제20회 한국정보처리학회 추계학술발표대회 논문집 제10권 제2호 (2003. 11)

SYN Flooding 공격 상황에서 적합한 패킷선정을 위한 엔진 모듈

정연광**,문종욱*, 최경희**,정기현*,임강빈***
* 아주대학교 전자공학부
** 아주대학교 정보 통신 대학원
*** 순천향대학교 정보보호학과
e-mail : jglory2k@madang.ajou.ac.kr

An engine module to select designate packets under the SYN flooding attack

YounKwang Jung**,JongWook Moon*,
Kyunghee Choi**,Gihyun Jung*, Kangbin Yim***
*Dept. of Electronics Engineering, Ajou University
**A Professional Graduate School
for Information and Communication Engineering, Ajou University
*** Dept. of Information security engineering, Soonchunhyang University

요 약

본 논문에서 제안하는 모듈은 기존에 존재하는 복잡하거나 혹은 리소스를 많이 차지하는 것이 아니라 Simple 한 방식을 사용하여 SYN Flooding 공격 중에도 적합한 패킷을 선별하는 모듈을 제안한다. 그리고 실험을 통해서 본 논문에서 제안하는 모듈의 성능을 알아본다.

1. 서론

인터넷이 대중화되는 가운데 그의 역작용인 여러 가지 Hacking 공격이 발생하게 되었다. 가장 대중적으로 이용되고 있는 공격중의 하나로 DDoS 공격을 들 수가 있다. 이 공격은 대부분의 유명한 모든 웹사이트가 공격을 받았다. 이러한 DDoS 공격의 90%가 TCP 프로토콜을 사용하고 있으며[2] 또한 TCP Protocol 중에 SYN flooding 공격이 가장 일반적으로 사용하고 있는 공격방식이다[1]. SYN Flooding 공격은 TCP 의 Three-way handshake 의 단점을 이용한 공격으로 Listen 상태에 있는 서버가 Client 의 요청으로 SYN 패킷을 받았을 경우에 서버에서는 일정량을 자원을 Client 를 위해서 할당을 하게 되는 것을 이용한 공격 방식이다. 공격용 SYN Packet 을 전송 시에 소스 IP 를 속이고 전송을 하게 되면 공격 당하는 서버에서는 three-way handshake 를 완료할 수가 없게 되고 공격 당하는 서버의 자원이 고갈되게 되어서 더 이상 서비스를 수행할 수가 없게 된다.

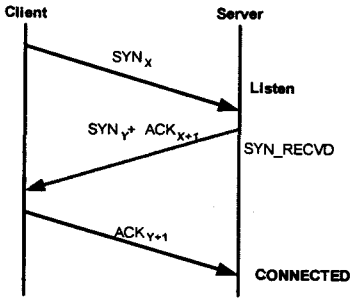
이러한 공격의 탐지에 대해서는 여러 가지 연구가 제시되었다. SYN-FIN(RST) Pair 방식[1]과 라우터에서 패킷의 불균형에 의한 탐지하는 방법[4] 등 외에 여러 가지 방법이 존재하지만 탐지를 하고 난 후에 대처방안에 대해서는 여러 문제점이 존재한다.

본 논문에서 제안하는 모듈의 방식은 공격용 패킷을 TCP/IP Protocol Stack 의 관여 없이 패킷을 처리하기 때문에 TCP/IP Protocol 에 존재하는 구조적인 문제를 이용한 SYN Flooding 공격에 대해서 강하게 대응할 수가 있다. 본 논문에서 제안한 모듈을 Proposed System 에서 구현을 하여 SYN Flooding 공격에 얼마나 효과적인지 실험을 통해 확인해 보았다.

2. 배경지식 및 관련연구

2.1 SYN Flooding 공격

SYN Flooding 공격은 TCP 프로토콜의 Three-way Handshake 의 특징을 이용한 공격이다. Three-way Handshake 는 클라이언트가 요구하는 데이터의 전송이 이루어 지기 전에 클라이언트와 서버간의 3 개의 패킷의 교환이 필요하다. 서버는 초기에 Listen 상태에서 초기 SYN 패킷을 받게 되면(SYN_RECV 상태) 서버는 SYN+ACK 패킷을 전송하게 된다. 그리고 마지막에 클라이언트가 ACK 패킷을 보내어 서버가 패킷을 받게 되면 Three-way Handshake<그림 1>가 끝나게 되는 것이다.



<그림 1>

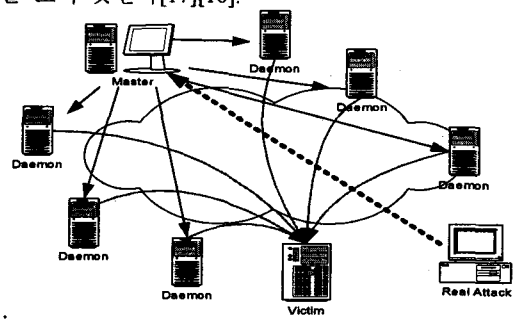
여기서 클라이언트가 ACK 패킷을 보내지 않게 되면 서버는 ACK 패킷을 기달 리는 상태가 된다[10]. 기본적으로 TCP 연결을 위해서는 BSD 계열의 네트워크 코드에서는 기본적으로 일정량의 자원을 할당하게 되어 있고[9] 동시에 half-open(SYN_RECV) 상태인 TCP 연결의 최대개수(Backlog Queue)는 다음과 <표 1> 같은 제한이 있다(Incoming Connection Request Queue).

OS	Backlog
Linux 2.2	130
RedHat 7.0	130
Solaris 2.5.1	32
WinNT 4.0	6
FreeBSD 2.1.5	128

< 표 1 >

위와 같은 Backlog Queue 의 크기[8]는 프로그램에서 설정에 따라 달라질 수가 있지만 결국 동시에 SYN_RECV 상태가 되는 개수의 제한이 있기 때문에 Backlog Queue 가 다 사용되고 나면 더 이상 TCP Connection 을 수행할 수가 없는 서비스 거부 상태(Denial Of Service)가 된다[5]. 현재는 공격의 효과를 높이기 위해서 보안의 헛점이 들어난 여러 개의 시스템의 권한을 획득하여 분산 공격<그림 2>을 시행하고 있다[11]. 실제 공격자(Real Attack)는 여러 개의 보안에 헛점이 존재하는 여러 개의 시스템의 권한을 획득하여 이 시스템에 공격용 프로그램을 설치하게 되고 이 시스템을 통하여 실제 공격하고자 하는 시스템(Victim)으로 수많은 패킷을 전송하게 된다. 이러한 분산 공격은 한 개의 시스템에서 공격하는 것보다 막는 것이 더 어렵다[12].

또한 공격패킷의 대부분은 소스 IP 를 랜덤 하게 생성을 하여 전송을 하게 된다. 소스 IP 를 변경하지 않으면 IP 추적을 통한 Blocking 기법이 존재하기 때문에 소스 IP 를 변경하지 않고 공격하는 경우에는 별 효과를 보지 못한다[17][18].



< 그림 2 >

2.2 관련 연구

이런 공격에 대응하기 위한 몇 가지의 연구가 진행되었다. 몇 가지 방식을 나열하면 다음과 같다.

- 라우터 방식
  - ingress & egress filtering [13][14][15]
    내부에서 나가는 패킷 혹은 외부에서 들어오는 패킷의 소스 IP 를 보고 타당하지 않은 IP 를 Blocking 하는 방법이다. 이 방식은 존재하는 모든 라우터의 설정이 필요하다는 점과 Mobile IP 일 경우에는 해당 IP 가 동적으로 움직이는 점을 간과한 방식이다.
  - History-Based IP Filtering[16]
    평상시 보이는 패킷의 IP 에 대한 database 를 구축하여 DDoS 공격이 탐지되면 그 동안 구축했던 database 를 통해서 패킷을 Blocking 하는 방식이다. 이 방식은 DDoS 공격 중에는 해당 database 에 없는 IP 를 가지고 있는 적합한 패킷은 접속할 수가 없다는 점과 database 를 관리해야 한다는 단점을 가지고 있다.
  - Rate-Limiting [6][7]
    패킷의 일정량 이상 통과하지 못하게 하는 방식으로 이 방식을 서버의 도달하는 패킷의 양을 줄이는 방식으로 공격용 패킷을 Drop 할뿐만 아니라 적합한 패킷도 Drop 하는 문제점을 가지고 있다.
- Gateway-Proxy
  내부 망과 외부 망을 직접적으로 연결하지 않는 방식으로 인증 기능으로 안전하다는 장점을 가지고 있지만 Traffic 의 병목현상으로 네트워크가 느려지는 단점이 있다. 이런 병목 현상을 없애기 위해서 또한 Load Balancing 을 이용한 방식[3]이 존재하지만 이 방식은 여러 개의 서버를 두어야 하는 비용적인 단점이 존재한다.

3. 구현

3.1 SYN Flooding Detection Module

SYN Flooding 공격을 탐지하는 모듈로 SYN-Fin(Pair)[1]방식을 사용하여 구현하였다. 공격을 탐지

하면 패킷 flow 를 <그림 3>의 (B)로 가게 되고 공격이 없으면 <그림 3>의 (A)로 가게 된다. 본 논문에서는 탐지에 관한 내용은 주제를 벗어나므로 더 이상 언급하지 않는다.

3.2 SYN Flooding Defend Module

1) Selective Table

SYN Packet 을 받았을 경우에 가장 먼저 검색되는 테이블로서 Establish IP 를 모은 곳이다.

2) Simple Packet Generate Module

패킷 생성 모듈이다. 여기서 생성하는 패킷은 SYN+ACK 패킷과 RST 패킷 두 개의 패킷만 생성하는 모듈이다.

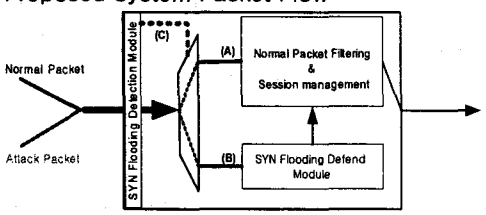
3) Establish IP Gathering

Proposed System 에서 보낸 패킷인지의 여부는 받은 ACK #를 가지고 판단하게 된다. ACK #는 일정 시간 동안 동일한 값을 유지하고 그에 대한 정보는 저장하고 있다. 이 ACK #를 자주 교환할수록 인증효과가 증가하지만 그만큼 성능의 저하가 발생하게 된다. 따라서 ACK #의 저장 개수는 시스템에 따라 다르게 된다.

3.3 Resource

할당해야 하는 자원은 Establish IP, ACK #만 있으면 되고 추가로 Establish IP 검색의 단축을 위한 Hashing & Linked list 가 필요하다.

3.4 Proposed System Packet Flow

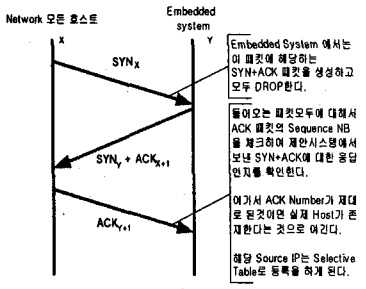


<그림 3>

전체적인 구성의 역할을 살펴보면 <그림 3>에 나타나 있다. SYN Flooding Detection Module 에서 공격이 탐지가 되면 모든 패킷은 (B)flow 를 향하게 되고 Selective Table 에 등록이 되지 않은 IP 의 SYN Packet 에 대해서는 SYN+ACK 패킷을 생성한다. 타당한 ACK 패킷이 오게 되면 Selective Table 에 등록을 하게 되어서 다음번 접속 시에는 (B)flow 를 통해 Normal Packet Processing(NPP)을 수행한다.

3.4 SYN Flooding Defend Module(SFDM)의 Flow

본 논문에서 제시하는 모듈은 다음과 같은 Flow 를 따른다.



<그림 4>

1) 들어오는 모든 SYN packet 에 대해서 SYN+ACK 패킷을 만들어 전송을 한다. 여기서 중요한 점은 SYN+ACK 패킷에 포함된 Sequence #을 기억해 두는 것이다. 그 외의 정보는 저장하지 않는다.

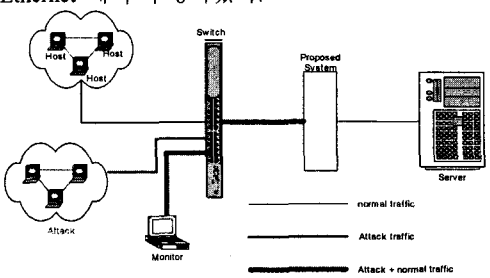
2) 여기서 Three-way Handshake 가 이루어 지는 IP 에 대해서 Selective Table 에 등록을 하고 이루어 지지 않는 것에 대해서는 지속적으로 Blocking 하게 된다.

3) Client 가 Establish 상태가 되었을 때 Proposed System 에서 RST packet 을 전송하여 클라이언트가 무한적으로 Establish State 가 되는 것을 방지한다.

4) 재 접속하는 클라이언트는 SYN Flooding Defend Module(SFDM) 에서 곧장 일반적인 패킷 처리 루틴으로 가게 된다.

4. 실험

<그림 5>의 실험환경을 살펴보면 Attack 시스템 군, Host, Switch, Proposed system, Server 로 구성이 되어 있다. Attack 시스템 군은 서버로 SYN Flooding 을 수행시키는 시스템이다. Host 는 실제 서비스를 요구하는 시스템이다. 여기에 웹 서버 앞에 본 논문에서 Proposed System 을 연결하였다. 이 테스트 상황은 100M Ethernet 에서 수행하였다.

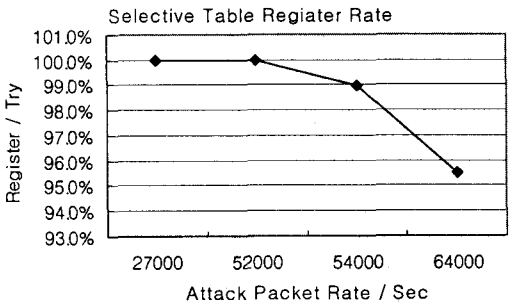


<그림 5>

본 논문에서 제시하는 모듈의 성능을 알아본다.

4.1 공격 중에 적합한 사용자 등록 실험

공격용 패킷의 증가에 따른 Host IP register Rate 를 보는 실험이다. 이 실험의 목적은 얼마나 많은 SYN Flooding 공격 중에 적합한 Host IP 를 선정하는 지를 볼 알아보는 실험으로 200 개의 서로 다른 IP 를 가지고 있는 시스템에서 접속을 시도하여 Proposed System 에 등록된 IP 개수와 비교한 수치를 나타내고 있다.

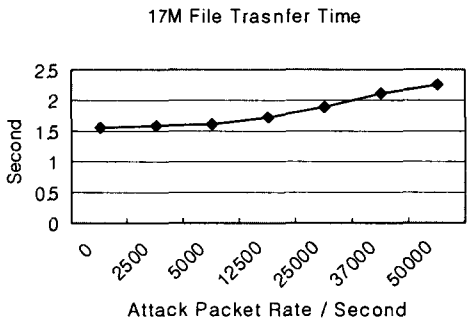


TCP 프로토콜의 특성상 재전송이 이루어 지므로 본 논문에서는 3 초 이내에 접속이 이루어 지지 않으면 접속 실패로 간주하였다.

실험의 결과는 대략 52000 여 개의 SYN Flooding 공격 중에도 거의 모든 host 를 등록하고 있는 수치를 보이고 있다. 이때 네트워크에서 보이는 총 패킷의 양은 SYN+ACK 패킷 때문에 두 배 수치인 102000 여 개의 패킷이 보이고 있다. 공격 패킷이 증가하여 등록율이 감소하게 되더라도 64000 여 개의 SYN Flooding 공격에도 95%가 넘는 등록 율을 보이고 있다.

4.2 시스템에 걸리는 부하량 실험

두 번째 실험으로는 SYN Flooding 공격 중 Proposed System 에 걸리는 부하량 실험이다. 이러한 실험을 하기 위해서 서버에 17M 정도의 파일을 저장해 놓고 host 에서 서버로 접속을 하여 해당 파일을 가지고 오는데 걸리는 시간을 측정하였다. 파일전송은 FTP 로 수행하였다. 본 실험의 목적은 SYN Flooding 공격 중에도 등록된 IP 에 해당하는 패킷요구의 처리수치를 나타내고 있다.



이 실험은 17M 크기의 파일을 여러 번 전송을 한 해당 시간의 평균값 이다. 이 실험 결과를 통해서 알 수 있는 것은 들어오는 SYN Packet 이 많을수록 file 전송 시간이 점차 길어짐을 알 수가 있다. 이것은 SYN Flooding 공격에 대한 처리의 양이 많아지게 되면 file 전송부분에 있어서 속도가 느려지지만 그리 큰 차이는 보이고 있지 않는다.

4.3 실험 결과

이 실험을 통해서 알 수 있는 것은 SYN Flooding 공격 중이라도 본 논문에서 제시하는 모듈을 이용하여 SYN Flooding 공격을 무산시키는 것과 모듈을 사용하는 경우에 성능의 저하는 크지 않음을 알 수가 있다.

5. 결론 및 향후 과제

본 논문에서 제시하는 모듈은 모든 SYN 패킷에 대해서 응답하여 Establish IP 만을 선택하는 방식으로 구현이 간단하여 Proposed System 에서 구현을 해보았다. 구현해본 결과 적은 량의 자원(3.3)을 가지고 만족할 만한 성능(4.1,4.2)을 보이고 있다. 향후 과제는 본 시스템에서 제안하는 모듈을 리눅스에 탑재 하여 어느 정도의 방어 효과 및 성능을 보이는지를 알아보는 연구가 필요하다.

참고문헌

[1] Haining Wang, Danlu Zhang, Kang G. Shin "Detecting SYN Flooding Attacks"

[2] D. Moore, G. Voelker and S. Savage, "Inferring Internet Denial of Service Activity", Proceedings of USENIX Security Symposium'2001, August 2001.

[3] Frank Kargl, "Protecting Web Servers from Distributed Denial of Service Attacks"

[4] Thomer M. Gil, Massimiliano Poletto "MULTOPS: a data-structure for bandwidth attack detection " (2001)

[5] Christoph L. Schuba, Ivan V. Krsul, Markus G. Kuhn, Eugene H. Spafford, Aurobindo Sundaram, Diego Zamboni, "Analysis of a Denial of Service Attack on TCP"

[6] David K. Y. Yau, John C. S. Lui, and Feng Liang. "Defending against distributed denial-of- service attacks with max-min fair server-centric router throttles". In Proceedings of IEEE International Workshop on Quality of Service (IWQoS), Miami Beach, FL, May 2002.

[7] Ratul Mahajan, Steven M. Bellovin, Sally Floyd, John Ioannidis, Vern Paxson, and Scott Shenker. "Controlling high bandwidth aggregates in the network".

[8] Project Paper :ECE 646 Defense Strategy against DoS Attack: Backlog Queue Alteration Soonyong Sohn

[9] R.W. Stevens and G. R.Wright. TCP/IP Illustrated, Volume 2, The Implementation. Prentice–Hall, Englewood Cliffs, New Jersey, 1995.

[10] S. Bellovin, "Security problems in the TCP/IP protocol suite," Comput. Commun. Rev., vol. 19, no. 2, pp. 32-48, Apr. 1989.

[11] S. Bellovin, "Distributed denial of service attacks," Feb. 2000, http://www.research.att.com/~smb/talks.

[12] Felix Lau, Stuart H. Rubin, Michael H. Smith, Ljiljana Trajkovic, "Distributed Denial of Service Attacks"

[13] Cisco Systems Inc. Defining Strategies to Protect Against TCP SYN Denial of Service Attacks, September 1996.

[14] Computer Emergency Response Team (CERT), Carnegie Mellon University, Pittsburgh, PA. TCP SYN Flooding and IP Spoofing Attacks, Sept. 1996. CA-96:21.

[15] P. Ferguson. "Network ingress filtering". Internet draft, Cisco Systems, Inc., September 1996.

[16] Tao Peng, Christopher Leckie, Kotagiri Ramamohanarao "Protection from Distributed Denial of Service Attack Using History-based IP Filtering"

[17] Alex C. Snoeren, Craig Partridge, Luis A. Sanchez, Christine E. Jones, Fabrice Tchakountio, Stephen T. Kent, and W. Timothy Strayer. "Hashbased ip traceback". In Proceedings of the 2001 ACM SIGCOMM Conference, San Diego, California, U.S.A., August 2001.

[18] Dawn X. Song and Adrian Perrig. "Advanced and authenticated marking schemes for ip traceback". In Proceedings of IEEE INFOCOM 2001, 2001. http://paris.cs.berkeley.edu/perrig/projects/iptraceback/tr-iptrace.ps.gz.