(공학석사)
1989년 University of Maryland Computer Science(공학박사)
1989년 Morgan State University 조교수
현 재 숭실대학교 정교수
관심분야: 정보보안, 전자상거래보안, 인터넷보안, 멀티미디어 보안