DoS 공격 차단을 위한 효율적인 SYN-Proxy 설계

김주호[0] 이희조
고려대학교 컴퓨터과학기술대학원 디지털정보공학과
peuciel@korea.ac.kr heejo@korea.ac.kr

Design of Efficient SYN-Proxy against DoS Attack

Ju-Ho Kim[0] Hee Jo Lee
Dept. of Digital Information Engineering,
Computer Science Technical Graduate school, Korea University

요 약

최근 네트워크를 통한 사이버 공격의 형태가 날로 지능화, 조직화, 복잡화의 추세를 보이고 있어 이에 대한 효율적인 탐지와 대응 또한 날로 어려워지고 있다. 무엇보다도 TCP 의 3-way handshake 의 취약점을 이용한 서비스 거부 공격 또한 갈수록 정교화, 대형화되고 있어 네트워크의 가용성에 대한 가장 심각한 위협으로 대두되고 있다. 이러한 서비스 거부 공격 차단을 위해 ACL 을 이용하여 내부 네트워크 주소만을 허용시키는 Packet Filtering 방안이나 위조 IP 를 찾기 위한 Packet Marking, 3-way handshake 의 취약점을 보완하기 위해 Proxy 방식으로 이를 대신하는 SYN-Proxy 방식에 대한 논의가 활발히 이루어져왔다. 하지만 이러한 방식으로는 급격한 PPS 를 발생시키는 최근의 서비스 거부 공격을 효율적으로 차단할 수 없고 SYN-Proxy 방식 또한 모든 TCP Protocol 에 대한 Proxy 를 수행함에 따라 이에 따른 네트워크의 전송 지연 증가와 SYN-Proxy 을 수행하는 네트워크 장비의 성능문제로 인하여 이를 실제 네트워크에 적용하기란 쉽지 않다. 본 논문에서는 위와 같은 SYN-Proxy 수행 시 발생하기 쉬운 문제점을 보완하여 실제 네트워크에 이를 적용하여 효율적으로 서비스 거부 공격을 차단할 수 있는 개선된 SYN-Proxy 모델을 제안한다.

1. 서 론

인터넷 사용인구의 폭발적인 증가와 접속된 호스트들의 수가 급격히 증가함에 따라 불법적인 침입과 악의적인 의도의 시스템 훼손 시도가 늘고 있다. 이러한 네트워크기반의 공격 기법들은 점차 자동화, 지능화, 대중화, 분산화, 대규모화, 고속화, 은닉화, 범죄화 되어 가고 있으며, 공격 유형도 서버(호스트)중심에서 네트워크 기반의 공격으로 변화되어 가고 있다.

서비스 거부 공격은 시스템이나 네트워크 자원을 고갈시켜 해당 시스템과 네트워크가 정상적인 서비스를 제공할 수 없게 만드는 모든 공격기법을 지칭한다[1]. 통상적으로 수백 수천대의 호스트가 서버에 일정 시간동안 지속적으로 무의미하고 불필요한 트래픽을 보냄으로써 서버를 다운시키거나 네트워크 대역폭을 소멸시켜 서버가 서비스를 하지 못하게 한다. 현재 이러한 서비스 거부 공격은 대부분 TCP 프로토콜을 이용하여 서버를 공격하고 있다[2].

이러한 서비스 거부 공격을 막기 위해 특정 공격에 대한 서명으로 차단하거나 내부 네트워크 주소를 사전 정의하여 egress 트래픽을 filtering 하거나 packet marking 기법을 통해 차단하려는 많은 연구가 이루어져왔다[3]. SYN-Proxy 기법 또한 이러한 서비스 거부 공격을 네트워크 단에서 차단하기 위한 하나의 방법이며 L7 스위치와 같은 네트워크 장비가 취약한 TCP 3-way handshake 을 서버대신 클라이언트와 수행함으로써 정상적인 요청은 허용하고 서비스 거부 공격과 같이 비정상적인 요청에 대해서는 차단하게 된다.

본 논문에서는 각 트래픽을 그룹별로 분류하고 각 그룹별로 처리 정책을 별도 설정하여 SYN-Proxy 을 통한 서비스 거부 공격 차단에 있어 문제가 되는 전체 네트워크 성능의 저하와 전송 지연의 증가, 모든 3-way handshake 을 대신함으로써 발생하는 네트워크 장비의 부하 문제를 극복하기 위한 방법을 제안하였다.

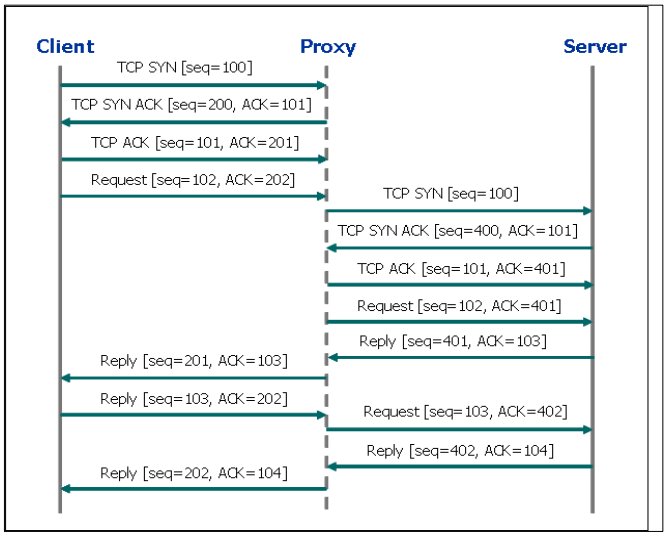
이 논문의 구성은 다음과 같다. 먼저 2장 관련연구에서는 SYN-Proxy 의 기법과 필요성에 대해 살펴보고, 3장에서는 본 논문에서 제안하는 개선된 SYN-Proxy 모델을 통한 서비스 거부 공격차단 방안을 설명하였다. 4장에서는 본 논문에서 제시한 모델에 대한 실험 결과를 제시한 후 이를 비교 분석하였고 마지막으로 5장에서는 본 논문의 결론을 상술하였다.

2. 관련 연구

SYN-Proxy 는 Delayed-Binding 이라고도 하며 이는 7계층 서버 로드 밸런싱을 하기 위해서 사용하는 기술이다. 4계층 서버 로드 밸런싱에서는 IP 주소와 포트번호를 기준으로 스위칭을 결정하기 때문에, TCP의 SYN 패킷만을 보면 어디로 연결할 지 판단할 수 있었다. 하지만 7계층은 데이터부분을 봐야 하기 때문에 접속이 완전히 이루어진 후 데이터 요청 패킷까지 검사해야 연결할 서버를 결정할 수 있기 때문에, 실제 서버와의 접속이 이루어지는 것은 클라이언트의 접속 요청이 있고 나서 어느 정도 시간이 지연된 후의 일이다.

즉 binding 이 지연되어서 이루어지기 때문에 delayed-binding 이라고 부른다.[4]

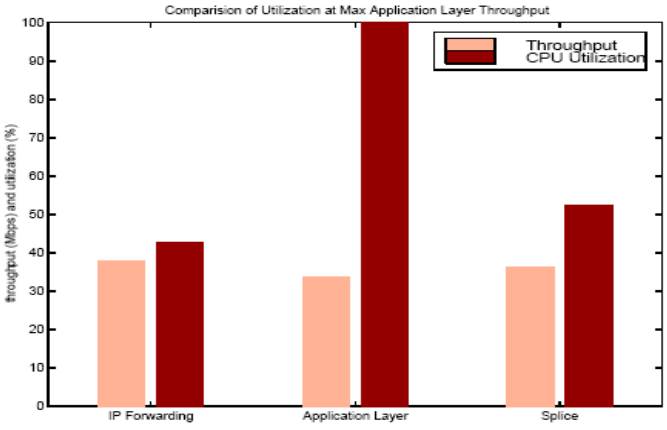
이런 SYN-Proxy 기법을 사용하면 클라이언트가 요청한 SYN 이 서버에 도착하기 전에 네트워크 프록시 장비가 이를 가로채어 SYN-ACK 를 클라이언트에 응답하고 정상적인 3-way handshake 가 완료되기 전에는 세션을 할당하지 않는다.



[그림 1] SYN-Proxy 개념

이것이 정상적인 요청일 경우만 서버로 binding 을 시도하고, 이것이 정상적인 요청이 아닐 경우엔 프록시 장비 자체에서 이를 차단하게 되므로 서버 단에는 어떠한 SYN Flooding 공격도 도달되지 않게 된다.[4]

하지만, 이러한 방식은 모든 트래픽에 대해 SYN-Proxy 역할을 수행해야 하므로 CPU 를 많이 소모하게 되는 문제를 수반하게 되고 이를 극복하고자 TCP Splicing 이란 기법에 대한 연구가 활발히 진행되었고 정형적인 Application level Proxy 에 비해 많은 성능 향상을 가져올 수 있다.[5]



[그림 2] 처리방식에 따른 CPU 사용율 비교

[그림 2]는 전형적인 Application Proxy 방식의 성능 문제를 극복하기 위한 연구의 결과로 TCP Splice 와 IP Forwarding 을 사용 시 CPU 사용율을 측정한 결과치이다. [6]

최근의 다양하고 광범위한 SYN Flooding 을 이용한 서비스 거부 공격에 대해 각각의 서명을 생성하여 차단한다는 것은 실제로 거의 불가능하므로 이러한 SYN-Proxy 을 수행하는 장비의 자체 보호 기능과 기능을 수행할 수 있는 충분한 처리 능력이 보장된다면 SYN Flooding 에 대해SYN-Proxy 은 좋은 방어 수단이 될 수 있다.

3. 개선된 SYN-Proxy 모델

3.1 개요

현재 사용되고 있는 SYN-Proxy 방식의 주된 한계는 모든 트래픽과 모든 TCP 프로토콜에 대해 SYN-Proxy 을 수행함으로써 발생되는 장비 자체의 성능과 전체 네트워크 전송 효율 저하의 문제이다. 이러한 문제를 극복하기 위해 본 논문에서는 3-way handshake 를 요청하는 클라이언트들을 그룹으로 분류하고 각 그룹별로 각기 상이한 정책을 적용하여 실제 SYN-Proxy 이 수행되는 클라이언트 범위를 축소시켜 장비의 성능 문제를 극복한다.

3.2 그룹 분류 알고리즘

최근의 SYN Flooding 공격과 같은 서비스 거부 공격은 주로 egress 트래픽이므로 우선적으로 egress 트래픽 중에서 내부 네트워크 IP 가 출발지가 아닌 패킷은 접근규칙 에 의해서 차단시켜 범위를 좁힌다. 따라서, 스푸핑된 IP들은 접근규칙에 의해 차단되는 것으로 가정하고 본 논문에서는 이에 대해서는 별도 고려하지 않는다.

본 논문에서는 3-way handshake 을 요청하는 클라이언트를 크게 세 개의 그룹으로 분류한다. Normal 은 정상적인 내부 클라이언트 IP 에 의한 정상적인 요청이 이루어진 것이며 Attacker 는 정상적인 3-way handshake 요청이 아닌 SYN Flooding 을 발생시키는 클라이언트그룹이고, 마지막으로 New_Client 는 어느 그룹에도 속하지 않는 즉, 최초로 요청된 IP 그룹이다. [그림3]은 각 그룹에 대한 정의이다.

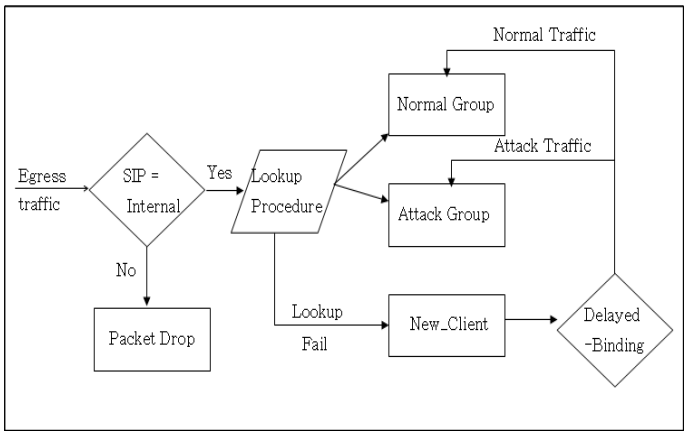
- *Normal*
  - *정상적인 내부 사용자 그룹*
- *Attacker*
  - *SYN-Flooding 발생시키는 사용자 그룹*
- *New_Client*
  - *최초로 요청된 사용자로서 Normal 혹은 Attacker 로 배정*

[그림 3] 각 그룹별 정의

클라이언트를 각 그룹으로 할당하게 위해서는 어느 정도의 학습 기간이 필요하게 되며 모든 클라이언트는 최초 연결 시에는 New_Client 로 분류되게 되고 최초 연결의 결과로 Attacker 나 Normal 그룹으로 배정되게 된다. 또한, 시간과 비례하여 New_Client 에 속하는 클라이언트는 점차 줄어드는 반면 Normal과 Attacker 그룹의 클라이언트는 증가하게 된다.

이 경우 보안상, 운영상의 문제가 발생하게 되므로 이를 해결하기 위하여 각 클라이언트 당 에이징 시간을 두어 일정 시간이 경과된 이후에는 테이블을 갱신하고 다시 New_Client 로 분류되게끔 한다.

[그림 4]는 위 절차를 도식화한 것이다.



[그림 4] 그룹 분류 알고리즘

3.3 그룹별 정책

[그림 5]은 각 그룹별로 트래픽 처리 정책을 나타낸 것이다. Normal 그룹에 속한 IP가 SYN을 요청 시는 이 IP 는 신뢰된 것으로 간주하고 이에 대해 SYN-Proxy 없이 그대로 바이패스 한다. 그리고 Attacker 그룹에 속한 IP가 SYN을 요청 시에는 이는 Malicious 한 트래픽으로 간주하여 차단시킨다. 마지막으로 Normal과 Attacker 그룹에 속하지 않은 IP 에 의해 요청되었을 경우는 이를 New_client 그룹으로 분류하고 이에 대해 SYN-Proxy 을 수행하여 처리하고 이에 대한 결과에 따라 정상일 경우엔 Normal 그룹으로 공격일 경우엔 Attacker 그룹으로 추가한다.

1) IF *Client_Group is Normal*
Then *Bypass ;*
2) IF *Client_Group is Attacker*
Then *Drop ;*
3) IF *Client_Group is New_Client*
Then *Delayed-Binding and Update Client_Group ;*

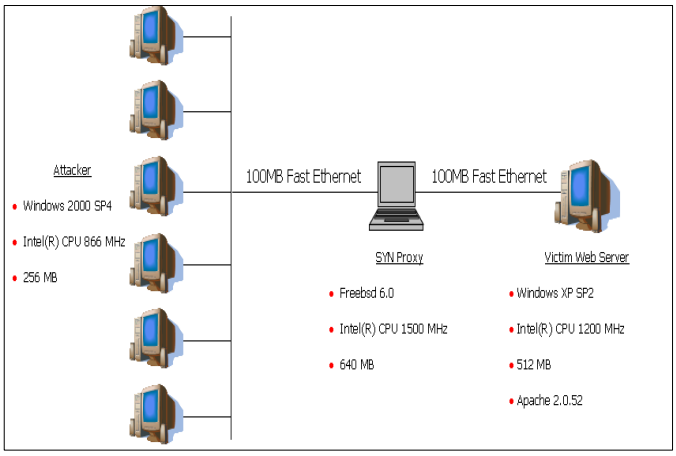
[그림 5] 그룹별 트래픽 처리 정책

3.4 오탐 처리 정책

제안된 SYN-Proxy 모델에서의 가장 큰 보안상의 문제점은 Normal 그룹에 대해서는 어떠한 검증도 이루어지지 않는다는 것과 시간이 지날수록 이 Normal 과 Attacker 그룹에 속하는 클라이언트의 수가 증가한다는 것이다.

이를 보완하기 위해서 각 그룹별로 에이징 시간을 두어 각 그룹 테이블을 갱신하여야 하며 Normal 그룹에 속한 각 클라이언트에 대해서는 초당 발생시키는 SYN 에 대한 임계치를 설정하고 이를 검사하여 이 임계치를 초과하는 클라이언트에 대해서는 Attacker 그룹으로 재 분류하는 처리 과정을 추가한다.

4. 실험 및 결과



[그림 6] 테스트 네트워크 토폴로지

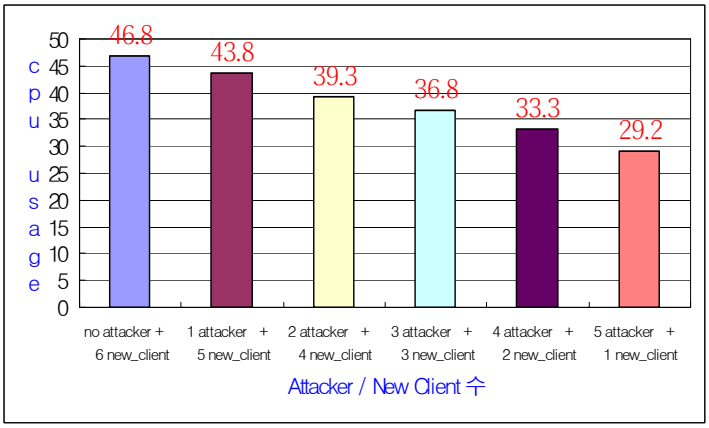
SYN-Proxy 수행에 따른 CPU 의 사용정도를 측정하기 위하여 [그림-6]와 같이 테스트 네트워크를 구성하였다. SYN Proxy 장비는 Freebsd 6.0 의 pf(packet filter) 를 사용하였고 현 release version 상의 pf 제약 상 SYN-Proxy 장비는 transparent mode 가 아닌 route mode 로 구축하였다.

각 클라이언트 당 차례대로 초당 3000개의 SYN Attack 을 발생시켰고 실 환경에서의 CPU 사용율을 측정하기 위하여 background 트래픽이 있을 경우와 없을 경우를 각각 측정해 보았다.

1) Attacker 그룹 증가에 따른 CPU사용 결과

개선된 SYN-Proxy 모델을 6개의 클라이언트에 대하여 실험을 하였으며 Attacker 그룹을 증가시키면서 CPU 사용 추이를 측정해 보았다.

Attack 그룹에 등록된 클라이언트가 증가할수록 SYN-Proxy 장비에서는 해당 IP를 Blocking 하기 때문에 모든 클라이언트에 대해 SYN-Proxy를 적용하는 전통적인 방식에 비해 CPU 사용율이 감소하는 것을 볼 수 있다.

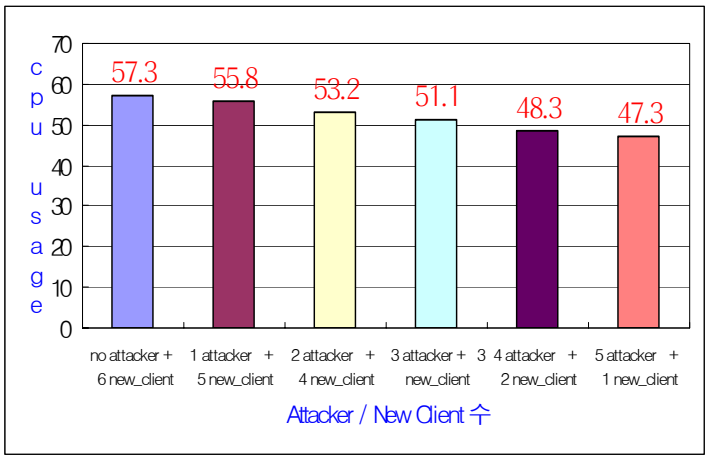


[그림 7] attacker 그룹의 증가에 따른 CPU 사용율

2) background 트래픽 추가에 따른 결과 비교

앞에서의 실험 결과 SYN-Proxy 의 수행 횟수와 CPU 사용율은

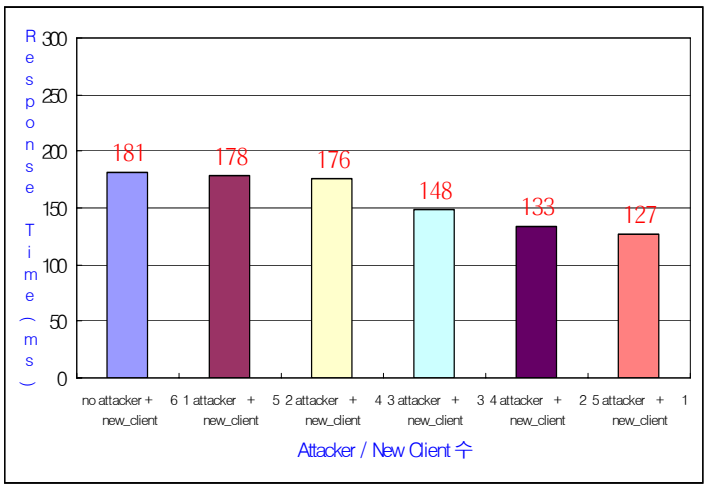
비례함을 알 수 있다. [그림 8]은 실제 환경에서의 영향을 측정하기 위하여 Attacker 와 Victim Web Server 간에 100Mbps dummy 트래픽을 발생시킨 후 1)번과 동일하게 측정한 결과이다.



[그림 8]100Mbps background traffic 하에서의 attacker 그룹의 증가에 따른 CPU 사용율

3) 응답 시간 측정

제안된 SYN-Proxy 모델에서 Attacker 그룹이 증가함에 따라 웹에 대한 응답 시간을 측정해보았다. 아파치 기본 설치 홈페이지로 실험을 하였으며 Attacker 그룹이 증가할수록 응답 시간 또한 개선되는 것을 [그림 9]에서 볼 수 있다.



[그림 9] 응답 시간 측정

5. 결론 및 향후 연구 과제

본 논문에서는 개선된 SYN-Proxy 모델을 적용하여 Attacker 가 증가할수록 또 트래픽이 증가할수록 SYN-Proxy 역할을 수행하는 장비의 CPU는 증가할 수 밖에 없는 기존의 SYN-Proxy 모델이 구조적으로 갖고 있는 성능 문제점을 극복하고자 하였다.

그 결과 각 그룹 분류 알고리즘에 의하여 Attacker 로 기 분류된 그룹에 대해서는 Blocking 을 수행하고 Client 그룹에 대해서는 Bypass 하며 New_Client 그룹에만 SYN-Proxy 를 수행함으로서 기존 SYN-Proxy 방식에 비하여 CPU 사용율 및 응답시간을 감소시킬 수 있었다.

향후 과제로는 본 논문에서 제안한 알고리즘에 대해 각 그룹에 속한 각각의 IP 를 가장 효율적으로 빨리 탐색하는 방안과 오탐 감소 및 보안성 강화를 위해 가장 적합하고 효율적인 에이징 시간을 찾아내는 방향으로 연구가 이루어져야 할 것이다.

참고문헌

[1] Frank Kargl, Joern Maier, and Michael Webber, "Protecting Web Servers from Distributed Denial of Service Attacks" Proceedings of the tenth international conference on World Wide Web Apr. 2001.

[2]Rocky K.C. Chang, Defending against "Flooding-Based Distributed Denial of Service Attacks: A Tutorial" IEEE Communications Magazine Oct. 2002.

[3] Kihong Park and Heejo Lee "On the Effectiveness of Probabilistic Packet Marking for IP Traceback under Denial of Service Attack" IEEE INFOCOM. 2001.

[4] Preventing Denial of Service, Nortel Networks http://www.eads-telecom.com/images/administrables/42/alteon_switch_2224.pdf

[5] Marcel-Catalin Rosu and Daniela Rosu, "An evaluation of TCP splice benefits in web proxy servers" Proceedings of the 11th international conference on World Wide Web May. 2002.

[6] David A. Maltz and Pravin Bhagwat, "TCP Splicing for Application Layer Proxy Performance" IBM Research Report RC 21139, March, 1998.