웹 서버에 대한 DDoS공격의 네트워크 트래픽 분석

이 철 호[†]·최 경 희[††]·정 기 현[†††]·노 상 욱[††††]

요 약

본 연구에서는 웹 서비스를 대상으로 한 다양한 DDoS 공격이 진행 중일 때 패킷들의 TCP 헤더 내에 SYN, ACK, 혹은 RST 등 다양한 플래그 값들이 설정된 패킷의 수와 총 패킷수와의 비율을 조사 분석하였다. 그 결과, 특정 플래그가 설정된 패킷 수의 비율이 각각의 DDoS 공격 유형에 따라서 매우 독특한 특성을 가짐을 발견하였다. 본 연구의 결과로 얻어진 이 특징들은 DDoS 공격을 조기에 탐지하는 기법과 시스템을 DDoS 공격으로부터 보호하는 기법 연구에 많은 도움을 줄 것으로 예상된다.

An Analysis of Network Traffic on DDoS Attacks against Web Servers

Cheolho Lee[†]·Kyunghee Choi[††]·Gihyun Jung[†††]·Sanguk Noh[††††]

ABSTRACT

This paper presents the analytic model of Distributed Denial-of-Service (DDoS) attacks in two settings: the normal Web server without any attack and the Web server with DDoS attacks. In these settings, we measure TCP flag rate, which is expressed in terms of the ratio of the number of TCP flags, i.e., SYN, ACK, RST, etc., packets over the total number of TCP packets, and Protocol rate, which is defined by the ratio of the number of TCP (UDP or ICMP) packets over the total number of IP packets. The experimental results show a distinctive and predictive pattern of DDoS attacks. We wish our approach can be used to detect and prevent DDoS attacks.

키워드 : 서비스거부공격(DoS), 분산서비스거부공격(DDoS), 웹(Web), 네트워크 트래픽(Network Traffic)

1. 서 론

Yahoo, eBay, E*Trade등 웹 사이트에 대한 분산 서비스 거부(Distributed Denial of Service : DDoS) 공격과 그에 대한 피해사례의 보고는 인터넷에 개방되어 있는 시스템들이 DDoS 공격에 매우 취약함을 보여주고 있다[20]. DDoS 공격은 짧은 시간 내에 대량의 패킷을 공격대상 호스트 또는 네트워크에 보냄으로써 그들이 제공하는 서비스를 일시적으로 중단시키거나 심지어 시스템의 고장을 유도하게 된다.

이들 DDoS 공격은 인터넷에 개방되어 있으면서 동시에 한정된 자원(네트워크의 대역폭, 시스템의 패킷 처리 용량, 시스템에 도착한 패킷의 처리를 위하여 이용되는 시스템의 제반 자원 등)을 가진 모든 시스템들을 쉽게 공격의 대상으로 하여 피해를 입힌다는 점에서 매우 심각하게 여겨지고 있으며, 이에 대한 다양한 연구가 진행되고 있다[4, 14, 16, 20, 22]. Zombi Zapper와 RID는 DDoS 공격이 진행되는 과정을 바탕으로 공격에 개입하는 시스템들 사이의 통신 경로를 차단하거나 혹은 공격에 이용되는 프로그램들의 존재를 검사해서 그들을 제거함으로써 공격 자체를 억제하는 예방적인 방법을 제시하고 있다[6-8]. 그리고, [23]에서는 CBQ (Class Based Queuing)로 서로 다른 대역폭을 갖는 몇 개의 큐를 구성하고, DDoS 공격시에 Source IP address가 spoofing되는 현상을 이용해서 Source IP address를 기준으로 DDoS 공격 트래픽으로 의심되는 패킷을 낮은 대역폭을 가진 큐로 보내고, 정상적인 서비스를 위한 패킷은 높은 대역폭을 가진 큐로 보냄으로써 상대적으로 DDoS 공격 트래픽의 양을 줄이는 방법을 사용한다.

그러나, Zombi Zapper와 RID가 제시하는 예방적인 방법들은 매우 빠르게 진화하는 해킹방법이나 DDoS 공격의 다양성을 고려할 때 한계가 있으며, [23]에서 제시하는 DDoS 공격 트래픽의 양을 줄이는 방법은 DDoS 공격 트래픽을 정상적인 서비스를 위한 트래픽과 구분하는 방법이 매우 모호할 뿐만 아니라 DDoS 공격의 특성과 본질에 대한 정확한 분석이 이루어지지 않았다는 점에서 한계가 있다. 따라서, 이 논문에서는 다양한 DDoS 공격 트래픽을 분석하

※ 이 논문은 2002년도 한국학술진흥재단의 지원에 의하여 연구되었음(KRF-2002-041-D00465).
† 준 회 원 : 아주대학교 정보통신전문대학원 정보통신공학과
†† 정 회 원 : 아주대학교 정보통신전문대학원 교수
††† 정 회 원 : 아주대학교 전자공학부 교수
†††† 정 회 원 : 가톨릭대학교 컴퓨터정보공학부 교수
논문접수 : 2003년 1월 13일, 심사완료 : 2003년 5월 9일

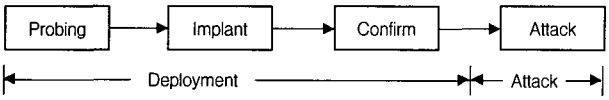
여 정상적인 서비스를 위한 트래픽과 확연히 구분되는 성질을 찾아낼 것이다.

특히, 최근의 DDoS 공격의 대상이 TCP를 기반으로 한 서비스에 집중되고 있다는 사실과 DDoS 공격의 90% 이상이 SYN Flooding 공격이며, SYN Flooding 공격은 TCP에 기반한 서비스에 치명적인 타격을 가할 수 있다는 점을 고려할 때[3], 웹 서버에 대한 DDoS 공격을 정확하게 분석하는 것은 DDoS 공격을 효과적으로 탐지하고 방어하는데 필수적이다. 이 논문에서는 전체 트래픽에서 특정한 유형의 트래픽이 차지하는 비율(Rate)을 이용해서 웹 서비스 트래픽과 DDoS 공격 트래픽을 분석했으며, 그 결과 웹 서버에 대한 DDoS 공격이 발생했을 때 웹 서비스 트래픽과 DDoS 공격 트래픽의 뚜렷한 차이점을 발견할 수 있었다.

이 논문의 2장에서는 DDoS 공격의 과정 및 구조를 살펴보며 3장에서는 제안된 여러 가지 DDoS 공격의 예방 및 탐지방법에 대해 소개하며, 4장에서는 본 논문에서 제안하는 트래픽 비율 분석법(Traffic Rate Analysis)에 대해 소개하고, 5장에서는 실험을 통해서 웹 서버에 대한 DDoS 공격이 발생했을 때 웹 서비스 트래픽과 DDoS 공격 트래픽의 차이점을 보인다. 마지막으로 6장에서는 결론 및 앞으로의 연구방향에 대해 언급한다.

2. 서비스 거부공격

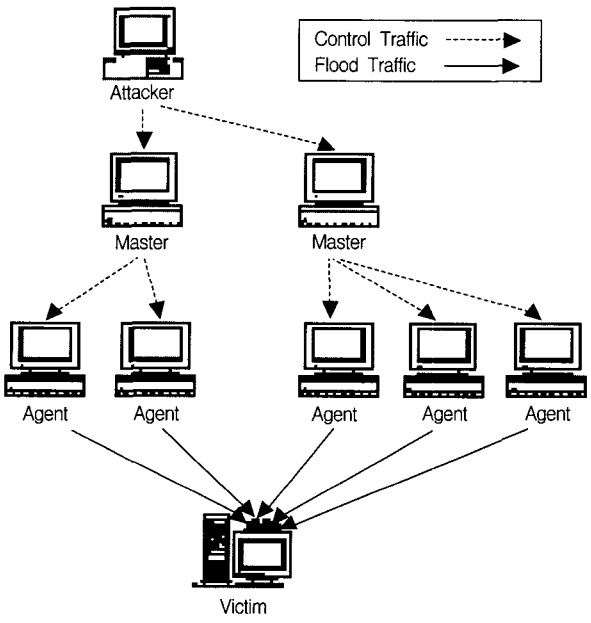
서비스 거부(Denial of Service : DoS) 공격은 1990년대 후반에 발견되기 시작했으며, 초기에는 Single Source-Single Target의 형태로 시작되었으나 최근 들어서, Multiple Source-Multiple Target의 형태, 즉 분산 서비스 거부공격으로 진화되었다[4, 5]. 분산된 형태의 공격을 위해서는 각각의 공격도구를 배포하고 설치하는 등의 공격준비과정이 필요하다. (그림 1)에서는 그 과정을 설명하고 있다.



(그림 1) DDoS 공격의 단계

Probing 단계에서 공격자는 DDoS 공격의 마스터(Master)와 에이전트(Agent)로 사용할 호스트를 선택하게 된다. 특정한 네트워크 취약성(Vulnerability)이 있는 호스트는 쉽게 침입이 가능하며 그러한 호스트들이 공격자의 목표가 된다. 그리고, Implant 단계에서는 침입한 호스트에 각각 마스터와 에이전트용 DDoS 공격도구를 설치하는 작업을 하며 Confirm 단계에서는 설치된 DDoS 공격도구를 실행해서 마스터와 에이전트 사이에 통신을하게 되는데 이 과정에서 주로 마스터와 에이전트 사이의 종속관계를 맺거나 IP Spoofing Level 등을 결정하는 작업을 수행한다. 또한, 최종 단계인 공격단계에서는 마스터에 설치된 DDoS 공격 도구가 자신의 에이전트들에게 공격명령을 전달하게 되고 각각의 에이전트들은 공격명령을 받으면 Victim을 향해서 일제히 공격 트래픽을 발생시키며 공격을 시작하게 된다. 최근에는 Code Red, Code Red II, Nimda 등의 자동화된 인터넷 웜(Worm)을 통해서 이러한 DDoS 공격 도구의 배포가 가능하다[4].

(그림 2)는 DDoS 공격이 시작될 때의 모습을 나타낸다. 공격자는 텔넷(Telnet) 또는 암호화된 전용 클라이언트 프로그램을 이용해서 마스터에 접속하고 공격명령을 전달한다. 그러면, 공격명령을 받은 마스터는 자신에게 속한 에이전트들에게 다시 공격명령을 하달하고, Agent들은 일제히 Victim을 향해서 공격을 시작한다.



(그림 2) DDoS 공격의 구조

3. 관련 연구

이번 장에서는 DDoS 공격 도구를 찾아내거나 또는 DDoS 공격 트래픽을 탐지하기 위해서 제안된 여러 가지 방법들에 대해서 알아보며 제안된 방법들과 본 논문을 통해서 제안하고자 하는 트래픽 비율 분석법(Traffic Rate Analysis)과의 차이점에 대해서 설명한다.

3.1 DDoS 공격 도구의 탐지

(그림 2)의 DDoS 공격 구조에서 보면, DDoS 공격에 의해서 발생되는 트래픽에는 DDoS 공격을 제어하기 위해 마스터와 에이전트간의 통신에 사용되는 제어 트래픽(Control traffic)과 실제 공격을 위해 에이전트가 Victim을 향해 뿜

어내는 공격 트래픽(Flood traffic), 이렇게 두 가지 종류가 있음을 알 수 있다.

현재 보안관련 기관에서 제공하고 있는 대부분의 DDoS 탐지 도구(Zombie Zapper, RID, … 등등)는 마스터와 에이전트 사이에서 발생되는 제어 트래픽을 잡아내거나, 통신을 위해 열어둔 특정한 통신포트를 찾아내거나, 또는 DDoS 공격 도구의 실행파일 이름을 찾는 등의 방법으로 DDoS 공격 도구의 존재유무를 판단해서 제거하는 방식이다[6-8]. 예를 들어서, RID의 경우는 Stacheldraht, TFN, Trinoo, TFN2K 등의 DDoS 공격 도구를 찾아내는데 각각의 공격 도구가 사용하는 통신포트를 발견해 냄으로써 탐지작업을 수행한다[8]. 하지만, 대부분의 DDoS 공격 도구는 그 소스코드가 공개되어 있을 뿐만 아니라 공격자가 변형해서 사용할 가능성이 매우 높다[10]. 심지어, RID의 경우는 <표 1>과 같이 마스터와 에이전트 간의 통신을 위한 기본 통신포트가 변경되지 않는 경우에만 DDoS 공격 도구를 탐지해 낼 수 있다고 명시하고 있다[10].

<표 1> RID가 DDoS 공격도구를 탐지할 수 있는 조건[10]

RID is a configurable remote DDOS tool detector which can remotely detect Stacheldraht, TFN, Trinoo and TFN2k ***if the attacker did not change the default ports.*** *By David Brumley.*

DDoS 공격의 진행과정 중에서 실제의 공격을 위한 준비단계에 해당하는 세 단계(Probing, Implant, Confirm)는 새로운 해킹기법의 개발 또는 인위적인 변형 등, 공격자의 의도에 따라 다양한 형태로 변화할 가능성이 매우 크다. 따라서, 이러한 공격 이전단계에서의 분석 및 탐지는 한계가 있다.

따라서, 이러한 한계를 극복하기 위해서는 DDoS 공격의 본연의 목적(네트워크 리소스 고갈)을 달성하기 위해서 DDoS 공격이 반드시 가질 수 밖에 없는 필연적인 특징을 분석해야 한다. 본 논문에서는 웹 서버를 대상으로 한 다양한 DDoS 공격에 대해서 공격 트래픽의 특성을 분석하고자 한다.

3.2 DDoS 공격 트래픽의 탐지

3.2.1 Source IP Address의 무작위성(Randomness)을 이용한 방법

대부분의 DDoS 공격 도구들은 공격자의 위치가 드러나는 것을 피하기 위해서 DDoS 공격 패킷의 Source IP Address를 Spoofing해서 사용한다. 여기에서 Spoofing되는 Source IP Address는 무작위로 생성되므로 이러한 특성을 이용하면 DDoS 공격을 탐지할 수 있을 것이다. 이 방법에 기반하여 Kolmogorov Complexity를 사용하는 DDoS 공격을 탐지하는 방법이 소개되었다[15]. 이 방법은 일반적인 상황에서의 네트워크 서비스는 패킷의 Source IP Address가 지역성(Locality)을 가지고 있으나, DDoS 공격이 발생하면 Flooding 패킷의 Source IP Address가 무작위로 spoofing 되기 때문에, Kolmogorov Complexity의 값이 서로 다르게 측정되는 성질을 이용한다.

그리고, 특정 subnet 내부로 들어가는 패킷 비율(to-rate)과 밖으로 나가는 패킷 비율(from-rate)의 불균형 현상(disproportion)을 이용해서 DDoS 공격을 탐지하는 방법도 있다[17, 18]. 임의의 Subnet P에 대해서 R(P)를 to-rate와 from-rate의 비율로 가정하면, $R_{min} < R(P) < R_{max}$인 경우를 정상적인 상태, 그 외의 경우를 DDoS 공격 상태로 인식한다. 이 방법 역시, DDoS 공격에 사용되는 공격 패킷의 Source IP Address가 무작위로 Spoofing되는 현상을 이용하며 DDoS 공격 패킷에 대한 Victim의 응답 패킷이 DDoS 공격이 시작된 방향이 아닌 다른 방향으로 빠져나가는 것을 모니터링함으로써, DDoS 공격을 탐지할 수 있다.

하지만, 이런 종류의 탐지방법은 공격자가 Souce IP Address에 대한 Spoofing의 정도를 변경하거나 실제의 주소를 사용하는 경우, 제대로 동작하지 못할 것임이 분명하다.

3.2.2 DDoS 공격 호스트의 MIB 통계를 이용한 방법

DDoS 공격의 탐지를 위한 다른 방법 중에 하나는 MIB (Management Information Base) 통계를 이용하는 방법이다[13]. NMS(Network Management System)를 이용하면 관리대상 호스트의 MIB 통계를 알 수 있는데, 만일, DDoS 공격을 위한 에이전트가 NMS 모니터링 범위내의 호스트에 설치되어 DDoS 공격이 시작되는 경우, NMS를 이용해서 해당 호스트의 MIB 통계 중에서 네트워크와 관련된 통계치의 변화를 모니터링 함으로써, DDoS 공격을 탐지할 수 있다. 하지만, 이 방법은 NMS 모니터링 범위 내에 반드시 DDoS 에이전트가 포함되어 있어야 한다는 단점이 있다.

3.2.3 TCP의 SYN과 FIN의 발생비율을 이용한 방법

DDoS 공격이 없는 경우 TCP 연결의 생성(Establishment)과 종료(Termination)는 재전송(retransmission)의 경우를 제외하면 거의 동일한 비율로 발생할 것이다. 하지만, SYN Flooding 공격이 발생하면 대량의 SYN 플래그를 가진 TCP 패킷이 급격이 증가하므로 SYN 플래그의 탐지비율이 FIN 플래그의 탐지비율 보다 월등히 많아지게 된다. 이러한 성질을 이용해서 [14]에서 사용된 'TCP SYN-FIN (RST) pairs' 방법과 [22]에서 사용된 'Intensity measure of SY11N segment' 방법은 DDoS 공격유형 중에서 SYN Flooding 공격을 효과적으로 탐지하는 방법을 제시하고 있다. 이 방법들은 위에서 언급한 바와 같이 TCP의 연결 생성(Connection Establishment)의 특성을 이용한다. 이 방법은 SYN Flooding 공격의 탐지에는 매우 유용하지만, 그 외의 다른 DDoS 공격 유형에 대해서는 탐지할 수 없다는 단점이 있다. 또한 [14]에서 제안된 'TCP SYN-FIN (RST) pairs'방법은 단순히 TCP

연결에 관련된 TCP 플래그들, 즉 SYN, FIN만을 대상으로 하여 SYN Flooding 공격을 탐지하는데 사용하지만, 본 논문에서 제안하고자 하는 트래픽 비율 분석법(Traffic Rate Analysis)은 TCP의 모든 플래그를 포함해서 TCP, UDP, ICMP 프로토콜의 비율도 함께 고려하므로 SYN Flooding 공격 뿐만 아니라 다양한 DDoS 공격에 대한 탐지에 이용할 수 있다.

4. 트래픽 비율 분석법 제안

본 논문에서는 웹 서버에 대한 DDoS 공격 트래픽을 분석하기 위해서 트래픽 비율 분석법(Traffic Rate Analysis)을 사용한다. 트래픽 비율 분석법이란 전체 트래픽에서 특정한 형태를 가진 트래픽의 비율(Rate)을 이용해서 트래픽을 분석하는 방법을 말하며 TCP 플래그 비율(TCP flag rate)과 프로토콜 비율(Protocol rate)로 구분된다. 다음의 수식은 TCP 플래그 비율(TCP flag rate)을 정의하고 있다.

$$R_{td}[Ki] = \frac{\sum flag(K)}{\sum TCP\ packets} \quad (inbound)$$

$$R_{td}[Ko] = \frac{\sum flag(K)}{\sum TCP\ packets} \quad (outbound)$$

TCP 플래그 비율(TCP Flag Rate)은 TCP 패킷만을 대상으로 하며, 특정한 TCP 플래그를 가진 패킷의 개수를 전체 TCP 패킷의 개수로 나눠서 구할 수 있다[1]. 수식에서 사용되는 TCP 플래그는 S, F, R, A, P, U, N이며, 각각 SYN, FIN, RST, ACK, PSH, URG, NULL[2]을 의미한다. 그리고, td는 트래픽 비율을 측정하기 위한 시간 간격을 의미한다. 예를 들어서, R_I[Si]는 Inbound 방향성을 갖는 트래픽, 즉 외부에서부터 측정 대상 호스트로 향하는 트래픽을 대상으로 1초 단위로 측정했을 때, 전체 TCP 패킷에서 SYN 플래그를 갖는 TCP 패킷의 발생비율을 의미한다.

$$R_{td}[(TCP|UDP|ICMP)i] = \frac{\sum(TCP|UDP|ICMP)packets}{\sum IP\ packets}(inbound)$$

$$R_{td}[(TCP|UDP|ICMP)o] = \frac{\sum(TCP|UDP|ICMP)packets}{\sum IP\ packets}(outbound)$$

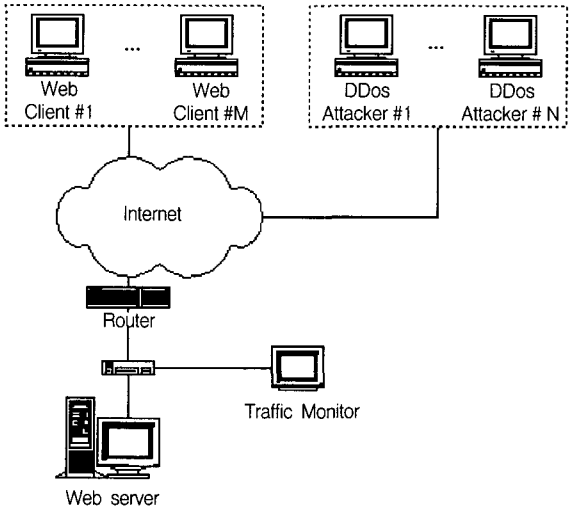
위의 수식에서 정의하는 프로토콜 비율(Protocol Rate)은 특정한 프로토콜(4계층 프로토콜 : TCP, UDP, 또는 ICMP)을 갖는 패킷의 개수를 전체 IP 패킷의 개수로 나눠서 구할 수 있다. 마찬가지로, td는 측정단위 시간을 의미한다. 예를 들어서, R_I[TCPo]는 Outbound 방향성을 갖는 전체 IP 패킷에서 TCP 패킷의 발생비율을 1초에 1회씩 측정한 결과값을 의미한다.

5. 실험 및 평가

이번 장에서는 트래픽 비율 분석법(Traffic Rate Analysis)에 의한 실험을 통해서 웹 서비스 트래픽과 DDoS 공격 트래픽의 특성을 살펴본다. 5.1에서는 실험환경에 대해서 살펴보고, 5.2에서는 웹 서비스 트래픽의 생성을 위한 방법에 대해서 알아본다. 그리고 5.3에서는 정상적인 웹 서비스 트래픽이 가지는 특성을 분석하며 5.4에서는 웹 서비스가 다양한 DDoS 공격을 받게 되는 상황을 분석할 것이다. 또, 5.5에서는 실험 결과를 정리하여 각각의 DDoS 공격 유형에 대해서 트래픽 비율 분석법을 적용했을 때의 네트워크 트래픽의 특징을 제시한다.

5.1 실험 환경

(그림 3)은 본 논문에서 실험을 위해서 사용한 네트워크의 구성을 나타낸 것이다. 이 실험에서 사용된 트래픽 모니터(Traffic Monitor)는 libpcap을 이용해서 개발했으며 Promiscuous 모드로 동작해서 네트워크 트래픽을 모두 캡쳐하여 트래픽 비율 분석법에 의해서 통계치를 계산하게 된다. 웹 클라이언트(Web Client)는 4개의 호스트로 구성하여 SPEC web99를 사용했고 DDoS 공격도구는 2개의 호스트로 구성하여 TFN2K를 사용했다[10, 12, 21]. 그리고, 모든 호스트는 리눅스(Linux 2.4.18)를 사용했고 웹 서버는 아파치(Apache)를 사용했다. SPECweb99는 보통의 웹 브라우저처럼 웹 서버를 향해서 다양한 형태의 HTTP 요청을 전송하고 웹 서버로부터 HTTP 응답을 전달받는 역할을 한다[12].



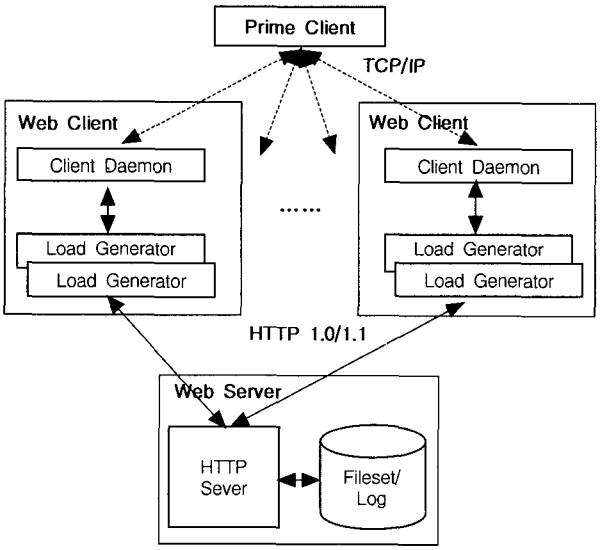
(그림 3) 실험을 위한 네트워크의 구성

1) TCP 패킷은 복수개의 flag를 가질 수 있으므로 TCP flag rate의 합은 1보다 큰 값이 될 수도 있다.

2) NULL은 어떤 flag도 set되지 않은 상태를 의미한다.

5.2 웹 트래픽의 생성

(그림 3)은 SPECweb99의 작동구조를 나타낸다. SPECweb99는 웹 브라우저와 동일한 역할을 수행하여 웹 트래픽을 생성시킨다.



(그림 4) SPECweb99의 작동구조

프라임 클라이언트(Prime Client)는 각각의 웹 클라이언트(Web Client)들의 클라이언트 데몬(Client Daemon)과 통신을 통해서 웹 트래픽의 생성을 지시한다. 그러면, 클라이언트 데몬은 다시 자신의 트래픽 생성기에게 웹 트래픽을 생성하도록 지시하며, 트래픽 생성기가 웹 서버를 향해서 HTTP 요청을 전송하고 해당 웹 서버로부터 그에 대한 HTTP 응답을 전달 받는다. 그리고, 웹 서버 측에서는 웹 클라이언트의 트래픽 생성기가 요청할 웹 문서들이 위치하는데 이것을 파일셋(Fileset)이라 하고, 각 파일셋이 웹 클라이언트로부터 요청될때 로그(Log)가 작성되어 SPEC web99의 수행 기록이 남는다.

[24-26]에서도 SPECweb99를 사용하여 웹 트래픽을 생성하고 웹 서버의 성능을 측정했다. <표 2>는 실제의 인터넷 환경에서 웹 서버에 전달되는 HTTP 요청의 종류와 그 비율을 나타낸 것이다.

<표 2> 실제 인터넷 환경에서의 HTTP 요청의 종류 및 비율[1, 27]

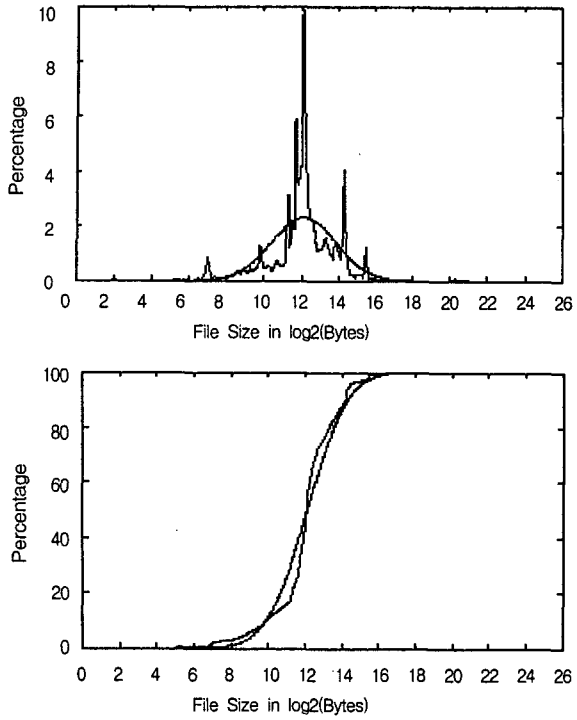
메소드(Method)	요청 비율(%)	데이터 전송 비율(%)
GET	99.88	99.62
HEAD	0.10	0.30
POST	0.02	0.08
Total	100.00	100.00

<표 2>에서 알 수 있듯이, GET 메소드가 대부분을 차지한다. 그리고 <표 3>은 SPEcweb99가 발생시키는 웹 트래픽에서의 HTTP 요청의 종류 및 그 분포를 보여준다.

<표 3> SPECweb99가 발생시키는 웹 트래픽에서의 HTTP 요청의 종류 및 비율[12]

HTTP 요청		비율(%)	종합(%)
GET	Static GET	70.0	95.2
	Standard Dynamic GET	12.45	
	Standard Dynamic GET (CGI)	0.15	
	Customized Dynamic GET	12.6	
POST	Dynamic POST	4.8	4.8

<표 2>와 <표 3>을 비교할 때, 실제의 웹 트래픽과 SPECweb99가 발생시키는 웹 트래픽이 거의 동일함을 알 수 있다.



(그림 5) 실제 인터넷 환경에서 HTTP로 요청되는 파일크기의 분포[1]

(그림 5)는 실제 인터넷 환경에서 웹 서버로 요청되는 파일크기의 분포를 나타낸 것이며, 왼쪽은 Frequency, 오른쪽은 Cumulative Frequency를 나타낸다. 가장 많이 요청되는 파일의 크기는 $2^{12}=4096(Bytes)$이며, 최대파일 크기는 약 $2^{17}=131072(Bytes)$이다.

<표 4>는 SPECweb99가 생성하는 웹 트래픽에서 요청되는 파일의 크기와 그 분포를 나타낸 것이다. (그림 5)와 <표 4>를 비교할 때, SPECweb99가 발생시키는 웹 트래픽에서 요청되는 파일크기의 분포는 실제로 인터넷에서 발견되는 것과 매우 유사함을 알 수 있다. 또한, [27]에서는 일반적인 웹 접속 패턴(Web Access Pattern)이 Zipf 분포를 따른다고 제시하고 있는데, SPECweb99 또한 Zipf 분포를

따르도록 설계되어 있다[7]. 위에서 살펴본 것처럼, SPECweb 99는 실제로 인터넷 상에서 발견되는 웹 트래픽과 거의 동일한 트래픽을 생성하며, 본 논문에서 웹 트래픽 생성기로 사용하기에 매우 적당하다.

〈표 4〉 SPECweb99가 발생시키는 웹 트래픽에서 요청되는 파일 크기의 분포[12]

Workload Class	Filesize requested	Rate
Class 0	Less than 1K	35 %
Class 1	Less than 10K	50 %
Class 2	Less than 100K	14 %
Class 3	Less than 1000K	1 %

5.3 웹 트래픽의 분석

웹 트래픽의 일반적인 성질을 실험을 통해서 분석하기 위해서는 실제의 인터넷 환경과 거의 동일한 웹 환경을 구성해야 할 것이다. 웹 트래픽은 TCP 연결을 기반으로 이루어 진다[11]. 예를 들어서, 사용자가 웹 브라우저에서 특정 웹 주소를 입력하면 웹 브라우저와 웹 서버는 TCP 연결을 맺고, 입력된 웹 주소가 HTTP의 GET 메소드를 통해서 대상 웹 서버에게 전달되고 그에 대한 응답으로 웹 서버는 요청 받은 문서의 내용을 클라이언트에게 전송하고 전송이 완료된 후에는 TCP 연결을 종료한다.

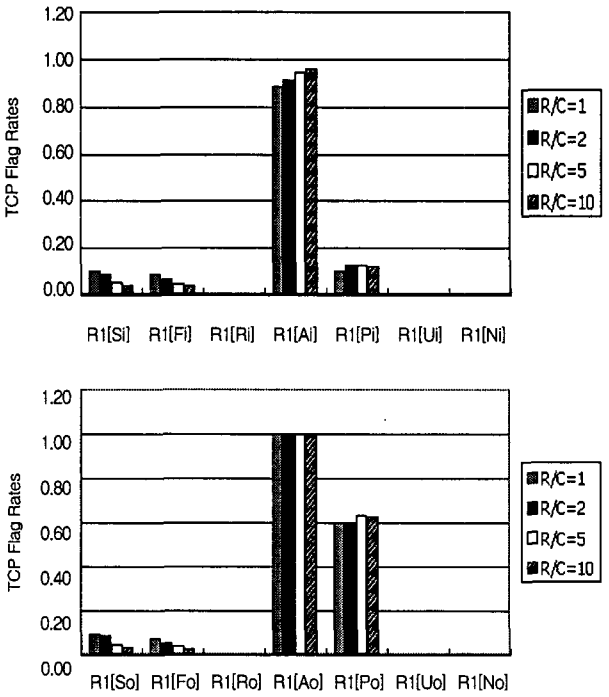
이처럼, 웹 서비스에 사용되는 HTTP는 TCP를 기반으로 한 프로토콜이므로, 하나의 TCP 연결에 수반되는 HTTP 요청의 개수(R/C : Requests per Connection)와 동시 연결 수(SC : Simultaneous Connections) 등의 값을 달리함에 따라서 다양한 형태의 웹 서비스 환경이 만들어질 수 있을 것이다.

〈표 5〉 실험에 사용된 웹 트래픽의 설정

설정 항목	사용된 값	비 고
동시 연결 수 (SC : Simultaneo us Connections)	5 10 50 100 150 200	동시에 웹 서버에 접근하는 웹 클라이언트의 수(웹 서버에 걸리는 부하의 정도)
연결 당 요청 수 (R/C : Requests per Connection)	1 2 5 10	하나의 연결 내에 서버로 전달되는 HTTP 요청의 개수(Persistent Connection 여부)

위와 같이, SC에 대해서 6개의 값을 적용했고, R/C에 대해서 4개의 값을 적용했으므로, 24가지의 다양한 웹 서비스 환경을 구성했다.

(그림 6)에서 SC와 R/C의 변화에 따른 TCP Flag Rate의 변화를 보여주고 있다. R/C(연결 당 요청 수)의 값이 증가함에 따라, R_1[Si], R_1[Fi], R_1[So], R_1[Fo]의 값이 조금씩 감소함을 알 수 있다. 그 이유는 R/C의 값이 증가하면 하나의 TCP 연결에 여러 개의 HTTP 요청과 응답이 발생하게 되므로, 상대적으로 TCP 연결과 관련되는 SYN, FIN 플래그를 가진 트래픽의 발생비율이 감소하기 때문이다. 또한, TCP Flag Rate는 기본적으로 비율의 개념을 사용하므로 SC(동시 연결 수)와는 무관하게 일정한 모습을 보인다.



(그림 6) 정상적인 웹 서비스 트래픽의 TCP Flag Rate 분포

(그림 6)에서 나타난 결과를 종합하면 다양한 웹 트래픽에서의 TCP Flag Rate는 대략적으로 다음과 같은 일정한 규칙을 가진다는 사실을 알 수 있다.

$$0.8 \le R_1[Ai] \le 1.0$$

$$R_1[Ao] \cong 1.0$$

$$R_1[Si] \le 0.1,\ R_1[So] \le 0.1$$

$$R_1[Fi] \le 0.1,\ R_1[F0] \le 0.1$$

여기에서 Inbound와 Outbound 모두 ACK 플래그의 비율이 거의 1.0에 가까운 수치로 높게 나타나는 이유는, 데이터를 받았을 때 그에 대한 확인정보(Acknowledgement)를 되돌려주는 TCP의 연결 지향적(connection-oriented) 특성 때문에 생기는 현상이다. 또한, SYN 또는 FIN등 TCP 연결에 관계되는 플래그들은 ACK에 비하면 상대적으로 그 비율이 매우 적게 나타난다. 이는 HTTP가 TCP 연결에 기초한 프로토콜이기 때문이며, Keep-alive(Persistent) 연결을 사용한 HTTP의 경우 R/C(연결 당 요청 수)의 값이 증가할수록 TCP 연결과 관련된 SYN과 FIN의 발생 비율이 더욱 더 낮게 나타난다[11].

5.4 DDoS 공격 트래픽의 분석

DDoS 공격 뿐만 아니라, 모든 네트워크 공격은 공격자의 위치를 최대한 숨기고 Victim에게는 최대한 큰 피해를 주는 것을 목적으로 한다. 특히, DDoS 공격은 네트워크 자원을 고갈시키는 것이 목적이므로 공격 패킷의 발생비율 또는 크기 등이 공격의 성공을 좌우하는 매우 중요한 요소이다[4]. 따라서, DDoS 공격 트래픽에는 분명히 이러한 공격 본위의 목적을 달성하기 위한 속성을 찾아볼 수 있을 것이다. 이런 성질을 이용해서 DDoS 공격 트래픽이 필연적으로 가질 수 밖에 없는 속성을 추출해낼 수 있다. 여기에서는, 웹 서비스가 다양한 유형의 DDoS 공격을 받는 상황을 각각 실험을 통해 분석할 것이다.

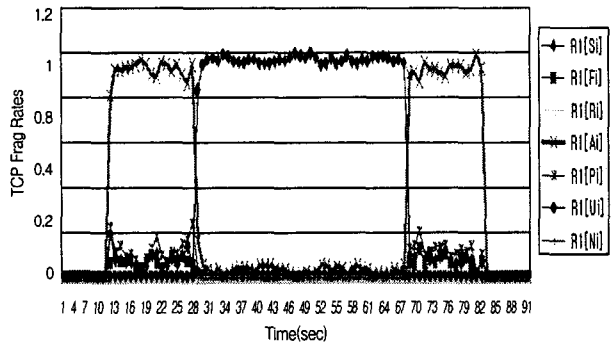
5.4.1 SYN Flooding 공격

웹 서비스에 대한 SYN Flooding 공격이 발생했을 때 트래픽 비율의 변화는 (그림 8)과 같이 나타난다. (그림 7)에서 제시된 실험결과는 SPECweb99를 통해 웹 트래픽을 발생시키고 1초에 1회씩 TCP Flag Rate를 측정한 결과이며, 중간정도의 시간대에서(26초~68초) 약 42초 동안 TFN2K를 사용해서 SYN Flooding 공격(공격포트는 임의선택)을 실시했다.

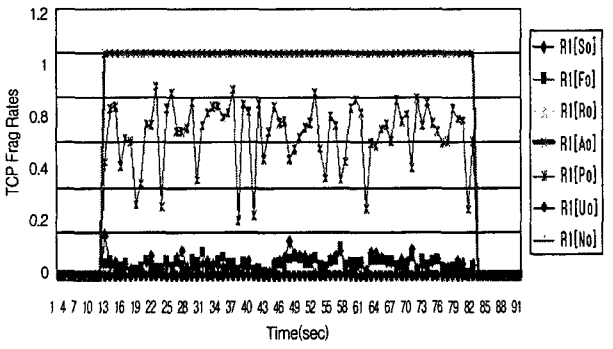
(그림 7)(a)에서 보면, SYN Flooding 공격이 진행되면서 (26초~68초)며, $R_1[Ai]$가 급격히 감소하고 $R_1[Si]$와 $R_1[Ui]$은 급격히 증가하는 것을 알 수 있다. 이것은, SYN Flooding 공격에 의해서 SYN 플래그와 URG 플래그를 가진 패킷이 증가함으로써 생기는 현상이다. 반면에, (그림 7)(b)에서 제시하는 Outbound 트래픽에 대한 TCP flag rate는 눈에 띄는 변화가 없다. 또한, (그림 7)(c)와 (그림 7)(d)에서 보듯이 웹 트래픽과 SYN Flooding 공격에 사용된 트래픽은 모두 TCP 트래픽이므로 Protocol Rate에도 아무런 변화가 없다.

5.4.2 UDP Flooding 공격

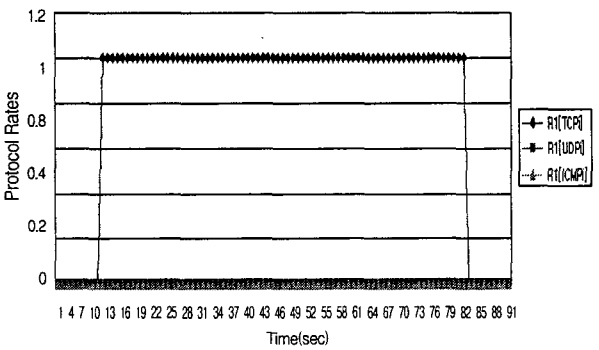
(그림 8)은 웹 서버에 대한 UDP Flooding 공격시의 TCP Flag Rate와 Protocol Rate을 분석한 결과이다(공격포트는 임의선택). UDP Flooding 공격은 18초~60초 구간에서 실시되었으며 TFN2K를 사용했다. (그림 8)(c)를 보면, UDP Flooding 공격이 시작되자 급격하게 $R_1[UDPi]$가 증가하고 $R_1[TCPi]$ 가 감소한다는 사실을 알 수 있다. 그리고, (그림 8)(a), (그림 8)(b)에서 TCP Flag Rate은 DDoS 공격에도 불구하고 거의 변함이 없는 것처럼 보이는데 이것은 $R_1[TCPi]$이 급격하게 저하된 상태에서 TCP Flag Rate의 분포를 살펴보았기 때문이다. 즉, UDP Flooding 공격 중에도 TCP 플래그의 구성 비율은 변함이 없다는 것을 의미한다.



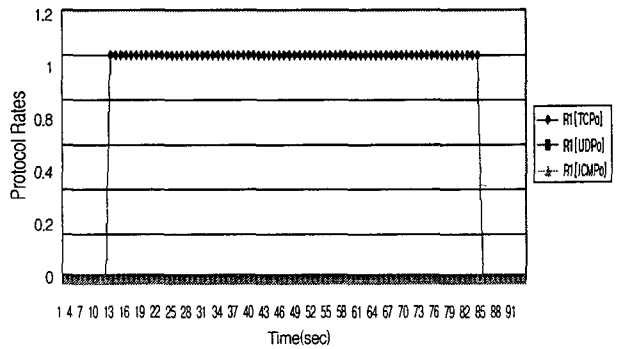
(a) Inbound TCP flag rates



(b) Outbound TCP flag rates

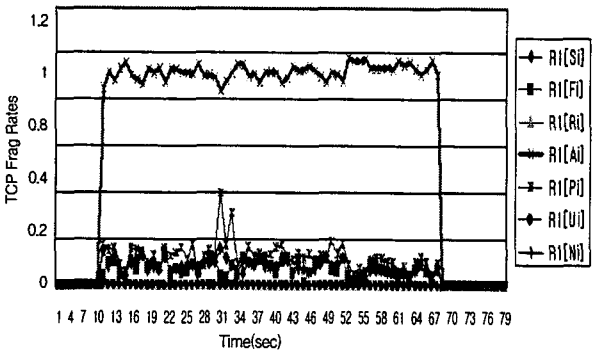


(d) Inbound protocol rates

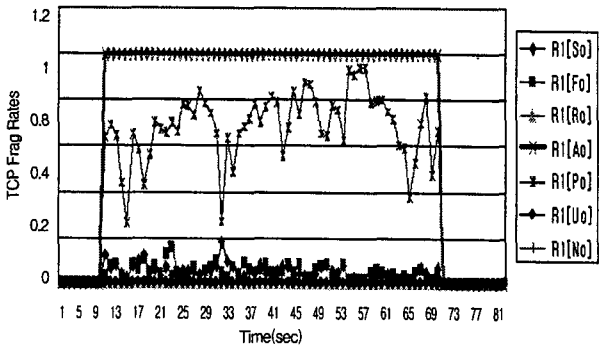


(c) Outbound protocol rates

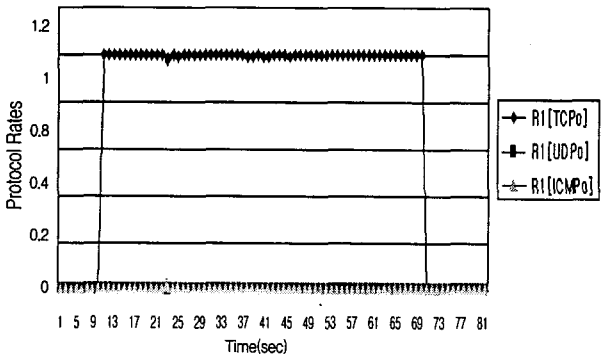
(그림 7) 웹 서버에 대한 SYN Flooding 공격시의 TCP Flag Rate 변화



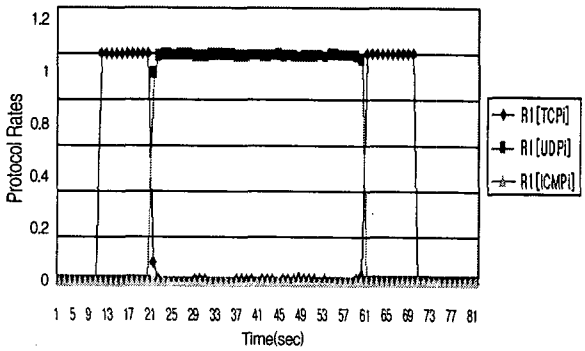
(a) Inbound TCP flag rates



(b) Outbound TCP flag rates

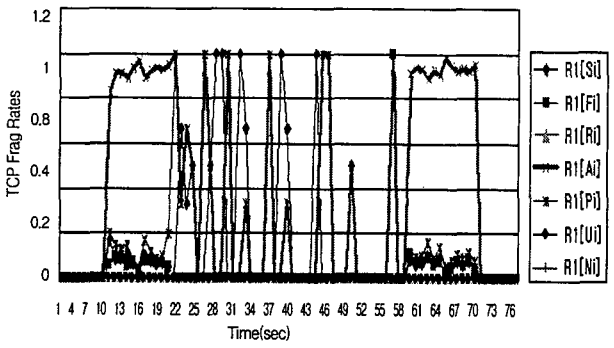


(c) Inbound protocol rates

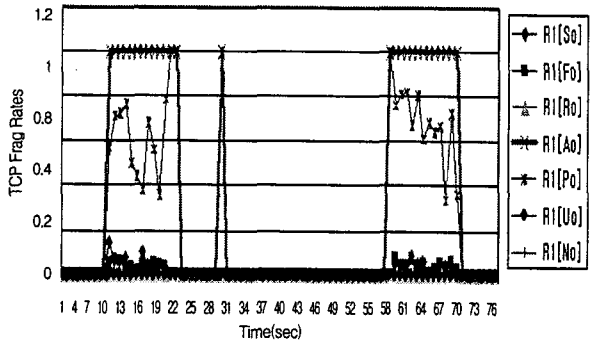


(d) Outbound protocol rates

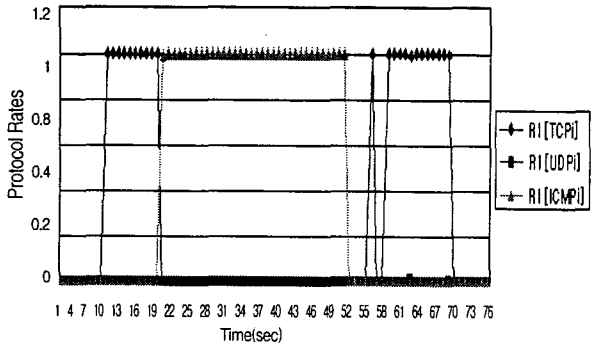
(그림 8) 웹 서버에 대한 UDP Flooding 공격시의 트래픽 비율의 변화



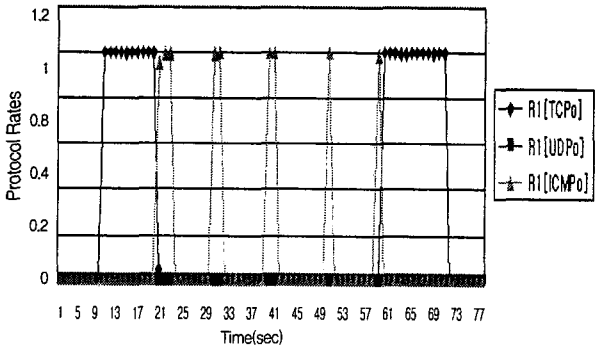
(a) Inbound TCP flag rates



(b) Outbound TCP flag rates



(c) Inbound protocol rates



(d) Outbound protocol rates

(그림 9) 웹 서버에 대한 ICMP Flooding 공격시의 트래픽 비율의 변화

5.4.3 ICMP Flooding 공격

(그림 9)은 ICMP Flooding 공격의 경우이며, 18초~53초 구간에서 DDoS 공격이 실시되었다. (그림 9)(c)에 보면, 다른 유형의 공격과 마찬가지로 R_1[TCPi], 즉 TCP 패킷의 발생비율이 급격하게 감소하고, R_1[ICMPIi], 즉 DDoS 공격 트래픽이 급격하게 증가함을 알 수 있다. (그림 9)(a)와 (그림 9)(b)는 TCP flag rate가 공격 구간에서 전체적으로 감소함을 보여주는데, 이것은 ICMP Flooding 공격에 의해서 웹 트래픽이 방해받고 있음을 확연히 보여준다. 또, (그림 9)(d)에서 보듯이 R_1[ICMPo]의 값이 공격 구간 내에서 간헐적으로 증가하는 이유는 ICMP Flooding 공격이 ICMP Ping 요청을 사용하므로, 그것에 대한 응답으로서 ICMP Ping 응답 패킷이 발생되기 때문이다.

5.4.4 기타 유형의 DDoS 공격

MIX Flooding 공격은 TFN2K에서 제공하며 SYN와 UDP, 그리고 ICMP의 세 가지 공격유형을 모두 혼합한 DDoS 공격의 유형이다[9]. (그림 10)(c)에서 Protocol Rate를 보면 TCP, UDP, ICMP가 동일한 비율로 섞여있음을 알 수 있다. 그리고, MIX Flooding 공격이 진행되는 동안(20초~56초) 다음과 같은 사실을 확인할 수 있다.

$$R_1[Ai] \cong 0$$

$$R_1[Si] = R_1[Ui] \cong 1.0$$

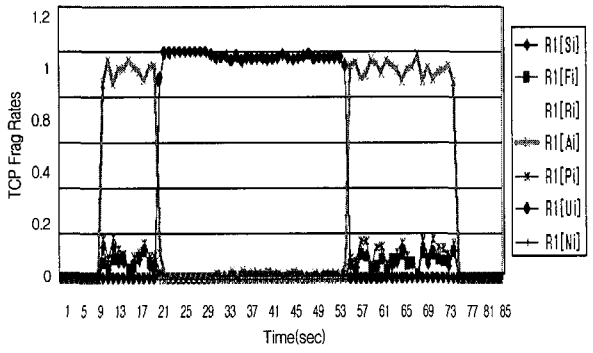
$$R_1[TCPi] = R_1[UDPi] = R_1[ICMPi]$$

여기에서, $R_1[Ai] \cong 0$와 $R_1[Si] = R_1[Ui] \cong 1.0$은 MIX Flooding 공격에 포함된 SYN Flooding 공격에 의한 것이다.

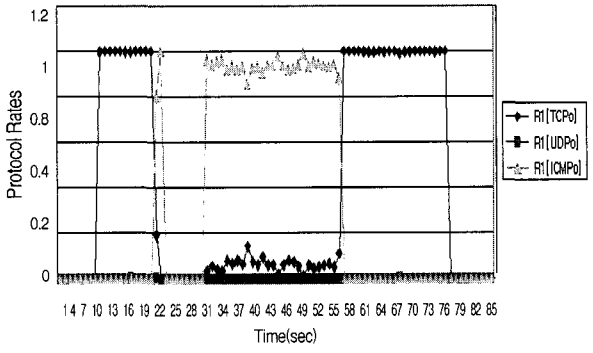
TARGA3 Flooding 공격은 TFN2K에서 제공하며 MIX Flooding 공격과 XMAS Flooding 공격을 혼합한 형태를 띄고 IP fragmentation의 약점을 이용한 공격 방법이다[9]. (그림 11)에서 Protocol Rate를 보면 MIX Flooding 공격과 유사함을 알 수 있고, TCP Flag Rate를 보면 XMAS 공격과 유사하게 모든 TCP 플래그가 설정되어 있음을 알 수 있다.

5.5 평 가

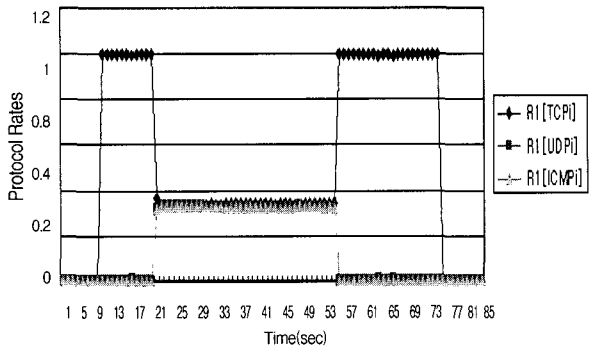
이번 장에서는 5.3과 5.4에서 제시된 실험결과를 통해서 발견된 웹 서비스 트래픽과 DDoS 공격 트래픽의 차이점을 종합 분석하고, 제시된 분석방법 (트래픽 비율 분석법)을 통해서 DDoS 공격을 효과적으로 탐지할 수 있는 가능성을 살펴본다. <표 6>은 5.3과 5.4에서 제시된 실험결과를 종합하여 나타낸 것이다. <표 6>에서, ↑은 그 값이 급격하게 증가함으로 나타내고, ↓은 그 값이 급격하게 감소함을 나타낸다. 또, ≈은 그 값이 동일함으로 의미한다. 5가지 DDoS 공



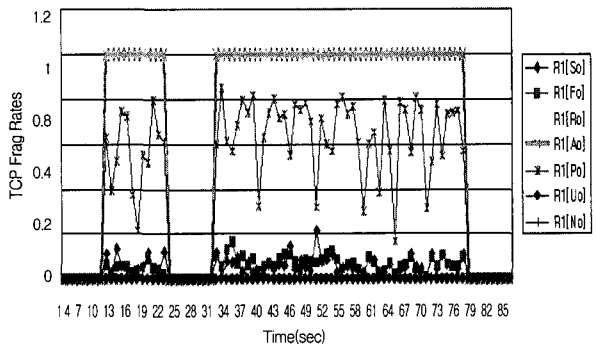
(a) Inbound TCP flag rates



(b) Outbound TCP flag rates

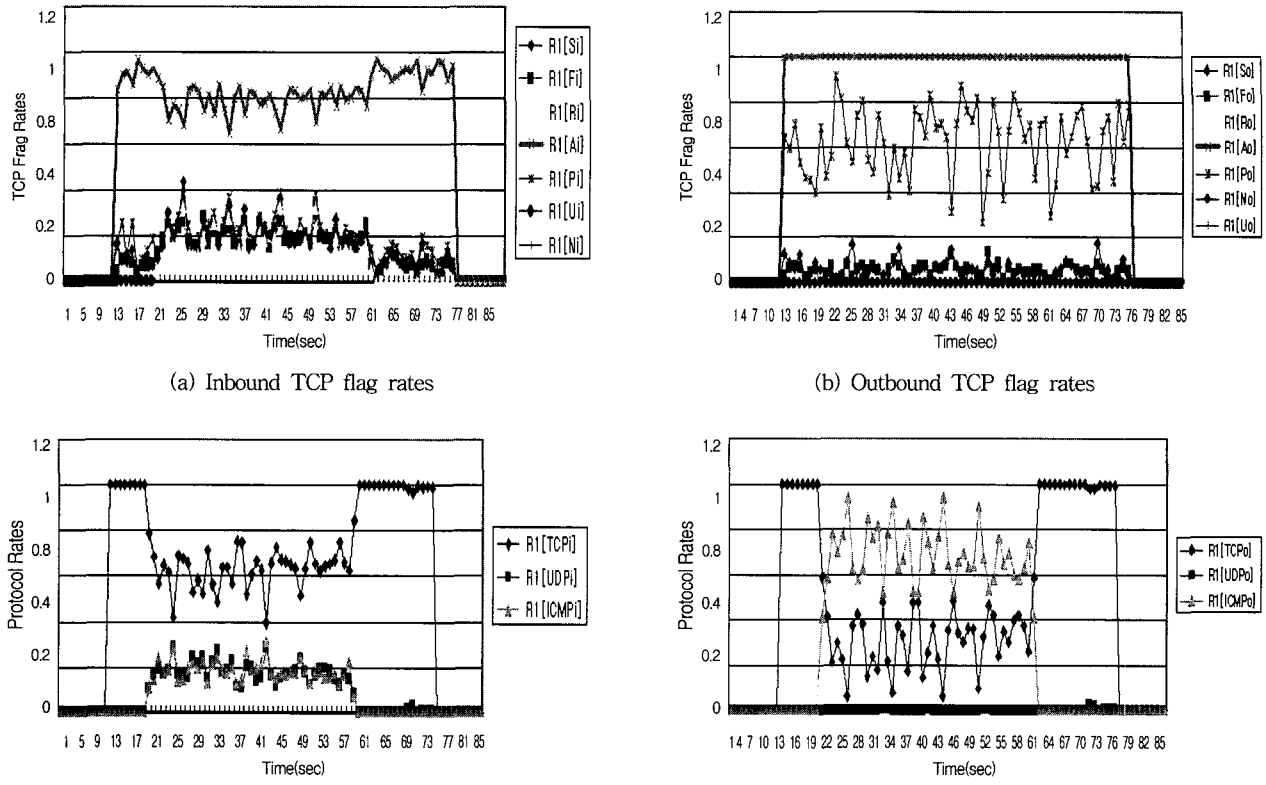


(c) Inbound protocol rates



(d) Outbound protocol rates

(그림 10) 웹 서버에 대한 MIX Flooding 공격시의 트래픽 비율의 변화



(a) Inbound TCP flag rates

(b) Outbound TCP flag rates

(c) Inbound protocol rates

(d) Outbound protocol rates

(그림 11) 웹 서버에 대한 TARGA3 Flooding 공격시의 트래픽비율의 변화

격 유형들은 모두 독특한 특징을 가지고 있으며, 각각은 DDoS 공격이 없는 상황에서의 웹 트래픽과 확연히 다른 모습을 보인다. 또한, 네트워크 트래픽의 변화되는 모습을 기반으로, 어떤 DDoS 공격이 발생했는지 판단하고 그 징후(Symptom)를 빠르게 포착할 수도 있을 것이다.

〈표 6〉 DDoS 공격 유형에 따른 트래픽 비율의 변화

Types of attacks	Inbound		Outbound	
	TCP flag rates	Protocol rates	TCP flag rates	Protocol rates
SYN	R[Si]↑, R[Ui]↑, R[Ai]↓			
UDP		R[UDPi]↑, R[TCPi]↓		
ICMP		R[ICMPi]↑, R[TCPi]↓		R[ICMPo]↑, R[TCPo]↓
MIX	R[Si]↑, R[Ui]↑, R[Ai]↓	R[TCPi] ≈, R[UDPi] ≈, R[ICMPi] ≈		
TARGA3	R[Si] ≈, R[Ui] ≈, R[Fi] ≈, R[Pi] ≈, R[Ui] ≈			

6. 결 론

본 논문에서는 DDoS 공격 트래픽을 분석하기 위한 방법으로 전체 패킷에서 특정한 유형의 패킷이 차지하는 비율을 표현하는 트래픽 비율 분석법(Traffic Rate Analysis)을 제안했으며, 웹 서비스에 대해서 DDoS 공격이 발생하지 않은 경우와 DDoS 공격이 발생할 경우로 각각 나누어서 주기적으로 트래픽 비율(TCP Flag Rate, Protocol Rate)을 모니터링 하였다. 그 결과, 웹 서비스 트래픽에 대해서 DDoS 공격이 없는 상태의 트래픽 비율과 다양한 DDoS 공격이 발생할 때의 트래픽 비율은 TCP 플래그 비율(TCP Flag Rate)과 프로토콜 비율(Protocol Rate)에서 모두 뚜렷한 차이가 있음을 밝혔다. 따라서, 본 논문에서 제안된 트래픽 비율 분석법을 적절히 이용하면 DDoS 공격의 효과적인 탐지가 가능할 것으로 예상된다. 하지만, DDoS 공격이 없는 상황에서의 웹 트래픽과 다양한 DDoS 공격이 발생했을 때의 웹 트래픽을 구분함에 있어서, 그 명확한 기준을 수치화하여 제시하지 못한 점에서 본 논문의 한계가 있다 하겠다. 향후 과제로는 기계 학습(Machine Learning) 등의 방법을 도입하여 각각의 DDoS 공격을 정확하게 판단하기 위한 기준을 설정하는 방법을 연구할 것이다.

참 고 문 헌

[1] M. Arlitt and T. Jin, "Workload Characterization of the 1998

World Cup Web Site," IEEE Network, Vol.14, No.3, pp. 30-37, May/June, 2000.
[2] V. Paxson, "Growth Trends in Wide-Area TCP Connections," IEEE Network, Vol.8, No.4, pp.8-17, July, 1994.
[3] David Moore, Geoffrey M. Voelker and Stefan Savage, "Inferring Internet Denial-of-Service Activity," In Proceedings of the 10th USENIX Security Symposium, pp.9-22, August, 2001.
[4] Kevin J. Houle, George M. Weaver, "Trends in Denial of Service Attack Technology," CERT Coordination Center, October, 2001.
[5] Rich Pethia, "Internet Security Trends," CERT Coordination Center, February, 2001.
[6] NIPC(National Infrastructure Protection Center), "find_ddos," http://www.nipc.gov/wanings/advisories/2001/01-005.htm, 2001.
[7] BindView's RAZOR Security Team, "Zombie Zapper," http://razor.bindview.com/tools/ZombieZapper_form.shtml, 2001.
[8] TheoryGroup, "Remote Intrusion Detector(RID)," http://www.theorygroup.com/Software/RID, 2001.
[9] Dave Dittrichs, "Dave Dittrichs Homepage," http://www.washington.edu/People/dad, 2002.
[10] Packet Storm, "DDoS Attack Tools," http://packetstorm.widexs.nl/distributed/indexdate.shtml, 2002.
[11] Fielding, R., Mogul, J., Frystyk, H., Masinter, L., Leach, P. and Bernerslee, T., "Hypertext Transfer Protocol - HTTP/1.1," Tech. Rep. RFC 2616 IETF, http://www.ietf.org/rfc/rfc2616.txt, June, 1999.
[12] Standard Performance Evaluation Corporation, "SPEC web99 benchmark," http://www.spec.org/osg/web99, August, 2000.
[13] Joao B. D. Cabrera, "Proactive Detection of Distributed Denial of Service Attacks using MIB Traffic Variables A Feasibility Study," In Proceedings of International Symposium of Integrated Network Management, May, 2001.
[14] Haining Wang, Danlu Zhang and Kang G. Shin, "Detecting SYN Flooding Attacks," In Proceedings of IEEE INFOCOM '02, 2002.
[15] A. B. Kulkarni, S. F. Bush and S. C. Evans, "Detecting Distributed Denial-of-Service Attacks Using Kolmogorov Complexity Metrics," GE Research and Development Center, December, 2001.
[16] Allen Householder, Art Manion, Linda Pesante and George M. Weaver, "Managing the Threat of Denial-of-Service Attacks," CERT Coordination Center, October, 2001.
[17] Thomer M. Gil and Massimiliano Poletto, "MULTOPS : a data-structure for bandwidth attack detection," In Proceedings of the 10th USENIX Security Symposium, pp.23-38, August, 2001.
[18] Thomer M. Gil, "MULTOPS : a data structure for denial-of-service attack detection," Master thesis, Division of Mathematics and Computer Science, VRIJE University, December, 2000.
[19] Alan Piszcz, Nicholas Orlans, Zachary Eyler-Walker and David Moore, "Engineering Issues for an Adaptive Defense Network," MITRE Technical Report, June, 2001.
[20] L. Garber, "Denial-of-Service Attacks Rip the Internet," IEEE Computer, pp.12-17, April, 2000.
[21] The Tcpdump Group, "LIBPCAP 0.6.2," http://www.tcpdump.org, June, 2001.
[22] Pars Mutaf, "Defending against a Denial-of-Service Attack on TCP," In Proceedings of the 2nd International Workshop on Recent Advances in Intrusion Detection(RAID '99), 1999.
[23] Frank Kargl, Joern Maier and Michael Weber, "Protecting Web Servers from Distributed Denial of Service Attacks," In Proceedings of the 10th International Conference on World Wide Web, April, 2001.
[24] Neil Macehiter, "Web Server Performance and Scalability," Zeus Technology, November, 2000.
[25] David J. Morse, Yi-Ming Xiong, "Exploring the Impact of Hyper-Threading on Web Workloads," Dell Computer Corporation, August, 2002.
[26] WinCom System, "Enhancing Web Performance with the WinCom Switching Server and Storage Area Networks," Application Note, January, 2002.
[27] Venkata N. Padmanabhan, Lili Oiu, "The Content and Access Dynamics of a Busy Web Site : Findings and Implications," ACM SIGCOMM '00, August, 2000.

이 철 호

e-mail : cheolholee@cesys.ajou.ac.kr
2002년 아주대학교 정보통신대학 정보및컴퓨터공학부(학사)
2002년~현재 아주대학교 정보통신전문대학원 정보통신공학과 석사과정
관심분야 : 네트워크 보안, 분산 시스템 등

최 경 희

e-mail : khchoi@madang.ajou.ac.kr
1976년 서울대학교 사범대학 수학교육과(학사)
1979년 프랑스 그랑데꼴 Enseeiht 정보공학과(공학석사)
1982년 프랑스 Paul Sabatier 정보공학과(공학박사)
1982년~현재 아주대학교 정보통신전문대학원 교수
관심분야 : 운영체제, 분산시스템, 실시간 및 멀티미디어 시스템 등

정 기 현

e-mail : khchung@madang.ajou.ac.kr

1984년 서강대학교 공과대학 전자공학과 (학사)

1988년 미국 Illinois 주립대 EECS(공학석사)

1990년 미국 Perdue 전기전자공학부(공학박사)

1991년~1992년 현대전자 반도체 연구소

1993년~현재 아주대학교 전자공학부 교수

관심분야 : 컴퓨터구조, VLSI설계, 멀티미디어 및 실시간 시스템 등

노 상 욱

e-mail : sunoh@catholic.ac.kr

1987년 서강대학교 이과대학 생명과학과 (학사)

1989년 서강대학교 공과대학 컴퓨터학과 (공학석사)

1999년 미국 University of Texas at Arlington, Computer Science(공학박사)

1989년~1995년 국방과학연구소 연구원

1995년~1999년 University of Texas at Arlington, Department of Computer Science an Engineering, Research Assistant

2000년 Oregon Graduate Institute of Science and Technology, Department of Computer Science and Engineering, Postdoctoral Fellow

2000년~2002년 University of Missouri-Rolla, Department of Computer Science, Assistant Professor

2002년~현재 가톨릭대학교 컴퓨터정보공학부 교수

관심분야 : Knowledge Management, Intelligent Agent, Multi-Agent System, Distributed Real-Time System 등