검증된 IP 테이블을 사용한 통계 기반 DDoS 대응 시스템

박 필 용[†] · 홍 충 선[††] · 최 상 현[†††]

요 약

DDoS는 네트워크나 개인 호스트를 위협하는 대표적인 공격 트래픽이다. DDoS 공격은 특정한 패턴을 가지고 있지 않기 때문에 탐지가 어려울 뿐 아니라, TNF2K와 같은 간단한 도구로 공격이 가능하여, 현재 추진 중인 BcN 환경에서도 그 심각성이 초래될 수 있다. 이러한 DDoS를 탐지하기 위한 메커니즘이나 알고리즘은 많이 개발되었다. 하지만 DDoS의 근원지를 판별하고 대응하는 것이 아닌, 단지 방어 지점에서 전체 한계치를 낮추거나 리키버킷처럼 수용 능력 이상의 패킷을 폐기하는 방법으로 네트워크나 개인 호스트를 보호한다. 무분별하게 전체 트래픽을 줄이는 것은 네트워크의 자원을 고갈 시키지는 않지만, 정상적인 클라이언트가 공격당하고 있는 호스트에 연결을 할 수가 없다. 이를 위해 여러 단계의 테스트를 통해 합법적인 검증 IP 테이블을 만들고, 검증 IP 테이블에 있는 소스 IP를 제외한 나머지 트래픽을 차단한다면, DDoS 공격에 대해서 대응을 하면서 정상적인 클라이언트의 연결을 보호 할 수 있다. 제안된 메커니즘을 Linux Zebra라우터환경에서 구현되었다.

키워드 : DDoS 공격, IP 테이블, IDS

A Statistic-based Response System against DDoS Using Legitimated IP Table

Park, Pilyong[†] · Hong, Choong Seon[††] · Choi, Sanghyun[†††]

ABSTRACT

DDoS (Distributed Denial of Service) attack is a critical threat to current Internet. To solve the detection and response of DDoS attack on BcN, we have investigated detection algorithms of DDoS and implemented anomaly detection modules. Recently too many technologies of the detection and prevention have developed, but it is difficult that the IDS distinguishes normal traffic from the DDoS attack. Therefore, when the DDoS attack is detected by the IDS, the firewall just discards all over-bounded traffic for a victim or absolutely decreases the threshold of the router. That is just only a method for preventing the DDoS attack. This paper proposed the mechanism of response for the legitimated clients to be protected. Then, we have designed and implemented the statistic based system that has the automated detection and response functionality against DDoS on Linux Zebra router environment.

Key Words : DDoS(Distributed Denal of Service), IP Table, IDS(Intrusion Detection System)

1. 서 론

지난 몇 년간 초고속 인터넷 보급이 급속도로 성장하면서 전자 상거래, 인터넷 뱅킹, 전자 정부 등 일상생활이 인터넷 속으로 이동하고 있다. 하지만 이러한 편리한 인터넷 사용과 더불어 인터넷을 위협하는 공격들도 많이 발생하고 있다. 인터넷 보급이 활성화되기 이전의 공격 형태는 해킹과 같이 전문가만이 할 수 있고 특정 한 호스트에게만 피해를 주는 것이 대부분이었다.

하지만 최근 인터넷 침해형태는 간단한 도구로서 불특정 다수의 호스트에 공격을 할 수 있고, 서비스 거부와 같은 치명적인 피해를 준다는 것이 특징이다. 또한 여러 망들을 통합한 BcN(Broadband Convergence Network)환경에서 이러한 공격은 더욱더 심각한 피해를 초래할 수도 있다. DDoS(Distributed Denial of Service)[1]는 2000년 2월 야후, 아마존과 같은 인터넷 포털 사이트에 심각한 피해를 주면서, 주요 공격 트래픽으로 부상했다. 그리고 전 세계 인터넷 트래픽을 관장하는 미국 내 13개 루트 서버가 DDoS 공격을 받아 그 중 9대가 일시적으로 정상 작동이 불가능한 사례가 발생했으며, 공격의 진원지로 인터넷 사용자 수와 초고속 인터넷 보급 율이 높은 우리나라와 미국이 제기되기도 했다[2]. DDoS 공격 특징은 네트워크나 개인 호스트의 리소스를 고갈 시켜, 공격회생 네트워크나 호스트에 접속할 수 없도록 한다. DDoS의 특징은 flash crowd [3]와 같은 정상적인 인터넷 사용 트래픽과 구별이 되지 않고, 명확한 패턴이 없기 때문에 탐지하기가 어렵다. 또한 공격자가 좀비와 같은 감염된 호스트를 통해서 간접 공격을 하기 때문

※ 본 연구는 정보통신부 ITRC 및 NCA 지원으로 수행되었음.
† 준 회 원 : 경희대학교
†† 종신회원 : 경희대학교 컴퓨터공학과 교수
††† 준 회 원 : 한국전산원 선임연구원
논문접수 : 2005년 7월 21일, 심사완료 : 2005년 9월 21일

에 공격 시작지를 찾았다고 해도 실질적인 공격자는 찾을 수가 없다. 이러한 특징 때문에 DDoS에 대한 연구가 많이 이루어졌지만 아직까지도 명확한 대응 방법은 개발되지 않고 있다. 대부분의 DDoS 대응 시스템은 DDoS 방어지점에 전체적으로 트래픽을 감소시켜 네트워크를 보호하는 것이 대응책이다[4]. 하지만 정상적인 클라이언트의 입장에서 보면 DDoS에 대응하는 방법 또한 서비스 거부공격과 다를 바가 없는 상태이다. 이런 무분별한 차단이 아닌, 미리 준비된 대처 방법을 가지고 선별적인 차단이 필요하다. 이를 위해서 정상적일 때의 트래픽 측정을 하여 그 자료를 바탕으로 통계적인 수치를 내고, 여러 단계의 테스트를 통해 합법적인 검증 IP 테이블을 만든 다음, 검증 IP 테이블에 있는 소스 IP를 제외한 나머지 트래픽을 차단한다면, DDoS 공격에 대해서 대응을 하면서 정상적인 클라이언트s의 연결을 보호 할 수 있다.

본 논문의 구성을 2장에서는 DDoS 공격의 형태에 대해서 살펴본다. 3장은 이러한 DDoS를 탐지 할 수 있는 알고리즘에 대해서 살펴본다. 4장은 선별적 대응을 할 수 있는 검증된 IP 테이블에 대한 개요를 살펴본다. 5장은 DDoS 탐지와 대응을 할 수 있는 시스템에 대한 개요를 살펴본다. 마지막으로 6장에서는 실험 방법 및 결과를 제시하였으며 마지막 7장에서는 결론으로 마무리 한다.

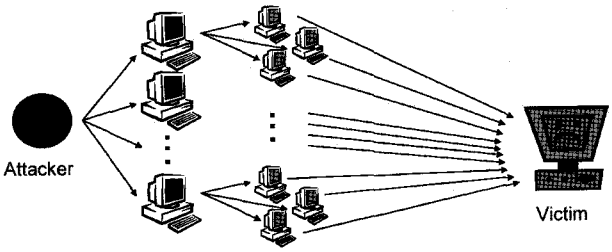
2. DDoS 공격

2.1 DoS 공격 유형

DoS도 사용자들에게 피해를 준다는 점에서 크래킹의 한 종류라고 할 수 있지만, 요즘 인터넷으로 배포되는 공격 프로그램을 이용하면 전문 지식이 없어도 공격이 가능하기 때문에 그 방법이 비교적 간단하지만 그 피해 범위와 정도는 매우 광범위하다. DoS의 주요 공격 대상은 시각적인 서비스를 하는 웹서버나 라우터, 네트워크 같은 기반 시설이다. DoS는 한 사용자가 시스템의 리소스를 독점하거나 모두 사용, 또는 파괴함으로써 다른 사용자들이 이 시스템의 서비스를 올바르게 사용할 수 없도록 만드는 것을 말한다. 이런 의미에서 시스템의 정상적인 수행에 문제를 일으키는 모든 행위를 DoS라 할 수 있다. 그런데 이런 공격이 일어나는 방법은 매우 다양하다. 이 공격은 고의적으로 발생할 수 있지만 사용자의 의도와는 무관하게 발생할 수도 있으며, 공격자는 서비스 요구를 통해 서비스 중단을 초래할 수 있다. 예를 들어 공격자가 1초당 10개의 메일을 받을 수 있는 서버에 초당 20개의 메일을 보내면 공격 대상이 된 메일 서버는 메일을 제대로 전송하지 못할 것이며, 20개보다 훨씬 많은 메일을 보낼 경우엔 서버가 다운될 수도 있다[5].

2.2 DDoS 공격 유형

분산 서비스 거부 공격(DDoS)은 그 이름에서 알 수 있듯이, 서비스 거부공격(DoS)의 분산 형태이다. DoS 공격은 많은 양의 IP 킷을 전송함으로써 원격 호스트나 네트워크의 자원을 소비한다. 단일 호스트는 최대 전송률의 패킷 송신에 의해서도 상당한 피해를 입을 수 있지만, 공격자는 다중 호스트의 자원에 영향을 줌으로써 더 강력한 공격을 할 수 있다. 전형적인 DDoS 공격에서 공격자는 먼저 가능한 많은 호스트로 침입하고 두 종류의 좀비 프로그램[제어 프로그램(마스터 좀비)와 플러딩(flooding) 프로그램(슬레이브 좀비)]을 설치한다. 공격자가 마스터 좀비를 트리거 했을 때, 이는 악의적인(malicious) 트래픽을 목표 서버로 전송하도록 슬레이브 좀비에 명령한다. 따라서 서버의 자원을 소비함으로써 합법적인 사용자가 서버를 사용하지 못하도록 한다.



(그림 1) DDoS 공격의 형태

이 공격은 논리(logic) 공격과 플러딩(flooding) 공격의 두 가지 형태가 있다.

논리 공격은 서버 소프트웨어의 결점을 이용하여 서버를 다운시키거나 CPU 사용, 파일 저장소나 메모리 같은 서버 시스템의 자원을 고갈시킨다. 이런 형태의 공격의 일반적인 예는 SYN flood, IP 분할 overlap, 버퍼 오버플로우를 들 수 있다. 이 공격에 대한 방지 메커니즘은 많은 소프트웨어 회사에 의해 발달되어서, 시스템 관리자는 서버 소프트웨어를 업그레이드하거나 특정한 연속 패킷을 필터링함으로써 사이트를 방어할 수 있게 되었다.

반면에 플러딩 공격에서 공격자는 어떤 소프트웨어가 사용되는지 상관하지 않는다. 대신에 많은 양의 트래픽으로 공격함으로써 목표 네트워크의 가능한 모든 대역폭을 소비하려고 시도한다. 이에 대한 예는 TCP flood, UDP flood, Smurf, Fraggle, ICMP flood등이 있다. TFN, TFN2K(Tribe Flood Network 2000), Stacheldraht, intensify 같은 대부분의 DDoS 공격도구는 이런 플러딩공격을 여러곳에서 시도하게 한다. 현재까지 이 문제를 효율적으로 다룰 수 있는 기술이 없기 때문에 본 논문에서는 이런 타입의 공격을 탐지하고 대응 하는 방법에 대해서 초점을 맞췄다.

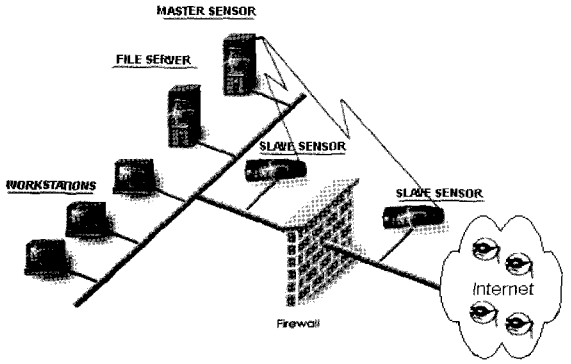
3. DDoS 탐지

3.1 시그니처(signature) 기반의 탐지

3.1.1 구조

정적 침입탐지기로서는 대표적인 공개소스 IDS인 Snort[6]를 사용하였다. snort는 경량화 네트워크 침입 탐지 시스템으로써 실시간 트래픽 및 패킷의 내용을 분석하고, 버퍼 오버플로우, 포트스캔 등 다양한 공격과 징후를 탐색과 매칭 기법으로 탐지한다. 오픈 소스의 장점인 지속적인 업그레이드가 지원

되어 지금은 Snort 2.1.2 버전까지 나왔으며, 룰셋이 80 여가지로 대폭 증가 되었다. 또한 수정 가능한 룰셋으로 이루어져있어 상황에 맞게 룰을 수정할 수 있으며, 많은 룰셋 중에서 필요한 룰만을 적용 시킬 수 있다. Snort는 기본적으로 tcpdump와 같은 패킷 sniffer, 네트워킹 트래픽을 디버깅하는데 사용되는 패킷 logger(기록계), 공격에 대한 침입탐지 이렇게 크게 세 가지 부류의 기능을 수행한다. snort는 침입에 대한 탐지가 이루어 졌을 때, 그 탐지된 공격의 종류와 페이로드 정보, 패킷의 정보 등 공격과 관련된 정보들을 저장하기 위해서 데이터베이스가 필요하다.



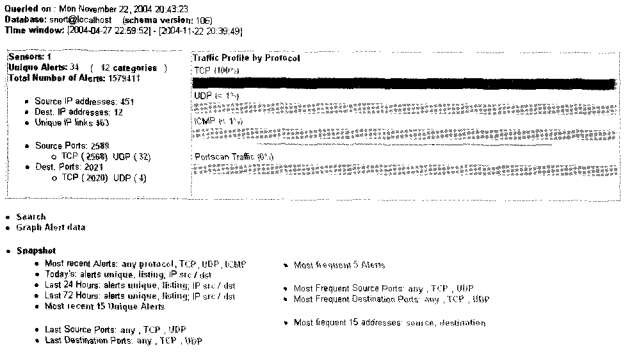
(그림 2) Snort IDS sensor 배치 구조

기본적인 Snort의 배치는 라우터나 방화벽 바로 옆에서 붙어서 전송되는 패킷들을 캡처한다. Libnet은 낮은 레벨에서 패킷을 캡처하여 버퍼에 저장하고 libpcap은 그 버퍼에 있는 패킷을 가져와서 스노트의 룰셋과 매칭 시킨 다음 alert라고 생각되면 시그니처를 생성하고 데이터 베이스에 저장한다.

snort에서는 MS-SQL, My-SQL, ORACLE 등 모든 데이터베이스를 지원하며, 데이터 베이스를 사용하지 않을 경우에는 일반 txt 파일로 로컬 디렉토리에 저장 할 수 있다.

3.1.2 측정 결과

다음 그림은 대전 노드에 설치한 snort에서 탐지한 결과이다. alert의 종류와 카테고리, 그리고 탐지된 모든 alert에 대해서 표현이 되어 있으며, 탐지된 패킷의 종류와 날짜까지 표현하고 있다.



(그림 3) Snort 메인화면

현재 (그림 3)에서는 대부분의 alert가 tcp 패킷에서 탐지가 된 것을 보여주고 있다.

3.1.3 분석

탐지된 alert는 다음과 같다. 시그니처는 탐지된 alert의 이름을 나타내고 옆에 연결된 링크는 탐지된 alert의 정보를 웹 문서형태로 보여준다. 분류(classification)는 탐지된 alert를 그 특성에 따라 카테고리로 분리를 시킨 것을 의미한다. 예를 들어 포트스캔과 관련 된 alert는 network-scan이라는 카테고리로 표현된다. total 카운트는 탐지된 alert의 수를 나타낸 것이다. 마지막으로 First, Last는 alert가 처음 탐지된 날짜와 시간, 그리고 마지막으로 탐지된 날짜와 시간을 나타내고 있다.

Signature	Classification	Total #	Sensor #	Src. Addr.	Dest. Addr.	First	Last
[snort] ICMP Destination Unreachable (Communication Administratively Prohibited)	misc-activity	1 (0%)	1	1	1	2004-10-15 06:32:25	2004-10-15 11:55:58
[cve][cat][bugtraq][snort] BAD-TRAFFIC IP Proto 103 (PIM)	non-standard-protocol	51893 (4%)	1	1	1	2004-11-15 13:25:26	2004-11-20 00:34:01
[arachNIDS][snort] WEB-MISC WebDAV search access	web-application-activity	282 (0%)	1	81	1	2004-10-03 02:55:59	2004-11-22 19:33:19
url[bugtraq][bugtraq][snort] MS-SQL Worm propagation attempt	misc-attack	59286 (4%)	1	15	10	2004-10-09 21:41:26	2004-11-12 15:05:30
[cve][cat][bugtraq][arachNIDS][snort] WEB-IIS ISAPI .ida attempt	web-application-attack	920 (0%)	1	36	1	2004-05-15 02:30:17	2004-10-24 21:55:41
[snort] WEB-IIS cmd.exe access	web-application-attack	138 (0%)	1	3	1	2004-10-06 16:44:18	2004-10-24 21:56:41
[arachNIDS][snort] ICMP PING NMAP	attempted-recon	70 (0%)	1	62	1	2004-04-30 01:11:07	2004-11-19 22:59:38
[snort] SCAN Proxy Port 8080 attempt	attempted-recon	524 (0%)	1	85	1	2004-05-02 21:10:14	2004-11-20 00:42:03
[cve][cat][bugtraq][snort] NETBIOS DCERPC Messenger Service buffer overflow attempt	attempted-admin	129 (0%)	1	2	1	2004-05-04 23:17:12	2004-06-25 21:56:28
url[snort] SCAN SOCKS Proxy attempt	attempted-recon	972 (0%)	1	127	2	2004-04-28 20:49:53	2004-11-20 00:42:06
[cve][cat][snort] WEB-IIS msadcs.dll access	web-application-activity	21 (0%)	1	10	1	2004-05-06 23:25:37	2004-11-19 23:31:03
[snort] SCAN Squid Proxy attempt	attempted-recon	434 (0%)	1	69	1	2004-05-04 11:27:23	2004-11-20 00:42:04
[bugtraq][arachNIDS][snort] WEB-IIS view source via translate header	web-application-activity	48 (0%)	1	16	1	2004-05-02 20:30:43	2004-10-25 19:18:35
[arachNIDS][snort] ICMP PING CyberKit 2.2 Windows	misc-activity	764 (0%)	1	42	1	2004-04-27 22:59:52	2004-11-10 00:17:33
[cve][cat][cve][cat][cve][cat][snort] SNMP public access udp	attempted-recon	264 (0%)	1	2	1	2004-05-15 10:50:37	2004-10-26 11:13:12
[snort] (snort_decoder): Truncated Tcp Options	unclassified	26 (0%)	1	5	1	2004-05-02 13:19:56	2004-08-04 03:54:55
[snort] ICMP superscan echo	attempted-recon	46 (0%)	1	21	1	2004-04-28 19:58:16	2004-11-19 23:39:37
[snort] SCAN SSH Version map attempt	network-scan	4 (0%)	1	2	1	2004-06-12 11:15:56	2004-06-30 01:23:08
[arachNIDS][cve][cat][snort] WEB-IIS ISAPI .printer access	web-application-activity	16 (0%)	1	5	1	2004-10-04 13:50:48	2004-10-24 21:55:24
[arachNIDS][snort] RPC portmap mountd request UDP	rpc-portmap-decode	2 (0%)	1	1	1	2004-04-28 19:06:00	2004-04-28 19:06:00

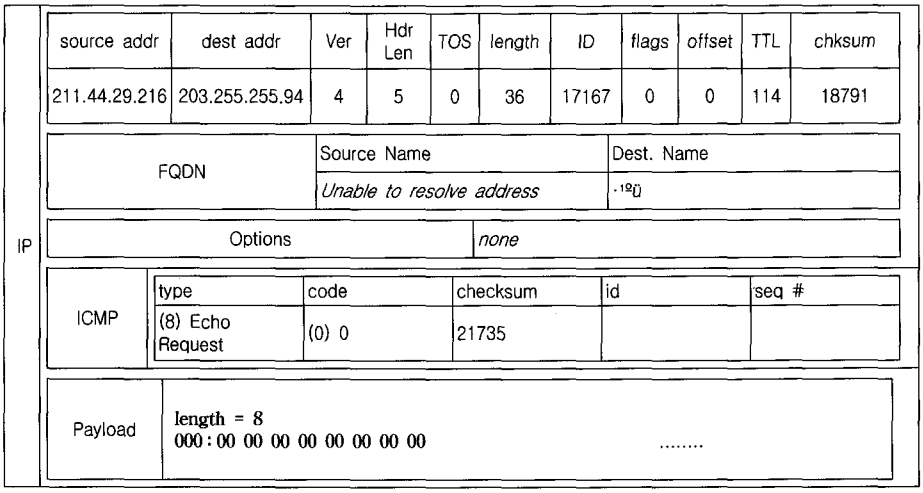
(그림 4) alert 리스트

가. ICMP PING NMAP

Nmap(Network Mapper)은 네트워크 보안을 위한 유틸리티로, 대규모 네트워크를 고속으로 스캔하는 도구이다. Nmap은 raw IP 패킷을 사용하여 네트워크에 어느 호스트가 살아있고, 그들이 어떠한 서비스(포트)를 제공하며, 운영체제(OS 버전)가 무엇이며,필터r/방화벽의 패킷 타입이 무엇인지 등 네트워크의 수많은 특징들을 점검할 수 있다. Nmap에서 사용되는 대부분의 기술은 호스트의 어떤 포트가 사용(listening) 되고 있는지를 스캔하기 위해 사용된다. 이 포트들은 통신 가능한 채널로 이들 포트들에 대한 매핑은 호스트에 대한 정보 교환을 용이하게 한다. 따라서 nmap은 시스템 관리자는 물론이고 해커를 포함한 네트워크 환경을 점검하기를 원하는 모든 사람에게 매우 유용한 도구로 사용된다. 여기는 네트워크 관리차원에서 보내지는 것이라 판단된다.

나. ICMP superscan 에코

supercan은 포트스캔을 위해서 일반적으로 사용되는 도구로써 윈도우 기반의 C++ 6.0 으로 만들어져있다. supercan의 기



(그림 5) 탐지된 ICMP superscan 에코 패킷

능은 다른 스캔도구와 마찬가지로 네트워크의 관리와 호스트와 라우터간, 또는 호스트와 호스트간의 연결이 되는지 확인하기 위한 도구로 사용되며, 아래 그림처럼 패킷의 Payload에 연속되는 16개의 0 즉 "0000000000000000"의 문자를 포함하고 있다.

다른 포트스캔과 마찬가지로 네트워크에서 액티브 호스트의 탐지를 확인하는 것으로 실제 사용되고 있지 않는 호트스를 검색해서 이것을 기반으로 해킹이나 공격을 시도 할 우려가 있다.

여기서는 한번 신호가 잡힌 것으로 단순히 네트워크 관리차원에서 이루어진 것으로 판단된다.

다. MS-SQL worm propagation attempt

Slammer worm이 MS SQL server의 취약점을 공략하는 것으로서 MS SQL 서버에게 많은 요청을 하여 버퍼에서 오버플로우가 발생하도록 유도하는 것이다. Slammer worm은 원격 공격자가 MICROSOFT SQL Server 2000의 취약점을 공략하여 타켓이 되는 호스트의 특정포트 1434에 접근하여 heap-base 기반의 오버플로를 발생시켜 서비스를 못하게 한다. 이를 막기 위해서 관리자는 포트 1434를 막아 놓거나 그 포트에 접근하려는 패킷을 필터링 함으로서 침입을 줄일 수 있다. (그림 4)의 alert 리스트를 보면, 15개의 호스트(Src. Address)가 KOREN에 있는 10개의 노드들에 대한 공격 시도 있었다는 것을 Sort가 탐지하여 보여주고 있다. (그림 4)의 alert 리스트에는 자세한 정보를 표시하지 않았지만 150.162.xx.xx 호스트는 공격 호스트들 중에 하나이다. 공격대상이 되는 호스트의 주소는 보안상 표기하지 않았다.

라. BAD-TRAFFIC IP Proto 103(PIM)

Snort에 의해 탐지된 신호를 보면 150.162.28.240 호스트가 Snort 센서(Snort가 설치된 호스트)에 접근을 시도했다는 것을 알 수 있다. Cisco IOS 소프트웨어를 구동하고 IPv4 패킷을 처리하도록 구성된 시스코 라우터와 스위치들은 DoS(Denial of Service) 공격에 취약하다. 흔치는 않지만 특별히 조작된 일련의 IPv4 패킷을 해당 장치에 직접 보낼 경우, 입력 큐가 가득 차면서 수신 인터페이스의 트래픽 처리가 중단될 수 있다. 공격자는 Cisco IOS 버전 11.x 에서 12.2 사이의 라우터의 취약점을 공략하는 것으로서, 많은 수의 IP 패킷을 사용하여 라우터에 보냄으로서 denial of Service를 유발하게 한다. 현재 KOREN 라우터의 ISO는 12.0 버젼을 사용하고 있는데 이러한 취약점을 방지하기 위해서는 IOS를 12.2 이상의 버전으로 Upgrade를 하거나 시스코 보안 권고문을 참고하여 패치를 적용할 필요가 있다. snort에 의한 잡힌 신호는 라우터가 정기적으로 인접 PIM 라우터들을 발견하기 위해 Hello 메시지를 보내는 것이 탐지되었다. 이 Hello 메시지는 224.0.0.13 주소(모든 PIM 라우터 그룹)를 사용하여 멀티캐스트된다. 라우터는 Hello 메시지가 수신되어도 알림 메시지를 보내지 않는다. 또한 다른 몇몇 프로토콜(예 : DVMRP)과는 달리, Hello 메시지가 수신되었을 때 자동적으로 멀티캐스트 트래픽을 전달할 송신 인터페이스 목록에 인터페이스가 자동으로 추가되지 않는다.

마. SCAN Proxy(8080) attempt

SCAN Proxy 8080 포트 역시 supercan의 기능은 다른 스캔도구와 마찬가지로 네트워크의 관리와 호스트와 라우터간, 또는 호스트와 호스트간의 연결이 되는지 확인하기 위한 스캔이 탐지된 것으로 호스트의 포트8080이 탐색됐다는 것을 알려준다.

바. SCAN Socks Proxy attempt

이는 외부의 어떤 호스트가 특정 호스트의 포트 1080으로 통신하기 위해 시도하기 위한 것이다. 즉 게이트에서 잘못 구성된 Socks이나 Proxy를 이용하여 연결을 시도 하는 것이다. 하지만 어떤 사용자가 단지 연결을 설립하기 위한 시도가 포함되어 있는 것으로 시스템에 대한 공격으로 생각하지 않아도 될 것으로 보인다.

사. SCAN Squid Proxy attempt

호스트에서 스캔이 이루어 진 것을 탐색하는 것으로서 공격에 대한 전조로 간주된다. 공격자는 포트 21과 20이 FTP 서버로 사용되었다는 것을 파악한 다음에 FTP 서비스에 대한 취약점을 파악하고 이를 공략할 수 있다.

아. SNMP public access udp

라우터에는 접근을 허용하는 커뮤니티 네임이 있다. 일반적으로 default로 public@로 설정되지만 다른 접근자를 차단하기 위해서 커뮤니티 이름을 바꾸어둔다. 이는 라우터에 접근하여 환경을 변경함으로써 악용될 우려가 있기 때문이다. 여기서는 라우터에 어떤 접근 시도를 탐지한 것이다.

자. PROBE-FIN_SCAN

FIN 스캔을 하는 패킷을 가리키는 것으로 TCP 헤더에 FIN flag만 세팅 되어있다. 이것은 은밀한 포트 스캔을 할 때 나타나는 것으로써 네트워크의 취약성을 찾아내기 위해서 사용되기도 한다.

차. WEB-MISC robots.txt access

robots.txt. 파일에 접근하려는 시도가 탐지 된것이다. Robots.txt 는 일반적으로 웹 사이트의 indexing을 찾기 위해서 존재한다. 웹 마스터는 여기에 자신의 정보를 표기한다. 공격자는 이러한 정보를 가지고 공격하려는 웹사이트의 정보를 알아낼 수도 있다.

3.2 통계적 기반의 동적 탐지

3.2.1 통계적인 탐지 알고리즘

DDoS 공격은 일반적인 패킷으로 공격이 이루어지므로 합법적인 패킷과 구분하기 어렵고, 각각의 공격 소스에서 보내는 패킷의 양이 적기 때문에 local 관리자가 쉽게 탐지할 수 없다. 따라서 이를 탐지하기 위해서는 통계적인 방법을 사용하는 것이 가장 효율적이다.

통계적인 탐지 알고리즘에는 트래픽 볼륨, 패킷 속성값의 엔트로피(엔트로피), 카이 제곱(Chi-Square) 검증법등이 사용되고 있다[7].

이 가운데 트래픽 볼륨 측정은 패킷이 이더넷 카드에 도착하는 시간을 계산하는 것이다.

$$\sum_{i=1}^{n} PAT[i] - PAT[i-1] \qquad \text{(식 1)}$$

위 식은 각각의 패킷이 도착하는 시간을 측정하여 그룹단위로 패킷이 도착한 시간을 합한다. 즉 100개의 패킷을 한 그룹으로 설정을 했다면 100개의 패킷이 이더넷 카드에 도착하는 시간을 측정하는 것이다. 트래픽 볼륨 값이 낮을 수록, 현재의 트래픽이 갑자기 증가했다는 것으로 이상 트래픽 발생 가능성을 암시한다.

다음 엔트로피 연산법은 어떠한 네트워크 속성값에 대한 임의성(randomness)를 계산한 뒤, 그 값의 평균의 변화량을 탐지하는 방법이다.

$$H = -\sum_{i=1}^{n} \pi \log \pi \qquad \text{(식 2)}$$

위의 식은 n개의 속성값(예, 소스주소, 목적지주소)에 대한 엔트로피 H를 구하는 공식이다.

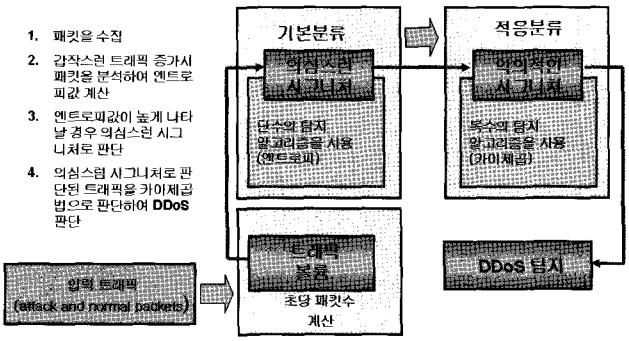
카이 제곱 검증법은 속성값에 대한 분산도를 측정하는 방법이다. 여기서는 기대값에 대한 분산도를 계산하여 그 값에 따라 비정상적인 속성값을 탐지할 수 있다. 이 방법의 구체적인 식은 다음과 같다.

$$x^2 = \sum_{i=1}^{B} \frac{(N_i - n_i)^2}{n_i} \qquad \text{(식 3)}$$

여기서 B는 샘플 패킷들이 가질 수 있는 값들을 묶어놓은 binning 값이다(ex. 패킷길이는 0-64, 65-128, 129-255로 binning 될 수 있다). N_i 는 N개의 샘플 패킷에서 각각의 binning 범위에 속하는 패킷의 개수고, n_i는 일반적인 분포에서 binning에 속하는 기댓값이다.

본 논문에서 제안한 메커니즘에서는 이상 트래픽을 실시간으로 탐지하기 위한 방법으로 성능을 저하시키지 않으면서 탐지가 가능하도록 하기 위해서는 정상적인 트래픽일 때는, 간단한 연산을 수행하는 트래픽을 계산한다. 그리고 여기서 이상이 생기면 보다 정밀하지만 연산이 많은 엔트로피 탐지 알고리즘을 수행하고, 엔트로피 값에도 이상이 탐지되면, 최종적으로 카이제곱 연산을 실행함으로써, 성능 저하를 시키지 않으면서도 이상 트래픽을 탐지 할 수 있도록한다.

3.2.2 탐지 기법



(그림 6) 시그니처 생성과정

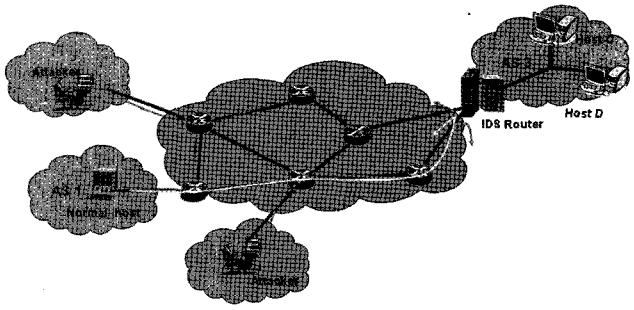
위의 그림은 탐지 알고리즘에서 시그니처가 생성되는 과정을 설명한 그림이다. 이는 기본분류(elementary classification)와 적응분류(adaptive classification)의 두 과정으로 나뉜다. 먼저 기본분류 과정에서는 탐지부를 지나는 패킷들을 모니터링하면서 연산 난이도가 비교적 낮은 엔트로피 계산법을 사용하여 의심스런(suspicious) 시그니처를 생성하게 된다. 이런 의심스런 시그니처 생성과정에서는 혼잡 문제를 줄이기 위해, 정밀도가 낮도록(FPR, NPSR이 높도록) 임계값이 낮게 설정된다. 다음에 공격패킷으로 오분류된 일반패킷의 양을 줄이기 위하여 적응과정이 수행된다. 이 과정에서는 어떤 기간 동안 계속 의심스런 시그니처에 속하는 패킷속성값을 복수의 탐지 알고리즘을 사용하여 더 정밀히 분석하여 악의적인 시그니처를 생성하게 된다. 여기서 악의적인 시그니처는 간접 피해문제를 해

결하기 위해, 정밀도가 높도록(FNR, NPSR이 낮도록) 임계값이 높게 설정된다.

4. 검증된 IP 테이블

4.1 기존의 침입 대응 시스템

기존에 나와있는 DDoS 탐지 및 대응 시스템은 많다. 대표적인 것으로 통계적으로 트래픽을 측정하고 소스 IP에 대해서 가중치를 포함한 확률값을 적용하여 탐지한다[8]. 그리고 점수가 높은 것을 기준으로 임계값(threshold)까지 허용하고 나머지는 차단하는 것이다. 또 다른 방법으로는 리키버킷를 사용하여 호스트나 네트워크의 자원이 허용하는 범위 내에서만 트래픽을 수용하는 방법이 있다[9]. 이러한 방법은 DDoS를 차단할 수는 있지만, DDoS가 발생 할 때 공격 트래픽이 아닌 정상적인 클라이언트 역시 호스트나 네트워크에 접속 할 수 없다는 단점이 있다. 다른 면으로 생각 할 때 그 정상적인 클라이언트는 DDoS 공격으로 인한 서비스 거부와 다를 바가 없다.



(그림 7) DDoS 대응시 정상호스트도 차단

위 그림을 보면 공격자가 DDoS 공격을 할 때 이를 차단하는 모습이다. 하지만 정상 적인 클라이언트 또한 DDoS 공격을 당하고 있는 네트워크나 호스트에 접속 할 수 없다는 것을 알 수 있다. 이러한 DDoS 대응의 문제점을 보완하기 위해서 검증된 IP 테이블 방법을 사용한다.

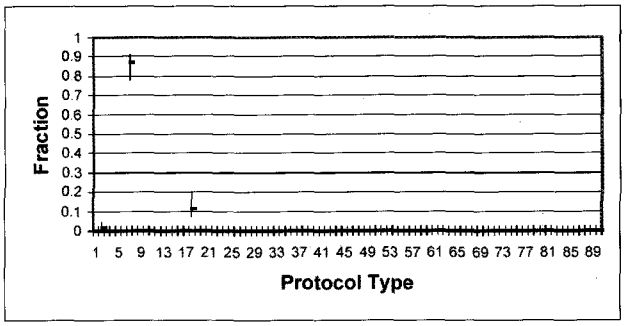
4.2 통계적 History-based IP

Auckland Trace		Small ISP Trace	
Date	Percentage	Date	Percentage
01-Mar-26	88.7%	00-Apr-20	81.4%
01-Mar-27	90.3%	00-Apr-21	76.9%
01-Mar-28	89.1%	00-Apr-22	77.5%
01-Mar-29	89.2%	00-Apr-23	79.6%
01-Mar-30	90.2%	00-Apr-24	80.7%
01-Mar-31	89.9%	00-Apr-25	79.4%
01-Apr-1	88.1%	00-Apr-26	80.1%

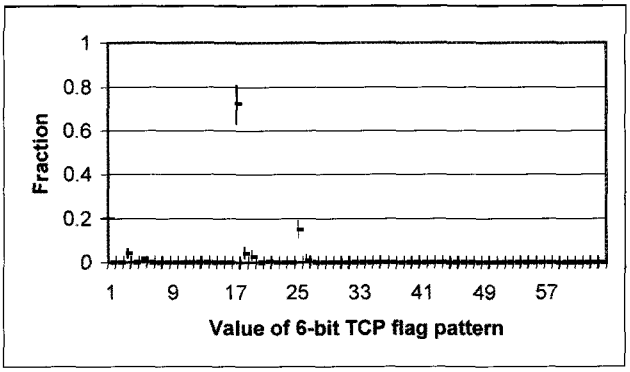
(그림 8) 통계적으로 반복되어 나타나는 IP 백분율

(그림 8)은 뉴질랜드의 대표적인 트래픽측정 전문 연구 그룹인 WAND에서 인터넷 트래픽을 관찰하고 인터넷 서버에 접속하는 소스 IP를 매일 측정하여 그 반복성을 나타낸 그림이다[10]. 내용을 살펴보면 26일에 옆에 나타난 88.7%라는 값은, 그날 측정된 IP중에서 전날 나타났던 IP, 즉 반복해서 다음날 나타난 백분율을 나타내고 있다. 쉽게 이해하면, 우리가 어떤 특정 웹 사이트나 네트워크에 접속 할 때, 이전에 방문 했던 곳을 다시 간다는 것을 의미한다. 역으로 웹 서버에서 생각해 보면 오늘 웹에 방문한 사람이 내일도 방문할 확률이 88.7% 된다는 것을 의미한다. 이러한 자료가 중요한 것은, 검증된 IP 테이블에서 하루 동안 들어오는 트래픽을 측정하고 이를 소스 IP 주소의 전체 트래픽에 대한 백분율 값을 구한 다음, 소스 IP 주소들을 여러 가지 합법성 테스트를 거친 후 검증된 IP로서 사용하기 때문이다.

4.3 정상적인 패킷의 유용한 확률 값



(그림 9) 패킷 프로토콜별 빈도

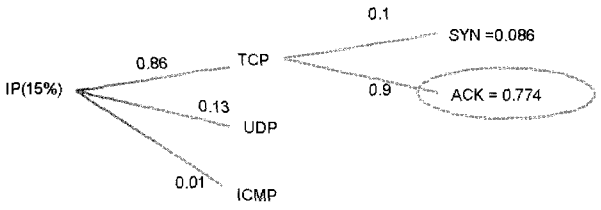


(그림 10) TCP flag 값 빈도

(그림 9)와 (그림 10)은 WIDE 프로젝트에 하루 동안 인터넷의 패킷들을 분석한 그래프이다[11]. (그림 9)를 보면 인터넷 패킷 중에서 90% 가까운 값을 보여주고 있는 것은 프로토콜 type 6번이다. 이것은 TCP 패킷을 의미한다. 또한 프로토콜 type 1번은 ICMP를 의미하며, 17번은 UDP를 의미한다. 이 그래프를 보면 전체 트래픽에서 TCP가 대부분을 차지한다는 것을 알 수 있다. (그림 13)은 TCP 패킷 중에서 flag 값을 분석해서 나타낸 것이다. 이것을 보면 80%에 가까운 값을 보여주는 것은 flag 값이 010000, 즉 16 값인 ACK 값이다. 여기서 SYN 메시지를 보면 2%정도 밖에 안 된 다는 것을 알 수 있다.

이러한 값들이 중요한 이유는 앞에서 DDoS에 대한 특징에서 살펴봤던 것처럼 DDoS 공격의 특징 중 SYN flooding을 이용한 공격 때문이다. 이러한 SYN flooding 공격은 비정상적인 연결 요청을 계속 함으로써 서비스를 제공하는 서버의 자원을

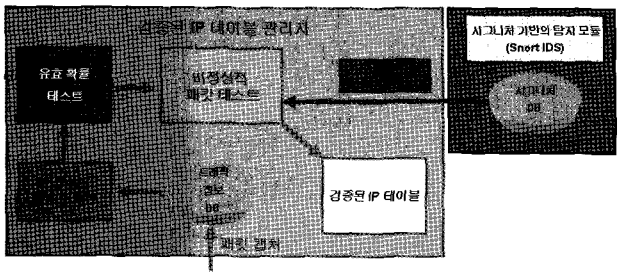
소모한다[12]. 우리가 일상적으로 인터넷이나 FTP, TELNET 등의 네트워크 프로그램을 사용할 때 제일 처음 SYN 메시지를 보내서 연결 요청을 한다. 그리고 연결을 제 설정 할 때는 제외하고 대부분 데이터를 주고받음을 확인하는 ACK 메시지를 사용한다.



(그림 11) TCP ACK의 유용한 확률

WIDE 프로젝트나 WAND 그룹에서 인터넷 트래픽을 측정하고 이를 바탕으로 가장 정상적인 패킷의 속성값들이 무엇인지를 알 수 있었다. (그림 11)은 이러한 자료들을 바탕으로 가장 정상적인 것이 TCP ACK라는 것을 알 수 있었고, 특정 소스 IP 주소에서 TCP ACK의 백분률값을 구하는 것을 나타낸 것이다.

4.4 검증된 IP 테이블 메커니즘

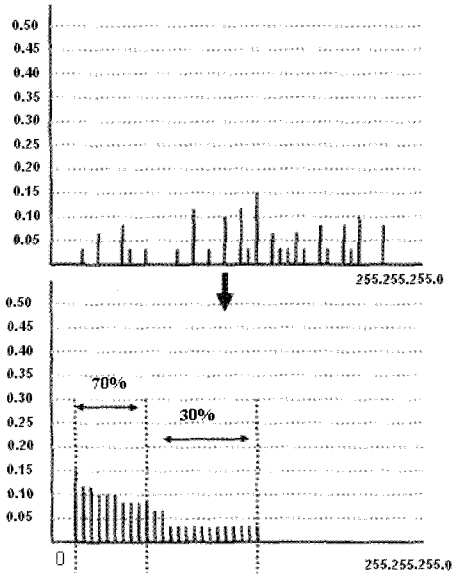


(그림 12) 검증된 IP 테이블 구조

(그림 12)는 검증된 IP 테이블 구조를 나타내고 있다. 검증된 IP 테이블 구조에서 제일 처음 하는 것은 네트워크에서 패킷들을 캡처하여 DB에다가 저장하는 것이다. 검증된 IP 테이블은 실시간 트래픽 측정이 아니라 통계를 위한 모듈이기 때문에 트래픽정보(Traffic Information) DB를 사용한다. 이렇게 DB다가 저장을 한 후, 특정 시간이 되면 이러한 자료들을 통계를 낸다.

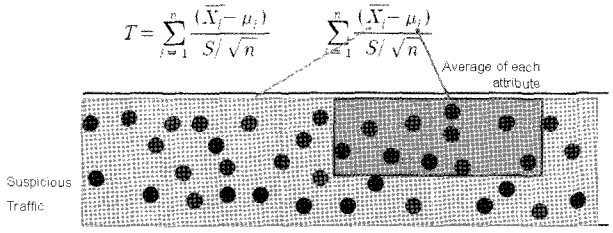
4.5 소스 IP 정렬

(그림 13)은 source IP 주소의 네트워크 prefix 별로 백분율을 구한 것 이다. 여기서 전체 100%가 되는 값은 하루 동안 측정된 트래픽의 패킷 수를 의미한다. 우선 DB에서 소스 IP 주소의 전체 트래픽에 대한 지분을 구한다. 그런 다음 가장 높은 값을 가진 소스 IP 주소 별로 정렬을 한다. 이러한 이유는 가장 많이 나타나는 소스 IP 주소를 검증된 IP 테이블에 우선적으로 넣는 다는 것을 의미한다. 네트워크 서비스 차원에서, 가장 많이 방문하는 클라이언트에게는 서버에 긴급 사태가 일어났어도 우선적으로 서비스를 제공하겠다는 것을 의미한다. 여기서 70%라는 값은 앞에 통계적으로 반복해서 나타나는 IP의 퍼센트를 의미한다. 즉 통계적으로 70%가 매일 반복해서 나타난다면 그 수준까지만 검증된 IP 테이블에 넣는다는 것이다.



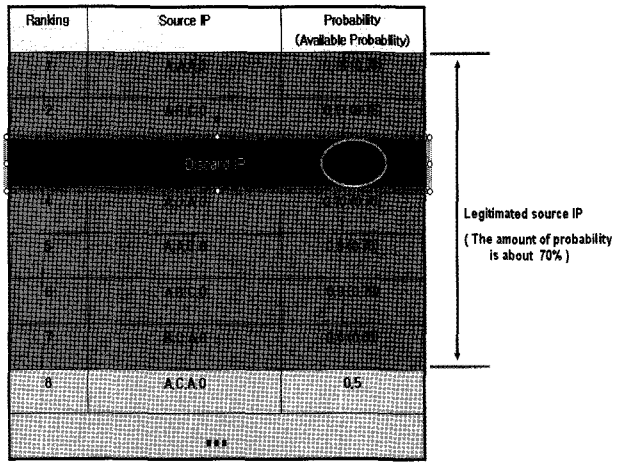
(그림 13) 트래픽 정보 DB에서 소스 IP 데이터 통계

4.6 T-테스트 검증



(그림 14) T-테스트 검증

(그림 14)는 T-테스트 검증의 개념을 보여주고 있다. T-테스트라는 것은 전체에서 샘플 표본을 추출 했을 때, 이 추출된 표본이 전체와 비교하여 비슷하게 추출된 것인가 하는 유의성 검증을 하는 것이다[13]. 다시 말하면 전체 트래픽에서 샘플 그룹을 추출 했을 경우, 이 그룹이 전체와 비슷한가 아니면 잘못 표본을 추출 한 것인가를 판단한다. 여기서 판단이 되는 기준은 앞에서 말했던 TCP ACK의 유용한 확률 값이다. 바로 전 단락에서 트래픽이 저장된 DB로부터 소스 IP 주소 별로 백분율이 높은 순으로 정열을 하여 검증된 IP 테이블에 넣는다고 했다. 하지만 단순이 자주 반복되는 소스 IP 주소라고 해서 무조건 검증 IP 테이블에 넣는다면 그건 보안적으로 위험한 생각이다. DoS처럼 특정한 소스 IP 주소에서 대량의 트래픽을 유발 시켰다면, 당연히 높은 백분율을 가질 것이고 이를 바탕으로 검증된 IP 테이블에 우선적으로 들어가게 될 것이다. 이를 막기 위해 전체 트래픽의 TCP ACK 확률 값과 비교해서 많이 차이가 난다면 정상적아님을 확인받고 테이블에서 버려지게 된다.

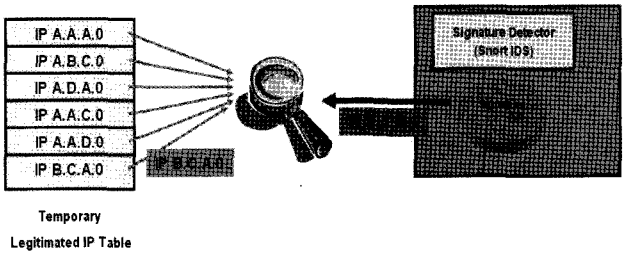


(그림 15) T-테스트 검증 후 검증된 IP 테이블에서 제거

(그림 15)를 보면 정렬 된 소스 IP의 TCP ACK 값을 T-테스트 검증을 통해서 이상으로 판정되면 검증된 IP 테이블에서 버리는 것을 알 수 있다. 이렇게 함으로써 단지 전체 트래픽에서 그 지분이 높다고 해도 합법성 검증을 통과하지 않는다면 검증된 IP 테이블에 등록되지 않는 다는 것을 의미한다.

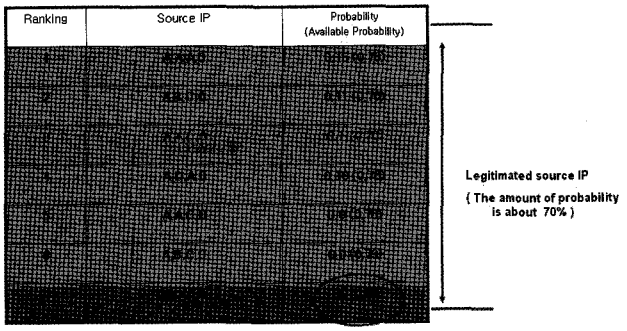
4.7 Snort 시그니처 검증

본 논문 전반부에서 시그니처 기반의 Snort IDS에 대해서 설명을 했었다. Snort는 DDoS의 사전 작업인 sniffer, port scan, bad traffic 등 그 패턴이 정의 되어있는 침입에 대해서 탐지를 하고 그 정보를 DB에 저장을 한다. 이러한 침입 정보가 필요한 이유는 검증된 IP 테이블은 확률적으로 비정상 패킷이라는 것 밖에 검증할 수 없다. 즉, 특정 소스 IP가 전체 트래픽과 비교 했을 때 확률적으로 비정상임을 판정한다. 하지만 T-테스트로는 정상적이어도 port scan처럼 침입을 위한 행위는 탐지 할 수가 없다. 그래서 snort의 침입 정보 DB의 자료가 필요하다.



(그림 16) Snort의 침입 정보 DB와 검증된 IP 테이블과 비교

Snort의 침입 정보 DB는 어떤 침입 종류가 있었는지, 그리고 어떤 소스 IP 주소로부터 그런 행위가 이루어 졌는지에 대한 자세한 정보가 있다. 여기서 침입 행위가 이루어 졌던 소스 IP 주소의 정보를 찾아서 검증된 IP 테이블에 있는 IP 주소와 비교를 한다. 만약 검증된 IP 테이블에 있는 소스 IP 주소 중에서 Snort에서 침입 행위가 탐지된 IP라면 그 소스 IP 주소는 테이블에서 삭제한다.

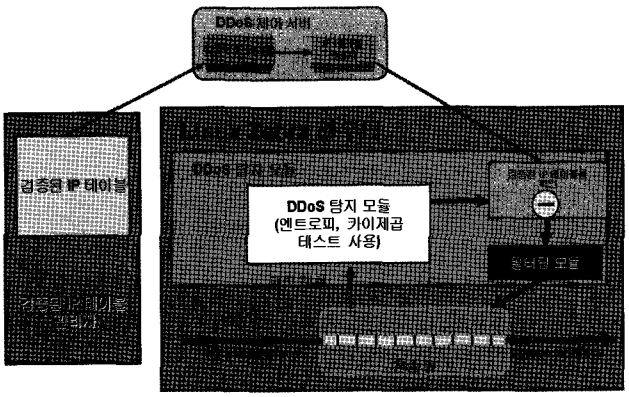


(그림 17) Snort와 검증 후 검증된 IP 테이블에서 제거

(그림 17)은 snort와 검증 후 불법 흔적이 있었던 소스 IP 주소를 검증된 IP 테이블에서 제거 하는 것을 나타내고 있다.

5. DDoS 침입 탐지 및 대응 시스템

5.1 DDoS 침입 탐지 및 대응 시스템 구조



(그림 18) DDoS 침입 탐지 및 대응 시스템 구조

(그림 18)은 본 논문에서 제안하고 있는 DDoS 침입 탐지 및 대응 시스템이다. 그림에서 보면 크게 두 가지 모듈이 있다. 하나는 검증된 IP 테이블이고 다른 하나는 통계 기반의 DDoS 탐지 모듈이다. 합법적인 IP 테이블 모듈은 일정 기간 동안 트래픽을 수집하여 여러 가지 합법성 검사를 한 후에 최종적으로 검증된 IP를 가지고 있는다. 그리고 Linux Zebra 라우터 위에 있는 DDoS 탐지 모듈은 실시간으로 현재 들어오고 있는 트래픽을 검사하고 엔트로피, 카이제곱 테스트 등을 측정하여 DDoS를 판정한다. DDoS 탐지 모듈에서 DDoS라고 판정을 내리면 측정된 트래픽 볼륨에 있는 모든 소스 IP 차단시켜, 현재 들어오고 있는 모든 트래픽을 차단시켜서 DDoS에 대응한다. 하지만 여기서 다른 DDoS 대응 시스템과 다른 점이 있다.

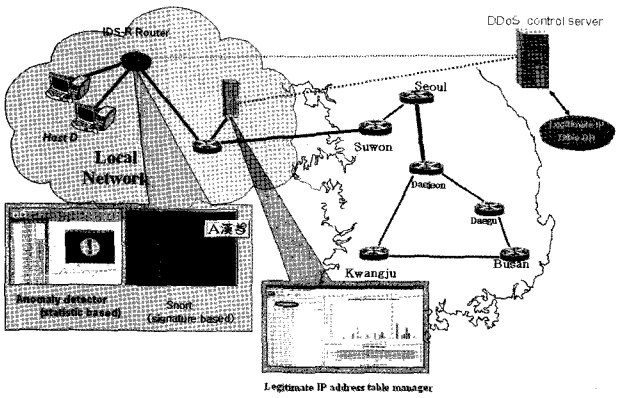
절대적으로 모든 트래픽을 차단시키는 것이 아니라, 검증된 IP 테이블에 있는 소스 IP 주소는 차단 목록에서 제외시킨다. 이렇게 함으로써 DDoS 공격에 대응 하면서, 기존에 합법적인 소스 IP 주소로 판별된 클라이언트들은 트래픽 차단과는 상관없이 지속적으로 서비스를 받을 수 있다.

또 다른 부가적인 모듈은 DDoS 제어 서버이다. 이 서버의

역할은 각각의 네트워크에 설치된 검증된 IP 테이블의 목록을 가지고 있다가, DDoS를 탐지한 곳에서 검증된 IP 목록을 요구하면 그 목록을 보내주는 역할을 한다. 이렇게 함으로써 검증된 IP 테이블과 DDoS 탐지 모듈이 분리되어 보다 유기적인 역할을 할 수 있다.

5.2 DDoS 침입 탐지 및 대응 시스템 설치

시스템 설치는 KOREN의 수원 노드에 설치에 되어 있다. 합법성 검사를 하는 검증된 IP 테이블 서버와 DDoS 탐지로 분리되어 KOREN 망을 통해서 들어오는 트래픽을 탐지하고 있다.

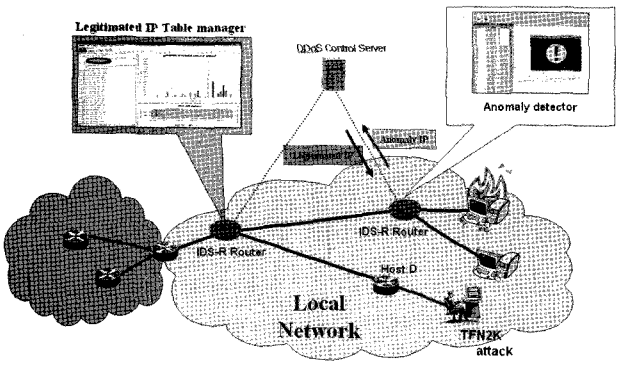


(그림 19) DDoS 탐지 및 대응 시스템 설치

(그림 19)는 DDoS 침입 탐지 및 대응 시스템을 KOREN에 설치된 형태를 보여주고 있다. IDS-R 라우터는 DDoS 탐지 모듈과 시그니처 기반의 Snort가 같이 설치되어 있다. 라우터는 Linux 기반의 Zebra 라우터로 되어 있다. DDoS가 발생시 DDoS 제어 서버에 있는 검증된 IP 목록을 받아서, 그 목록에 있는 IP를 제외한 나머지 모든 트래픽을 차단한다. 지금은 수원 노드에만 설치되어 있지만, 향후에는 KOREN의 모든 노드에 설치하여 통합적으로 보안망을 구성할 것이다.

6. DDoS 침입 탐지 및 대응 시스템 테스트

6.1 테스트 환경



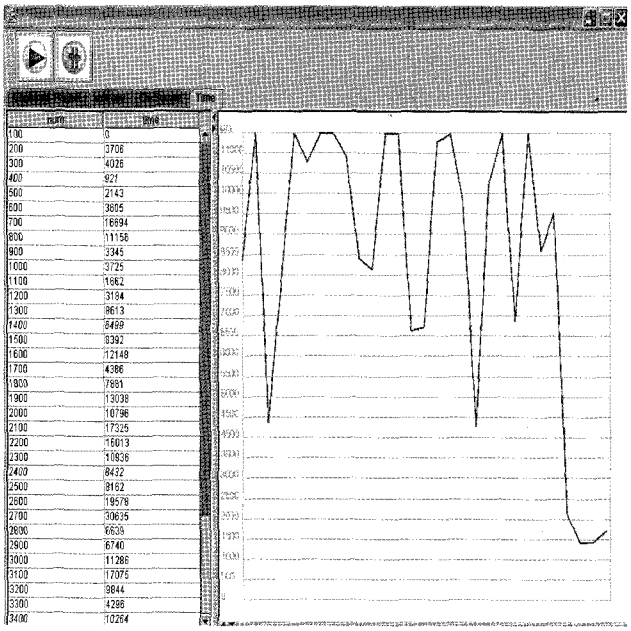
(그림 20) DDoS 탐지 및 대응 시스템 실험환경

위의 그림은 실험환경을 설명한 그림이다. DDoS 공격 도구로는 TFN2K를 사용하였고, 공격대상이 되는 호스트(victim) 앞에 IDS-R을 설치하여 DDoS를 탐지 하였다. 다음은 실험에 사용한 TFN2K에 관하여 설명한 것이다.

6.2 TFN2K

TFN은 trinoo와 거의 유사한 분산 도구로 많은 소스에서 하나 혹은 여러개의 목표 시스템에 대해 서비스거부 공격을 수행한다. TFN은 UDP flood 공격을 할 수 있을 뿐만 아니라 TCP SYN flood 공격, ICMP echo 요청 공격, ICMP 브로드캐스트 공격(smurf 공격)을 할 수도 있다. TFN 서비스 거부공격은 공격자가 클라이언트(혹은 마스터) 프로그램이 공격명령을 일련의 TFN 서버들(혹은 데몬들)에게 보냄으로써 이루어진다. 그러면 데몬은 특정 형태의 서비스거부 공격을 하나 혹은 여러개의 목표 IP 주소를 대상으로 수행한다. 소스 IP 주소와 소스 포트는 임의로 주어지고, 패킷의 사이즈도 바꿀 수 있다.

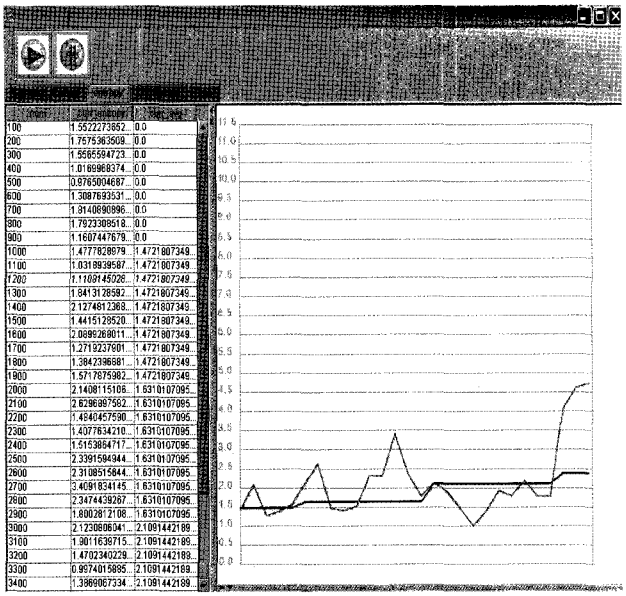
6.3 DDoS 탐지 실험결과



(그림 21) 트래픽 볼륨 그래프

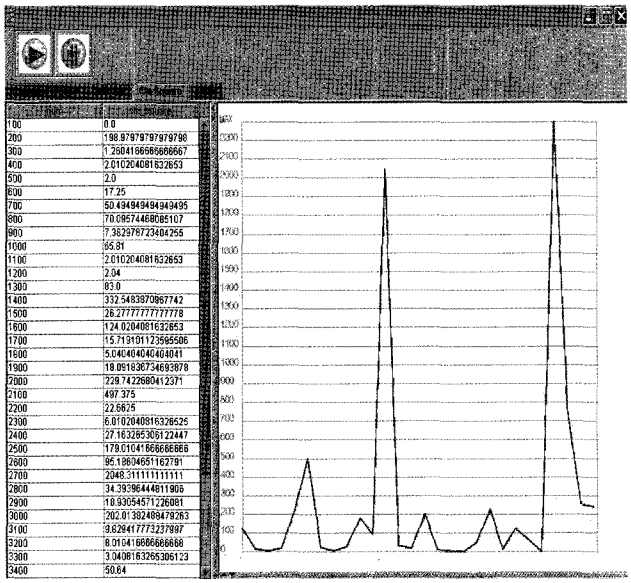
위의 과정은 NIC(Network Interface Card) 버퍼에 도달한 패킷이 도달한 시간을 측정한 것이다. 왼쪽 테이블은 패킷 100의 그룹이 NIC 버퍼에 도달한 시간을 나타냈으며 슬라이드 바로 특정부분 패킷그룹의 트래픽 볼륨 값을 알 수 있다. 오른쪽은 왼쪽 테이블 값을 그래프로 표현한 것이다. 여기서 X 축은 시간, Y축은 패킷의 개수를 나타낸다. 그림을 보면 동그라미가 있는 부분의 트래픽 볼륨 값이 급격히 떨어진 것을 볼 수 있다.

이것은 패킷의 그룹이 NIC 버퍼에 도달하는 시간간격이 짧아졌다는 것을 의미하고, 이는 DDoS와 같은 이상 트래픽이 나타날 수 있음을 암시하고 있다.



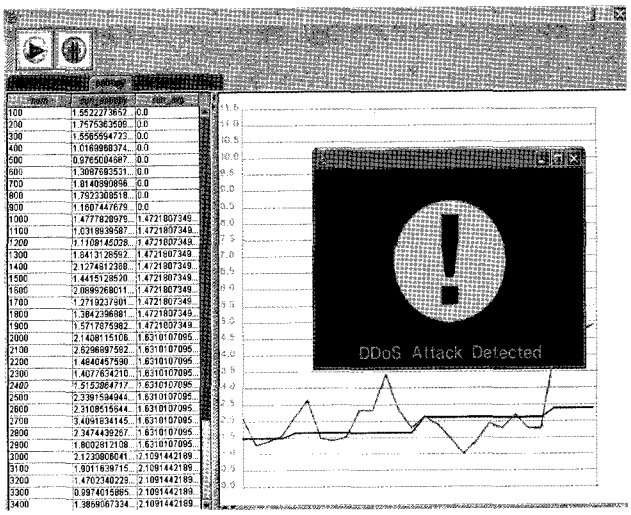
(그림 22) 엔트로피 테스트 그래프

위의 그림은 기본분류(Elementary Classification)에 해당하는 소스 주소의 엔트로피를 계산한 결과이다. Y축은 패킷의 수를 나타내고 X축은 엔트로피 값을 나타낸 것이다. 점선으로 된 라인은 엔트로피의 기댓값을 그려주고 있으며, 다른 선은 현재 측정되고 있는 엔트로피값을 보여주고 있다. 여기서 값을 계산하기 위한, 패킷 그룹의 크기가 커질수록 정확성이 높아지는 반면 연산이 오래 걸린다. 반대로 패킷 그룹의 크기가 작아질 수록 정확성은 떨어지지만 연산을 빨리 수행할 수 있다. 위의 결과를 살펴보면 TFN2K를 사용할 때 엔트로피 값이 급격히 증가했다는 것을 알 수 있다(동그라미 부분). 이 경우에서 임계값을 4.0정도로 하였을 때 이상 트래픽으로 탐지할 수 있을 것이다. 보다 정밀한 검사를 하기위해 카이제곱 테스트를 실시한다.



(그림 23) 카이제곱 값을 표시한 그래프

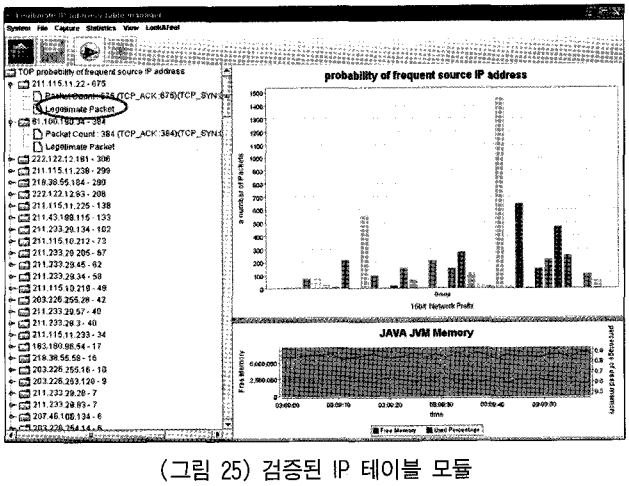
위의 결과는 의심스런 패킷으로 탐지된 패킷 속성값 중 패킷 소스 IP, 포트, 패킷길이를 카이제곱 테스트로 분석한 결과이다. 샘플링 되는 패킷의 개수를 100개로 줄여서 분석하였다. 결과를 살펴보면 값이 2000 이상인 경우가 2곳 발견되는데 이것은 카이제곱 테스트에 의하여 처음 그래프 부분은 DDoS처럼 보이지만 표준 편차 값에 의하여 이전 트래픽과 같다고 판정이 된 것이다. 하지만 다음 그래프(동그라미 부분)은 이전 트래픽과 다르다고 판별이 되었음을 나타난다.



(그림 24) DDoS 검출 그래프

위의 그림은 최종적으로 DDoS가 탐지된 그림이다. 이는 처음에 트래픽 볼륨에서 갑작스런 트래픽이 증가되어 엔트로피 탐지를 한 다음, 최종으로 카이제곱 판별을 거친 후 DDoS로 탐지 된 것이다.

6.3 검증된 IP 테이블 구현 결과

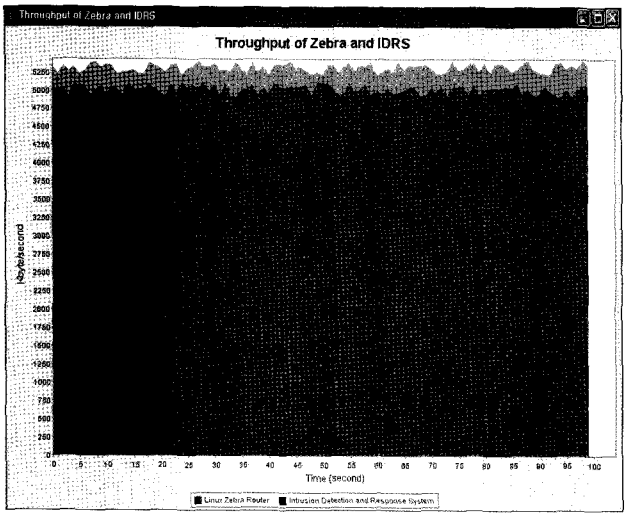


(그림 25) 검증된 IP 테이블 모듈

(그림 25)는 검증된 IP 테이블 모듈 구현을 나타낸 것이다. 그림 왼쪽을 보면 소스 IP 주소 별로 높은 빈도 값 순으로 정렬 된 것을 볼 수 있다. 윈도우에서 볼 수 있는 트리형식으로 표현함으로써 보다 많은 정보를 보려면 클릭하여 그 하위 정

보를 보면 된다. 여기서 동그라미 표시한 곳을 보면 검증된 소스 IP 주소라는 표시가 되어 있음을 알 수 있다.

그림의 오른쪽 막대그래프는 네트워크 prefix 별로 구분한 것이다. 막대그래프 위에 마우스를 올려 놓으면 그 네트워크에 대한 정보가 나타난다. 그림 아래를 보면 JAVA JVM의 메모리 사용을 실시간으로 보여주고 있다. 만약 저 메모리가 전부 사용되어 메모리부족 상태(out of memory)가 된다면, 프로그램은 더 이상 패킷들의 정보를 수집하거나 통계를 낼 수 없기 때문에 수행을 멈추고 중지 될 것이다. 그래서 프로그램이 JVM 메모리의 90% 이상을 사용한다면, 프로그램 메모리에 있는 모든 정보를 DB로 저장하고, 필요 없는 객체의 메모리가 해제 될 때까지 대기한다.



(그림 26) Linux 라우터와 IDRS 라우터 스루풋 비교

(그림 26)은 Xeon Dual 3.0G CPU, 1G RAM, 1G 이더넷 NIC에서 DDoS 탐지 및 대응 모듈을 설치한 Linux 라우터와 설치하지 않은 일반 Linux 라우터의 스루풋(throughput)을 비교한 것이다. 그림에서 일반 Linux 라우터보다 초당 250kbyte 정도의 성능저하를 보이지만, 전체적으로 봤을 때에는 일반 라우터와 크게 차이 나지 않는 다는 것을 알 수 있다.

7. 결론 및 향후 연구과제

본 논문에서는 기존의 DDoS 공격 대응에서 나타나는 문제점을 제시하였고 이에 관한 해결 방안을 제시하였다. 이러한 해결 방안은 현재 추진되고 있는 BcN에서 적용될 수 있을 것이다.

시그니처 기반의 탐지 기법과 통계기반의 탐지기법을 병행하여 DDoS 공격을 탐지한 결과 그 성능이 보다 향상 되는 것으로 나타났다. 또한 검증된 IP 테이블을 사용하여 기존과 다른 차별화된 대응 방법을 제시하였다.

하지만 이것에 대한 문제는 남아 있다. 만약 spoofing과 같이 검증된 IP 테이블에 있는 소스 IP 주소를 위장한 방법으로 공격을 한다면, 이 공격은 실제로 차단되지 않을 것이다.

이를 보안하기 위한 검증된 IP 테이블이 해쉬 함수를 사용하여 암호화된 방법으로 가지고 있다면 공격으로부터보다 안전 할 수 있을 것이다.

참 고 문 헌

[1] S.Gibson, "The Strange Tale of the Denial of Service Attacks Against GRC.COM" http://grc.com/dos/grcdos.htm, 2002.

[2] 안철수 보안 연구소 http://www.ahnlab.com

[3] Ari, I,Long, D.D.E."Managing flash crowds on the Internet", Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003.

[4] Eric Y.K Chan, H.W. Chan "An Intrusion Detection Router for Defending against Distributed Denial-of-Service (DDoS) Attacks, Parallel Architectures, Algorithms and Networks, 2004. Proceedings.

[5] 인터넷침해사고대응지원센터 http://www.krcert.or.kr

[6] 오픈 소스 IDS Snort, http://www.snort.org

[7] Laura Feinstein, Dan Schnackenberg, "Statistical Approaches to DDoS Attack Detection and Response", Information Survivability Conference and Exposition 2003.

[8] H. Jonathan Chao "PacketScore : Statistics-based Overload Control against Distributed Denial-of-Service Attacks", IEEE INFOCOM 2004.

[9] David K. Y. Yau, "Defending Against Distributed Denial-of-Service Attacks With Max-Min Fair Server-Centric Router Throttles", IEEE/ACM TRANSACTIONS ON NETWORKING, Vol.13, No.1, FEBRUARY, 2005.

[10] W.A.N.D, http://wand.cs.waikato.ac.nz

[11] WIDE-project, http://www.wide.ad.jp

[12] Rocky K.C Chang, "Defending against Flooding-Based Distributed Denial-of-Service : A Tutorial," IEEE/ACM TRANSACTIONS ON NETWORKING, Vol.13, No.1, FEBRUARY, 2005.

박 필 용

e-mail : coltpison@nate.com

2003년 경희대학교 컴퓨터공학과(학사)

2005년 경희대학교 대학원 컴퓨터공학과(공학석사)

2005년~현재 경희사이버대학 보안담당관

관심분야 : 네트워크보안, 차세대인터넷

홍 충 선

e-mail : cshong@khu.ac.kr

1983년 경희대학교 전자공학과(학사)

1985년 경희대학교 대학원 전자공학과(공학석사)

1997년 Keio University 정보통신공학전공(공학박사)

1988년~1999년 KT 통신망연구소 수석연구원/연구실장

1999년~현재 경희대학교 컴퓨터공학과 부교수

관심분야 : 초고속통신망, 이동네트워크, 네트워크보안, 센서네트워크라우팅

최 상 현

e-mail : csh@nca.or.kr

2002년 명지대학교 정보통신공학과(학사)

2002년~2005년 (주)파워콤 근무

2005년~현재 한국전산원 근무

관심분야 : 초고속통신망, 네트워크보안, 차세대인터넷