서비스 거부공격이 TCP 상태 전이에 미치는 영향

신범석*, 이재현*, 권경희**
*단국대학교 전자계산학과
**단국대학교 전자계산학과
e-mail:sbrock@dku.edu, wogusking@dankook.ac.kr,
khkwon@dku.edu

The Effect of DoS(Denial of Service) Attack on TCP State Transition

Bum-Suk Sin*, Jae-Hyun Lee*, Kyung-Hee Kwon**
*Dept of Computer Science, Dan-Kook University
**Dept of Computer Science, Dan-Kook University

요 약

서비스 거부공격(Denial of Sevice)이란 서버의 자원을 고갈시켜 더이상 정상적인 서비스를 할 수 없도록 하는 공격이다. DoS 공격 중에서 SYN Flooding DoS Attack을 받은 웹 서버는 외부로부터 들어온 공격 패킷에 의해 back log를 소모하게 된다. 그 결과 정상적인 연결 요청에 대해 서비스를 제공하지 못하게 된다. Dos 공격에 관한 다양한 연구가 진행되고 있지만, 본 논문에서는 서비스 거부공격이 TCP 상태 전이에 미치는 영향에 관한 연구를 하였다. 웹 서버의 Tcp 상태정보를 얻기 위해 GetTcpinfo 프로세스를 실행한 후 정상적인 접속을 시도해 보고 정상적인 접속이 진행되고 있는 상태에서 DoS 공격을 시도한다. GetTcpinfo 프로세스에 의해 파일로 저장된 TCP 상태전이 값을 분석하여 DoS 공격이 TCP 상태 전이에 미치는 영향에 대해 알아본다.

1. 서론

초고속 인터넷의 보급은 교육, 문화, 금융 등 사회 전반적인 분야에 걸쳐 인간의 생활양식을 변화시켰다. 다양한 웹서비스의 보급을 통하여 다양한 IT 사업이 등장하였고, 그와 더불어 중소형 웹서버의 운영이 빠른 속도로 확산되었다. 이러한 서버시장의 확산은 해커들의 주요 공격대상이 되었다. 서버 운영자들은 이러한 공격에 대안으로 많은 장비들과 소프트웨어의 추가공급을 하고 있지만, 아직까지 완벽하게 방어하기에는 역부족이다. 본 논문에서는 이러한 해커의 공격들 중에 서버의 서비스 중단을 초래하는 공격인 DoS (Denial of Service)공격에 대하여 연구하였다.

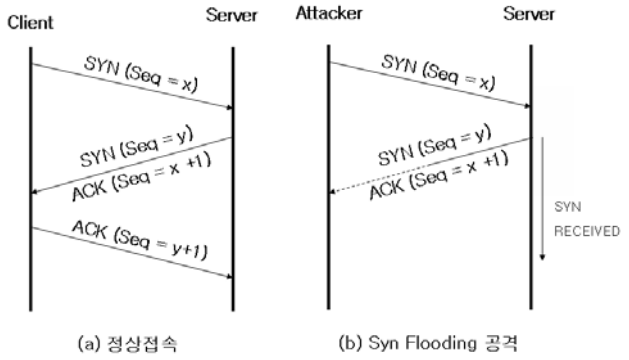
서비스거부공격(DoS;Denial Of Service)은 대역폭, 프로세스 처리 능력, 기타 시스템 자원을 고갈시킴으로써 정상적인 서비스를 할 수 없도록 하는 공격 형태이다. 대역폭을 목표로 한 공격은 시스템에 대량의 TCP, UDP 또는 ICMP 패킷을 보내는 공격이다. 프로세스 처리 능력 등 시스템 자원 고갈을 목표로 하는 공격에는 TCP 옵션 변경, 비정상적인 패킷 사이즈 등 비정상적인 패킷을 송신하여 자원을 고갈시키거나 비정상적으로 시스템을 멈추게 한다.[1]

기존의 DoS 공격 탐지는 웹 서버로 들어오는 트래픽 분석을 통해 DoS 공격의 유무를 판단했다.[2][3] 하지만 본 논문에서는 DoS 공격에 대한 웹 서버 내부의 TCP 상태 변화를 분석한다.
정상적인 웹 트래픽이 발생했을 때의 TCP 상태 변화와 SYN Flooding DoS 공격이 발생 했을 때의 TCP 상태변화를 비교하여 DoS 공격이 웹 서버의 TCP 상태 전이에 미치는 영향을 분석한다.

2. SYN Flooding DoS Attack

SYN Flooding Dos Attack은 서버와 클라이언트가

연결 설정을 하기 위해서 수행하는 3-Way Handshake를 완료하지 못하기 때문에 발생한다. [그림 1]의 (a)는 정상적인 3-way handshake를 보여주고 있고, [그림 1]의 (b)는 Syn Flooding 공격 시의 3-way



[그림 1] 3-way handshake

handshake를 보여주고 있다. Attacker는 client의 ip address를 변조하여 보낸다. 서버는 클라이언트의 ACK를 받을 수 없기 때문에 DoS 공격이 발생한다. 이때 SYN RECEIVED 상태의 연결을 저장하는 곳이 바로 BACK LOG QUEUE 이다. 서버의 BACK LOG QUEUE는 DoS 공격이 발생하거나, 동시에 다량의 정상 접속요청이 발생할 경우에 MAX 값을 가지게 된다. 이것은 곧 서버의 자원 고갈을 초래한다. 서버는 연결 요청을 받아들일 만한 여분의 BACK LOG QUEUE가 없기 때문에 정상적으로 연결 요청을 하는 클라이언트의 패킷을 DROP하게 된다.

일반적으로 SYN RECEIVED 상태의 연결은 75초가 지나면 BACK LOG QUEUE에서 자동으로 제거된다.[4] 하지만 75초 안에 BACK LOG QUEUE에 들어가 있는 SYN RECEIVED 상태의 TCP전이가 MAX 값까지 도달하고 그 상태가 계속 유지 된다면 BACK LOG QUEUE에 들어 있는 연결을 제거하더라도 서버는 계속해서 서비스를 할 수 없게 된다.

3. 실험 및 분석

3.1 Testing

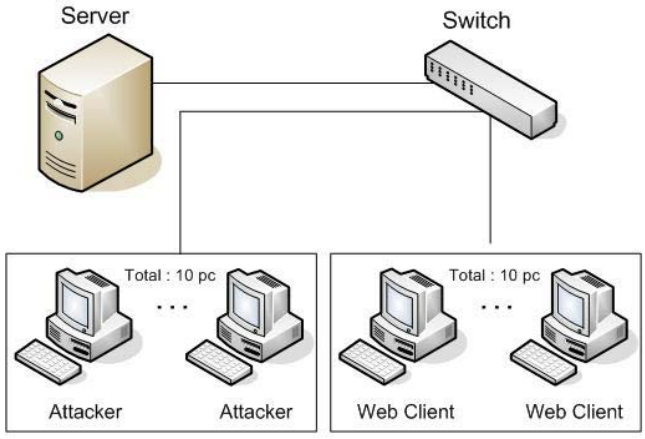
SYN flooding Dos 공격 분석을 위한 환경은 웹을 기반으로 하는 서버/클라이언트 구조를 기본으로 설정하였다. 서버와 네트워크 장비의 구성은 표[1]과 같이 설정하였으며, 네트워크 구조는 [그림 2]와 같이 구축 하였다.

클라이언트는 정상적인 접속을 하는 Web Client와 SYN Flooding 공격을 하는 Attacker로 분리하여 설치하였다.

Web client는 프로세스를 생성하여 웹 서버에게 정상적인 페이지를 주기적으로 요청하는 프로그램을 제작하였고, Attacker는 IP SPOOFING을 하여 서버에 접속하도록 제작하였다. 또한 서버의 TCP 상태정보

서버	
CPU	Pentium 3 - 700
MEMORY	256M
리눅스버젼	Linux RedHat 7.3
커널버젼	2.4.18-3
웹 서버	httpd-2.0.52(Apache)
네트워크 장비	
Switch	Catalyst 2948G-GE-TX (Cisco)
Bandwidth	100Mbps

[표 1] 서버와 네트워크 장비 구성



[그림 2] 네트워크 구성

를 실시간으로 파일에 저장하기 위하여 GetTCPinfo 프로그램을 제작하였다. GetTCPinfo는 TCP 상태 정보를 얻기 위한 프로그램으로서 서버의 상태를 5개로 분리하였으며 5초 간격으로 서버에 접속한 각각의 TCP 연결에 대한 통계를 작성하고 파일로 저장한다.[5][6]

실험은 정상적인 접속이 이루어질 때와 정상적인 접속과 공격이 병행해서 일어나는 두 가지 경우로 나누어 진행하였다. 이렇게 진행한 이유는 정상적인 접속일 때의 TCP 상태 전이와 정상적인 접속이 이루어지는 도중에 DoS 공격을 받았을 때의 TCP 상태 전이를 구분하기 위해서이다. 아래 표[2]는 정상적인 접속에서의 프로세스 수와 지속시간을 나타내며, 표

[3]은 정상접속과 공격이 병행 되었을 때 두 종류의 프로세스가 모두 종료되었을 때까지 소요된 시간과 각각의 프로세스 수를 나타낸 것이다.

	정상 프로세스 수	지속시간(초)
실험 1	40	879
실험 3	60	965

[표 2] 정상 접속일 경우

실험 1은 정상적인 접속요청 프로세스를 40개 실행한 실험한 것이고, 실험 3은 정상적인 접속요청 프로세스 60개를 실행한 실험이다.

	정상 프로세스 수	공격 프로세스 수	지속시간 (초)
실험 2	40	40	728
실험 4	60	60	984

[표 3] 공격과 병행한 경우

실험 2는 정상적인 접속요청 프로세스를 40개 실행하고 120초 후에 공격 프로세스 40개를 실행하고 시작해서 301초에 공격을 중지한 실험이고, 실험 4는 정상적인 접속요청 프로세스를 60개 실행하고 120초 후에 공격 프로세스 60개를 실행하고 303초에 공격 프로세스를 중지한 실험이다. 단, 실험 2와 실험 4에서 공격만 중지하고 정상적인 접속은 그대로 진행된다.

3.2 결과 분석

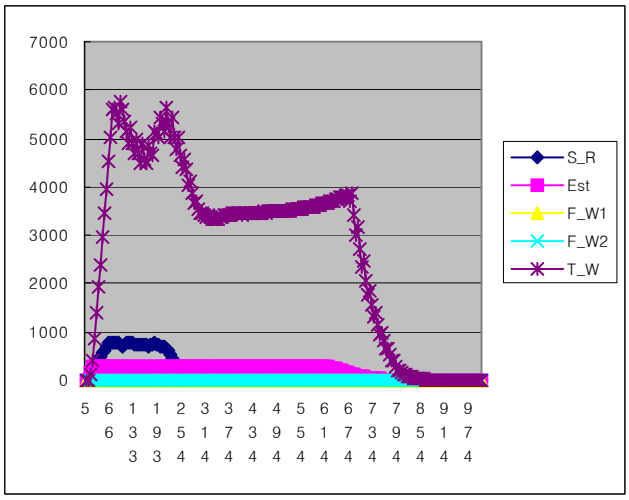
[그림 3], [그림 4], [그림 5], [그림 6]은 3.1에서 수집한 데이터를 그래프로 나타낸 것이다.

(실험1)-[그림 3]에서 초기에 대량의 연결 요청이 있었음을 알 수 있다. 그 결과 ESTABLISHED의 값이 큰 폭으로 증가하며 MAX(웹 서버의 최대 동시접속자 수)값에 도달했을 때부터 접속 요청이 들어오지 않을 때까지 거의 일정한 수준으로 MAX값을 유지한다. 반면 SYN RECEIVED는 초기에는 증가하지만 중반으로 넘어가면서 그 수가 현저히 떨어지고 있다. 또한 TIME_WAIT은 절정에 도달 했을 때부터 값의 증감이 뚜렷이 나타나다가 SYN RECEIVED 수가 줄어들수록 일정한 값을 유지한다.[7] 그러다가 ESTABLISHED 수가 줄어들자 급격하게 줄어드는 것을 볼 수 있다.

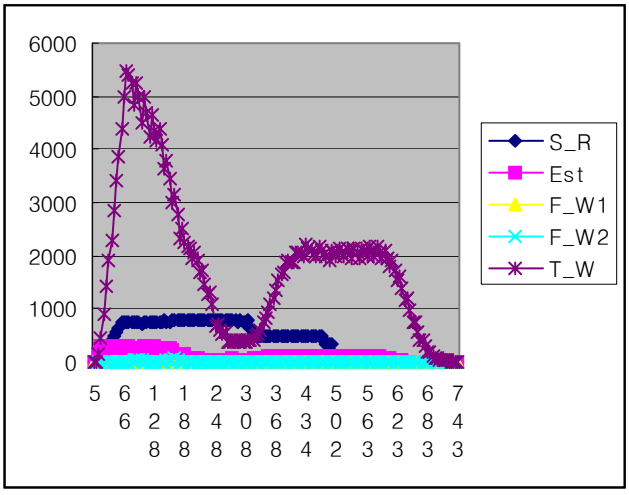
(실험2)-[그림 4]에서 보면 [그림 3]과 달리 SYN RECEIVED의 값과 ESTABLISHED의 값이 거의 반대의 경향으로 진행하고 있는 것을 볼 수 있다. 또한 TIME WAIT의 값이 128초에서 떨어지기 시작한다. 이 값은 300초에서 최저값이 된다.

(실험3)-[그림 5]에서 실험 초기에는 모든 값들이 증가한다. 85초에서 165초까지 ESTABLISHED와 SYN RECEIVED는 일정한 값을 유지하지만 TIME WAIT은 85초에서 165초 사이에 값이 감소하다가 165초 이후에는 다시 값이 증가하게 된다. 이 시간에 SYN RECEIVED는 서서히 감소한다. [그림 3]과 다르게 [그림 5]에서 TIME WAIT은 최대값이 되었을 때 수치의 변화가 심하게 나타난다. 하지만 ESTABLISHED는 725초 까지 일정한 값을 유지하게 된다. 725초 이후에 ESTABLISHED 값은 감소하고 이 시기에 TIME WAIT 값은 큰 폭으로 감소한다.

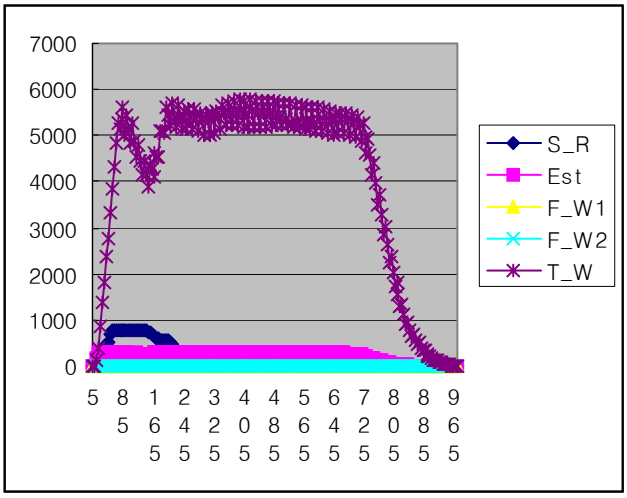
(실험4)-[그림 6]은 [그림 5]와 달리 ESTABLISHED의 값이 초기에는 증가하다가 중반까지 서서히 감소하고 있다. 이 시기에 TIME WAIT은 71초 까지 큰 폭으로 증가하다가 그 이후부터는 290초 까지 감소한다. SYN RECEIVED는 300초에서 하락 하지만 반대로 ESTABLISHED는 300초에서 증가한다. 이 시기에 TIME WAIT 또한 실험이 시작 되었을 때와 비슷한 추세로 증가하고 있다.



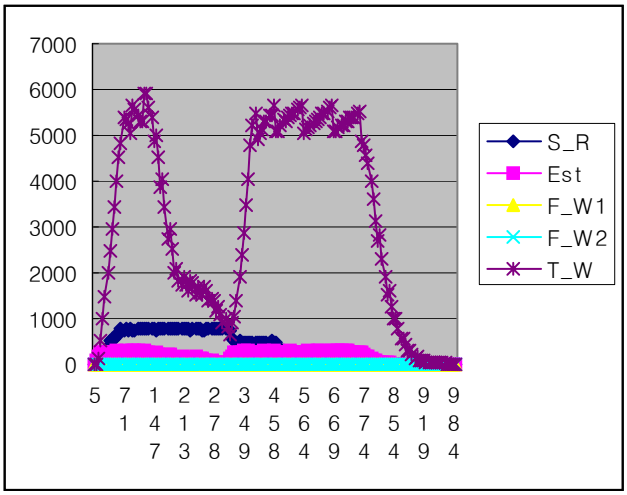
[그림 3] Request 프로세스 40개



[그림 4] Request와 Attack 프로세스 40개



[그림 5] Request 프로세스 60개



[그림 6] Request와 Attack 프로세스 60개

4. 결론

본 논문에서는 DoS 공격 중에서 서버에 치명적인 서비스 중단상태를 발생시키는 SYN Flooding 공격방식을 선택하여 서버의 TCP 상태전이에 미치는 영향에 대하여 실제 데이터를 근거로 그 결과를 도출하였다. TCP 상태전이 데이터는 기존에 단순히 서비스 중단 상태만을 검사하는 방식에서 좀 더 근본적인 대안을 마련할 수 있는 근거자료를 확보했다 할 수 있을 것이다. 향후 본 논문에서 제시한 실험방식을 통하여 다양한 공격방식을 시도하여 각 공격에 대한 패턴 분석과 함께 상호 상관관계를 추적한다면 SYN Flooding DoS 공격에 대한 근본적인 해법을 찾는데 중요한 자료가 될 것이다.

참고문헌

[1] 정현철, 현대용 “트래픽 분석을 통한 서비스거부공격 추적”, www.certcc.or.kr/paper/tr2003/030115-DoS.pdf

[2] 이철호, 김은영, 오형근, 이진석 “네트워크 트래픽 분석과 기계학습에 의한 DDoS 공격의 탐지”, 2004 한국정보처리학회 춘계학술발표대회, 제 11권 제 01호, 2004.5

[3] 박호상, 조은경, 강용혁, 엄영익 “데이터 마이닝을 이용한 서비스 거부 공격 탐지 기법”, 2003 한국정보과학회 추계학술대회, VOL.30 NO. 2-1, 2003 . 10

[4] JOINC http://www.joinc.co.kr/modules.php?name=News&file=article&sid=97#AEN275

[5] W.Richard stevens, "TCP/IP Illustrated, Volume 1", addison Westey, 1994

[6] W.Richard stevens, "UNIX network programming vol 1", Prentice Hall, 1999

[7] 김진희, 전철완, 오세민, 권경희 “TCP Time_wait 시간 조절을 통한 Intranet 환경에서의 웹서버 성능향상”, 2004 한국정보처리학회 추계학술대회, 제 11권 제 02호, 2004 . 11