Proceedings of KFIS 2002 Spring Conference, 2002. 5. 25

TCP 네트워크에서 서비스거부공격의 탐지를 위한 RTSD 메커니즘

A RTSD Mechanism for Detection of DoS Attack on TCP Network

이세열* · 김용수**
*대전대학교 컴퓨터공학과 · **대전대학교 컴퓨터정보통신공학부

Se Yul Lee* · Yong Soo Kim**
*Department of Computer Engineering, Daejeon University
**Division of Computer and Communications Engineering, Daejeon University
E-mail : ailab@dju.ac.kr

요 약 문

최근 네트워크 취약점 검색 방법을 이용한 침입 공격이 늘어나는 추세이며 이런 공격에 대하여 적절하게 실시간 탐지 및 대응 처리하는 침입탐지시스템 구현은 어렵다. 본 논문에서는 시스템에 허락을 얻지 않는 서비스 거부 공격(Denial of Service Attack) 기술 중 TCP의 신뢰성 및 연결 지향적 전송서비스로 종단간의 통신서비스를 지원하는 3 way handshake를 이용한 SYN flooding 공격에 대하여 침입시도패킷 정보를 수집, 분석 및 침입시도여부를 결정하는 네트워크 기반의 실시간 침입시도탐지(Real Time Scan Detector) 메커니즘을 제안한다.

ABSTRACT

As more critical services are provided in the internet, the risk to these services from malicious users increases. Several networks have experienced problems like Denial of Service(DoS) attacks recently. We analyse a network-based denial of service attack, which is called SYN flooding, to TCP-based networks. It occurs by an attacker who sends TCP connection requests with spoofed source address to a target system. Each request causes the targeted system to send instantly data packets out of a limited pool of resources. Then the target system's resources are exhausted and incoming TCP port connections can not be established. The paper is concerned with a detailed analysis of TCP SYN flooding denial of service attack. In this paper, we propose a Real Time Scan Detector(RTSD) mechanism and evaluate it's performance.

Keywords : Real Time Scan Detector, Dos Attack

Ⅰ. 서 론

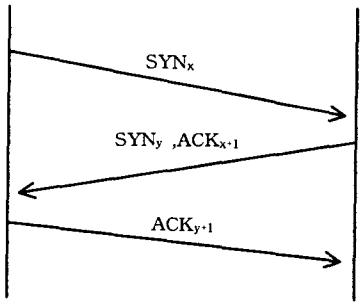
최근 네트워크 기술 발전으로 인하여 사회 전반에 걸쳐 인터넷 활용의존성이 매우 높아지고 있는 추세이다. 이러한 네트워크 기술 발전의 반대급부로 악의적 목적을 둔 침입을 위한 서비스 거부 공격이 심각한 문제로 대두되고 있다. 여기서 서비스 거부 공격이란 일반적으로 시스템의 자원을 고갈 또는 마비시켜 서비스지원을 하지 못하게 하는 일련의 침입을 위한 침입시도라고 볼 수 있다. 이들 중 가장 대표적인 서비스 거부공격으로서는 일명 SYN Flooding이라 불리는 공격형태가 있다. SYN flooding 서비스 거부 공격은 TCP 신뢰성 연결 지향적 전송서비스에 의하여 이루어지는데, 이러한 서비스 거부 공격은 인터넷환경에서 가장 많은 사용되어지는 TCP 기반의 프로토콜인 HTTP와 FTP 서비스를 지원하는 시스템에 크게 영향을 미치게 된다. 이 공격은 TCP 프로토콜의 구조적 약점을 이용하는데 이를 해결하기 위해서는 프로토콜의 수정이외에는 사실상 정확한 해답이 없다고 본다. 서비스 거부 공격은 크게

주요 파일을 훼손시켜 목적 시스템의 동작을 방해하는 우회적 서비스 거부 공격과 목적 시스템의 자원 및 네트워크 데이터전송을 위한 흐름제어 자원을 고갈시키는 공격으로 나눌 수 있다[1]. 현재 이를 해결하기 위한 여러 대안이 많이 연구되어 지고 있다. 본 논문에서는 제2장에서 서비스 거부 공격에 대해서 살펴보며 이를 해결하기 위한 방안을 알아본다. 제3장에서는 이러한 방안 중 TCP의 3 way handshake 연결과정에서 발생하는 half open 연결 상태를 실시간으로 탐지하는 실시간 침입 시도 탐지(Real Time Scan Detector) 메커니즘을 제안하며 마지막장에서 향후 연구방향과 결론을 제시한다.

II. SYN flooding attack

2.1 TCP SYN flooding attack

TCP SYN flooding 공격은 앞에서 거론되었듯이 TCP의 약점을 이용한 공격형태이다. 일반적으로 TCP는 신뢰성 지향적 연결이므로 서버와 클라이언트간에 연결 설정에는 그림 1과 같은 '3 way handshaking'라는 정상적 연결 흐름이 이루어진다.



[그림 1] 3 Way Handshake

만약, 여기서 클라이언트가 SYN_x를 요청하고 서버로부터 SYN_y와 ACK_{x+1}을 받은 후 ACK_{y+1}을 보내지 않으면 서버에서는 클라이언트로부터 응답이 올 것을 기대하고 반쯤 열린 'Half Open State'가 된다. 물론 얼마간 이런 상태가 유지된 후 다음 요청이 오지 않으면 해당 연결을 reset하게 된다. 이때 reset되기 전까지 메모리에는 backlog queue가 계속 쌓이게 되는데 이러한 reset이 되기 전에 지속적으로 이와 같은 요청이 아주 빠르게 이루어진다면 SYN packet은 backlog queue에 쌓이게 되어 결국 메모리 용량을 넘어서게 되면 해당 포드에 대한 연결을 받아들일 수 없는 상태인 서비스 거부 상태가 된다.

2.2 해결 방안

TCP SYN flooding attack에 대한 대안은 아래와 같은 것들이 있다.

(1) backlog queue

실제 서비스 거부가 발생하는 원인으로 backlog queue에 더 이상 받아들일 수 있는 조건이 되지 않기 때문이다. 이를 해결하기 위해서 backlog queue 크기를 증가시켜주는 방법이다. 그러나 H/W 및 OS마다 서로 다른 메모리 용량과 backlog queue 크기가 할당되어 있어 정확한 크기증가 선정이 어려워 진다. 예를 들어 Redhot Linux 6.x 시스템에서 메모리가 256MB인 경우 backlog queue 수치를 2048 이상으로 설정했을 때 TCP_SYNQ_HSIZE 와 tcp_max_syn_backlog의 수치를 조절하여야 한다. 그러나 이러한 대안은 지속적인 공격과 비용측면에서 볼 때 효율적이지 못하므로 적절한 대안이라 할 수 없다.

(2) syncookies

syncookies에는 크게 Berkeley, Linux, Reset cookie가 있으며 '3 way handshake'에서 TCP 헤더의 SYN's sequence number, 소스 및 목적 주소에 단방향 해쉬 함수를 적용한 암호화 알고리즘을 이용한 방식으로 연결 설정이 정상적으로 이루어지지 않으면 더 이상 소스 경로를 따라 가지 않고 정상적 연결 요청에 대해서만 연결 설정을 하여 자원의 낭비를 줄이는 방법이다. 이 방법은 backlog queue가 가득 쌓였을 경우에도 정상적인 접속 요구를 계속 받아들이므로 SYN flooding attack에 가장 효율적인 방법중 하나이다[2,3].

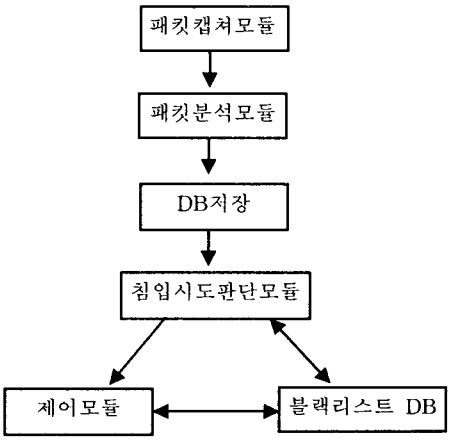
(3) packet monitoring

라우터 및 게이트웨이를 통과한 후 시스템 접근에 앞서서 모니터링을 하는 방법으로써 들어오는 패킷을 잡아 분석하여 'half open state'를 요청하는 포트 및 IP Address를 탐지하여 RST 등으로 연결 해제하는 방법이다. 본 논문에서는 제안하는 모니터링을 통한 탐지 또한 이 범주에 속한다[4].

이외에도 임의의 라우팅 테이블을 변경하여 트래픽이 전달되지 못하도록 ICMP redirects를 허용하지 않는 방법과 IP 소스 라우팅을 사용하여 목적지의 경로를 지정하여 믿을 수 있는 IP로 위장하지 못하도록 하는 소스 라우팅 패킷을 허용하지 않는 방법이 있다. 또한 방화벽을 설치하는 경우에는 시간지연, 트래픽 그리고 하드웨어 부하 등의 고려하여 설치하여야 한다.

III. RTSD Mechanism

본 논문에서 제안하는 RTSD(Real Time Scan Detector)는 128MB 메모리의 펜티엄III 리눅스 시스템과 SYN flooding attack 시스템 3대를 연결한 시험망에서 테스트 한 것이다. 전체적인 구조는 그림 2에서 보듯이 들어오는 패킷을 잡아 분석하고 DB 저장과 침입시도판단모듈을 통하여 'half open state'를 판단하는 시스템으로 되어있다.



[그림2] RTSD 모듈 구조

여기서 패킷 캡쳐 모듈에서는 promiscuous mode에서 데이터링크층의 패킷을 캡쳐한다. 패킷 분석 모듈에서는 패킷을 파싱하여 DB에 파싱하여 나온 로컬포트, IP Address, Sequence number, 윈도우 사이즈 및 공격시간 등으로 DB에 저장시키고 여기서 Sequence number, SYN 그리고 RST 등으로 1차 'half open state'를 탐지하게 된다. 여기서 공격으로 판단된 IP Address는 블랙리스트 DB에 저장되고 재차 공격시 블랙리스트 DB와 비교하여 공격을 탐지하는 방법이다.

그림 3은 DB에 저장된 스캔탐지로그 항목이며 'SYN' 과 'RST'의 수치가 각각 변경된 것과 그때 Sequence number 및 윈도우 사이즈도 바뀌었다는 것을 알 수 있다. 이러한 항목들의 패턴을 감시하면 실시간으로 'half open state'를 탐지 할 수 있는 것이다. 이러한 몇가지 항목으로 대응할 수 있는 방법은 아래와 같다.

첫째, 'half open state' 인 해당 포트 서비스를 중지하는 방법이 있을 것이다. 그러나 80포트 같은 경우 웹서비스를 지원해야하므로 서비스를 중지하는 방법은 그리 좋지 못하다고 생각된다. 다만, 일반적으로 사용하지 않는 포트에 대해서는 막아두는 것이 좋다. 여기서 일반적으로 Well-Known 포트(10여종)에 대해서는 서비스를 하고 나머지에 대해서는 중지한다고 보았을 때 높은 수준의 차단방법이지만 실질적 사용빈도수 측면에서 보았을 때는 아주 낮은 수준의 차단이라고 볼 수 있다.

둘째, 윈도우 크기를 변경하는 방법이다. 이 방법은 전송제어 또는 전송오류제어와 관련이 깊은 항목으로써 수신단(서버)의 버퍼 용량과 비례하는 항목이다. 즉, 윈도우 크기를 조절하여 연결 설정을 막는 방법이다[5].

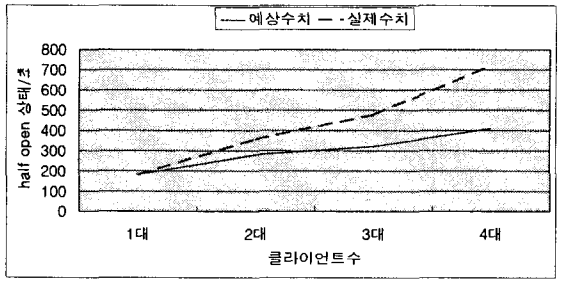
셋째, RTSD에서 블랙리스트DB에 저장된 IP Address에 대해서 Syn 재시도 횟수를 줄여주는 방법이다. 설령 위조된 IP라 할지라도 SYN Flooding attack횟수가 줄어들게 되는 것이다. 그러나 주의해야 할 사항으로는 블랙리스트 IP Address 이나 추후 SYN flooding attack의 징후가 나타나지 않으면 그에 대한 적절한 조절이 필요하다.

그림 4는 client에서 초당 SYN flooding attack인 경우 RTSD의 탐지율을 보여주고 있다. 테

Index	Scan Day	Scan Time	Port	IP	Seq num	Ack num	HLEN	RST	SYN	FIN	Window	Check Sum	half portScan
7	02-4-15	9:15:10	port 6	ip140.212	23094	0	5	0	1	0	2	36740	
8	02-4-15	9:15:10	echo(7)	ip140.212	23095	0	5	0	1	0	2	36482	
9	02-4-15	9:15:10	port 8	ip140.212	23096	0	5	0	1	0	2	36224	
10	02-4-15	9:15:10	discard(9)	ip140.212	23097	0	5	0	1	0	2	35966	
11	02-4-15	9:15:10	port 10	ip140.212	23098	0	5	0	1	0	2	35708	
12	02-4-15	9:15:10	systat(11)	ip140.212	23099	0	5	0	1	0	2	35450	
13	02-4-15	9:15:10	port 12	ip140.212	23100	0	5	0	1	0	2	35192	
14	02-4-15	9:15:10	daytime(13)	ip140.212	23101	0	5	0	1	0	2	34934	
15	02-4-15	9:15:10	port 14	ip140.212	23102	0	5	0	1	0	2	34676	
16	02-4-15	9:15:10	netstat(15)	ip140.212	23103	0	5	0	1	0	2	34418	
17	02-4-15	9:15:10	port 16	ip140.212	23104	0	5	0	1	0	2	34160	
18	02-4-15	9:15:10	gotd(17)	ip140.212	23105	0	5	0	1	0	2	33902	
19	02-4-15	9:15:10	msp(18)	ip140.212	23106	0	5	0	1	0	2	33644	
20	02-4-15	9:15:10	chargen(19)	ip140.212	23107	0	5	0	1	0	2	33386	
21	02-4-15	9:15:10	ftp-data(20)	ip140.212	23108	0	5	0	1	0	2	33128	
22	02-4-15	9:15:10	ftp(21)	ip140.212	23109	0	5	0	1	0	2	32870	
23	02-4-15	9:15:10	ftp(21)	ip140.212	16800331	0	5	1	0	0	0	32104	half portScaned
24	02-4-15	9:15:10	ssh(22)	ip140.212	23110	0	5	0	1	0	2	32612	
25	02-4-15	9:15:10	ssh(22)	ip140.212	16800332	0	5	1	0	0	0	31846	half portScaned
26	02-4-15	9:15:10	telnet(23)	ip140.212	23111	0	5	0	1	0	2	32354	
27	02-4-15	9:15:11	telnet(23)	ip140.212	16800333	0	5	1	0	0	0	31588	half portScaned
28	02-4-15	9:15:11	port 24	ip140.212	23112	0	5	0	1	0	2	32096	
29	02-4-15	9:15:11	smtp(25)	ip140.212	23113	0	5	0	1	0	2	31838	
30	02-4-15	9:15:11	port 26	ip140.212	23114	0	5	0	1	0	2	31580	

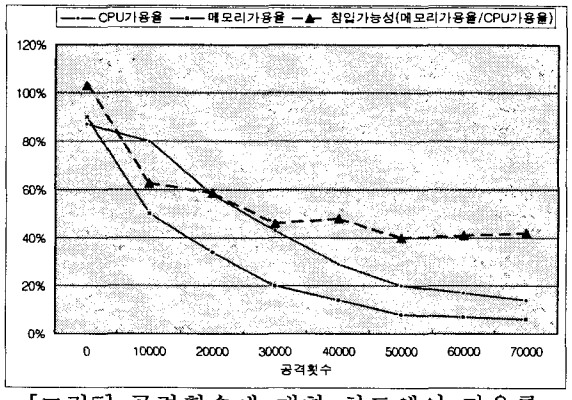
[그림3] DB에 저장된 스캔탐지로그항목

스트 결과 client가 1대인 경우 180 탐지/sec 의 half open state를 탐지하는 것으로 나타났으며 client 가 3대인 경우에는 320 탐지/sec 정도로 나타났다.
그림 4와 같이 탐지율은 클라이언트 수가 증가할수록 떨어지는 것을 볼 수 있다. 이는 실시간으로 패킷을 캡쳐하고 파싱하고 분석 그리고 저장 및 탐지결정에 따른 CPU의 부하와 메모리 용량에 의하여 패킷을 폐기 또는 캡쳐 도중에 발생한 손실로 보아야 할 것이다.



[그림4] 클라이언트 수에 대한 RTSD탐지율

이런 테스트 결과로부터 얻을 수 있는 것은 여러 가변 요소 중 어떤 요소에 의존성을 부여함으로써 가장 최적의 탐지를 할 수 있다. 그뿐만 아니라 탐지한 IP Address를 침입시도로 간주하고 블랙리스트 DB에 저장을 하여야 하는지를 결정하여야 한다. 그림 5에서처럼 시스템 부하 및 자원 고갈의 상태로 선정하는 것이 가장 타당하다고 보여지며, CPU가용율에 대한 메모리 가용율로 표현되는 수치가 50%이하로 떨어진다는 것은 메모리의 가용용량을 벗어난 것으로 간주하여 침입시도로 결정한다.



[그림5] 공격횟수에 대한 하드웨어 가용률

이때, 40%-60% 정도의 가용용량인 경우에는 메모리 가용율이 CPU 가용율 보다 낮을 경우에 더 많은 가중치를 두어 판단하는데 이는 메모리는 backlog queue 수치와 비례관계로 있으므로 메모리의 가용율이 낮다는 것은 backlog queue가 포화상태로 가고 있다는 뜻이 된다.

IV. 결 론

본 논문에서는 SYN flooding attack에 대해서 분석하였으며 해결책으로써 여러 대안 중에서 패킷을 분석하여 침입시도탐지기능을 수행하는 RTSD을 제안하고 시험망에서 테스트하였으며 탐지성능을 좌우하는 요소들간의 상호관계로부터 침입시도 여부를 판단하며 침입시도 여부를 명확히 판단 할 수 없는 가용용량 구역대(40%-60%)에서는 가변요소의 가중치에 의거하여 침입시도여부를 결정하였다. 시험망에서 테스트를 한 결과 하나의 시스템에서 실시간처리로 인하여 시스템에 부하로 개선된 성능을 나타내지는 못하였다. 향후 연구과제로 패킷 캡쳐와 분석 및 판단모듈을 각각의 시스템에 두어 하이브리드형태로 테스트를 하고 침입시도결정모듈을 학습 시켜 지능적으로 판단을 할 수 있는 형태로 연구해 나갈 예정이다.

V. 참고문헌

[1] Coputer Emergency Response Team, "TCP SYN Flooding and IP Spoofing Attacks," CERT Advisory: CA, 96-21, 1996.
[2] Syncookies mailing list. ftp://koobera.math.uic.edu/pub/docs/syncookies-archive, 1996.
[3] Aman Garg and A.L.Narasimha Reddy, "Policy Based End Server Resource Regulation," IEEE/ACM Transactions on Networking, Vol. 8, No. 2, pp. 146-157, 2000.
[4] C. L. Schuba, I. V. Krsul, M. G. Khun, E. H. Spaford, A. Sundram, and D. Zamboni, "Analysis of a denial of service attack on tcp," 1997 IEEE Symposium on Security and Privacy, 1997.
[5] Thomer M. Gil. "MULTOPS : a data structure for denial of service attack detection," Master thesis, Computer Science at the VRIJE Univ., December 2000.