DoS 공격 예방을 위한 확장 TCP 설계

박진원° 김명균
울산대학교 컴퓨터.정보통신공학과
zwsonic@cnlab.ulsan.ac.kr°, mkkim@mail.ulsan.ac.kr

Design of Extended TCP preventing for DoS attack

Zin-Won Park° Myung-Kyun Kim
school of Computer Science and Information Technology, University of Ulsan

요 약

보안의 중요성이 강조되고 있는 요즈음 해킹이라는 용어는 시스템에 침입하여 정보를 빼내거나 수정, 삭제하는 행위를 포함하여 서비스를 방해하는 행위로도 일컬어지고 있다. 보편적으로 많이 사용되고 있는 TCP 프로토콜 자체의 취약점을 이용한 서비스 거부공격이 갈수록 거대해지고 위험한 공격 방식으로 인식되고 있지만 이에 대한 적절한 예방법이 없는 것이 사실이다. 본 논문에서는 TCP 프로토콜을 확장하여 서비스 거부공격에 대한 예방 기능을 가진 프로토콜을 제안한다.

1. 서 론

HTTP, FTP, SMTP, Telnet 등 많은 서비스 프로토콜들은 TCP를 기반으로 동작한다. 그리고 거의 모든 운영체제들은 TCP를 제공하고 있다. 뿐만 아니라 많은 서버 프로그램과 클라이언트 프로그램들이 TCP를 기반으로 개발되어, 인터넷 상의 대부분의 통신들이 TCP를 이용하여 이루어지고 있다고 해도 과언이 아니다. 하지만 TCP는 자체적인 취약점을 지니고 있어 이를 이용한 서비스 거부(Denial of Service) 공격에 무방비 상태로 있다. 한 예로, 지난 2000년 야후(yahoo.com)에 대한 서비스 거부 공격을 들 수 있을 것이다.

이러한 서비스 거부 공격에 대처하기 위한 많은 연구들이 이루어 지고 있다. [1]에서는 Kihong Park과 Heejo Lee은 PPM (Probab-ilistic Packet Marking)기법을 이용하여 패킷의 이동 경로를 역추적(traceback)함으로써 패킷의 송신지를 확인하는 방법의 효과를 논하였다. Kanta Matsuura과 Hideki Imai는 [2]에서 Diffie-Hellman 키 분배 프로토콜(key-agreement protocol)을 이용하여 연결을 맺으려는 두 노드 사이의 인증 방법을 제안하였고 [3]에서는 Stamatis Karnouskos가 네트웍에 존재하는 에이전트들을 이용하여 DoS 공격을 감지하는 방법을 제안하였다.

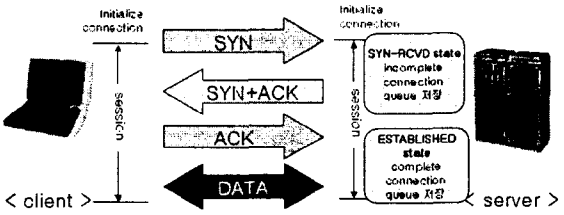
본 논문에서는 DoS 공격을 예방하기 위하여 서버와 클라이언트 간의 TCP 연결 설정 과정에 키 값을 주고 받도록 함으로써 DoS 공격을 효과적으로 막을 수 있는 방법을 제안한다. 또한 제안된 확장 TCP에 대한 설계와 Xinu 상에서의 구현에 대해 기술한다.

논문의 순서로는 2장에서 TCP의 취약점을 분석해 보고 3장에서는 본 논문의 주제인 확장된 TCP를 설명한다. 4장에서는 본 프로토콜의 효과를 평가한 후 5장에서 결론을 맺는다.

2. TCP의 취약점

2.1 3-way handshake

TCP는 두 호스트 사이의 연결을 설정하기 이전에 3-way handshake 과정을 거치게 된다. 3-way handshake는 두 호스트의 TCP process의 송/수신 버퍼에 Sequence Number 값을 동기화 시키는 작업을 하게 되는데 그 과정은 <그림 1>에서 볼 수 있다.



< 그림 1. TCP 3-way handshake>

접속을 요청하는 클라이언트는 서버에게 SYN 패킷을 전달한다. 이것을 받은 서버는 클라이언트에게 자신의 SYN 패킷과 클라이언트로부터 SYN 패킷을 잘 받았다는 표시로 ACK를 보낸다. 클라이언트는 ACK를 받고 자신의 SYN이 잘 갔다는 것을 알 수 있으며 서버가 보내온 SYN에 대한 응답으로 ACK를 보낸다.

이 과정이 TCP 3-way handshake이며 3-way handshake가 끝나면 두 노드간의 연결은 ESTABLISHED 상태가 되어 데이터의 전송이 이루어질 수 있게 된다.

2.2 취약점

TCP의 취약점은 앞에서 살펴본 3-way handshake에 있다. 이것은 악의의 공격자가 인터넷 상에 존재하지 않거나 현재 작동하지 않고 있는 호스트의 IP address를 송신자 주소로 조작(IP Spoofing)하여 접속 요청을 보내는 것이다.

공격자는 자신의 호스트 혹은 다른 호스트를 이용하여 공격 대상 호스트에 접속 요청(SYN)을 보낸다. 이 때 보내는 접속 요청은 조작된 IP address를 가지고 있다. 접속 요청을 받은 호스트는 그에 대한 응답으로 ACK와 SYN을 보내게 된다. 하지만 이 패킷들은 수신 호스트가 응답할 수 없는 상태이기 때문에 ACK가 전송되지 않는다.

공격 대상 호스트는 SYN+ACK를 보낸 상태에 있기 때문에 상대 호스트로부터 ACK가 오기만을 기다리게 된다. 여기에는 대기 큐가 사용된다. 큐는 한정된 크기(incomplete connection queue+complete connection queue)를 가지고 있기 때문에 요청들이 큐에 가득차게 되면 더 이상의 요청은 받아 들이지 못하게 된다. 따라서 호스트는 서비스 불가 상태가 되고 수 만개에 이르는 공격 패킷을 받을 경우 다운되어 버리기도 한다. 이러한 공격을 IP spoofing에 의한 DoS(Denial of Service) 공격이라고 한다. 비록 이 큐는 타이머를 가지고 있어 일정 시간동안 응답이 없을 경우 큐에서 삭제를 하지만 공격 패킷은 큐의 크기보다 매우 큰 양이기 때문에 공격은 성공할 수 있게 된다[4].

만약 공격자가 조작한 IP address를 가진 호스트가 작동중에 있다면 그 호스트는 자신이 SYN을 보내지 않는 호스트로부터 온 SYN+ACK를 받았기 때문에 잘 못 온 것으로 간주하고 RESET 시켜버려 공격은 실패한다.

3. 확장 TCP 설계

3.1 개요

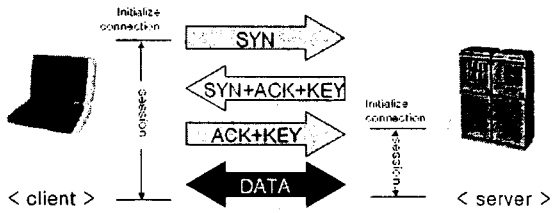
DoS 공격이 성공하게 되는 주된 핵심은 서버가 3-way handshake 과정에서 SYN+ACK를 보낸 후 상대의 응답을 기다리고 있고 그 응답은 도달하지 않는 다는 것이다. 이러한 공격 방식의 해법으로는 다음의 두 조건을 만족해야 한다.

1. 서버는 클라이언트의 요청을 받은 후 응답을 보내고 그것에 대한 상태를 유지하지 않는다.
2. ACK를 보낸 클라이언트가 SYN을 보낸 클라이언트임을 확신할 수 있어야 한다.

TCP의 3-way handshake는 두 호스트의 데이터 버퍼를 초기화하기 위해 필요한 과정이므로 이 과정을 생략할 수는 없다. 따라서 3-way handshake 과정에 새로운 기능을 추가하여야 한다.

3.2 알고리즘

<그림 2>는 본 논문이 제시하는 DoS 공격 예방을 위한 확장된 TCP의 3-way handshake를 위해 KEY를 추가하였다.



< 그림 2. 확장된 TCP 3-way handshake>

접속을 원하는 클라이언트는 서버에게 접속을 하기 위한 SYN 패킷을 전송한다. 요청을 받은 서버는 클라이언트의 IP address와 서버만이 가진 고유의 값을 조합한 Hash 값으로 KEY를 생성하여 클라이언트에게 서버의 SYN과 클라이언트가 보내온 SYN의 ACK에 KEY를 실어 전송 한다. 이 값을 받은 클라이언트는 ACK에 KEY를 실어 서버에게 전송한다. 서버는 클라이언트로부터 받은 KEY가 자신이 보낸 KEY가 맞는지 검사를 한다. 맞으면 ACK를 받아들여 기존의 클라이언트와의 연결에 필요한 큐를 할당하고 연결을 설정한다.

서버는 클라이언트가 보내온 KEY값을 자신이 보낸 KEY값이 맞는지 검사하는 과정에서 서버가 보낸 KEY 값을 유지하고 있다가 그 값과 비교하는 것은 바람직하지 못하다. 그 이유는 이 때에도 값을 유지하는 메모리 공간이 필요하게 되고 이 공간을 오버플로우(overflow) 시키는 공격이 가능하기 때문이다. 따라서 서버는 값을 유지하는 것이 아니라 클라이언트가 보내온 KEY값을 받은 후 KEY를 다시 만들어내어 비교 과정을 수행한다.

또한 서버가 보내주는 KEY는 Hash된 값으로 공격자가 이 값을 임의로 만들어내는 것을 방지하기 위하여 서버만이 알고 있는 값과 클라이언트의 IP address와의 조합을 이용하여 만들어져야 한다.

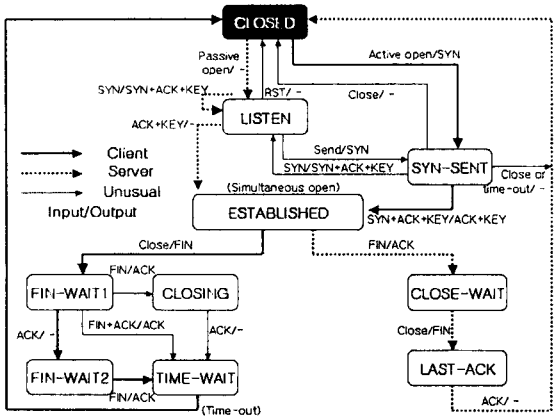
[2]에서 사용한 Diffie-Hellman 키 분배방식은 큰 수의 지수 연산을 해야 하고 동시 접속자가 많은 서버의 경우 연산에 많은 시간을 필요로 하기 때문에 처리 속도가 빠른 Hash 함수를 이용하였다. 또한 공격자가 KEY를 수집하거나 재사용하는 것을 방지하기 위하여 서버가 초기화 될 때 서버의 비밀값은 무작위로 선택되는 방식을 취하였다.

만약 클라이언트가 보내온 KEY값이 잘 못된 값이라면 서버는 이 요청을 무시한다.

3.3 확장TCP State Machine

<그림 3>은 기존의 TCP finite state machine에 확장 프로토콜을 적용한 것이다. 여기에는 SYN-RCVD 상태가 삭제 되었고 상태 전이에 따른 Input/ Output 값이 일부 수정되었다.

클라이언트 과정(굵은 실선)을 살펴보면, 클라이언트는 소켓을 오픈하고 SYN을 전송한 후 SYN-SENT 상태가 된다. 서버로 부터의 ACK가 도착하기를 기다리고 있다가 서버가 SYN+ACK+KEY를 보내오면 클라이언트는 ACK+KEY를 보내고 ESTABLISHED 상태가 된다. 이 이후는 기존의 TCP state machine과 같은 과정을 거친다.



< 그림 3. 확장된 프로토콜의 TCP state machine >

서버의 과정(굵은 점선)을 살펴보면, 서버는 소켓을 오픈하고 LISTEN 상태가 된다. 클라이언트로부터 SYN을 받으면 클라이언트의 IP address와 자신의 비밀값을 조합하여 KEY를 생성하여 ACK+SYN과 함께 클라이언트에게 전송한 후 상태를 유지한다. 클라이언트로부터 KEY+ACK을 받으면 해당 클라이언트의 IP address로 KEY를 생성한 후 받은 KEY와 비교하여 같으면 ACK를 보낸 후 ESTABLISHED 상태가 되고 그렇지 않으면 상태를 유지한다.

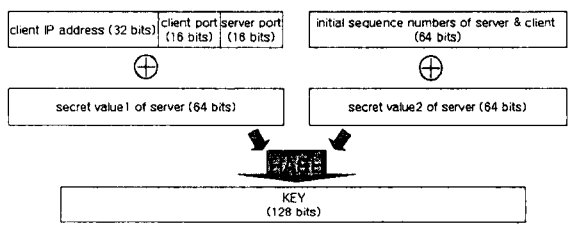
4. 확장 TCP 구현

4.1 구현환경 및 정의

본 프로토콜을 구현하는데 사용된 운영체제는 Xinu[5]이고 구현 언어는 C언어이다. Xinu는 운영체제 및 프로토콜 학습을 위해 만들어진 BSD 기반의 운영체제로 소스코드가 공개되어 있고 간략하게 구현되어 있으므로 수정 및 확장을 적용하여 시험하기에 용이하게 되어 있다.

KEY를 전달하는 flag로는 TCP의 옵션 필드를 이용하였다. 옵션 필드는 현재 0부터 3까지가 정의 되어 있으므로 KEY전달 옵션은 4를 부여하였다.

KEY 생성을 위한 서버의 비밀값은 128bit로 부팅시 무작위 선택 되며, KEY는 클라이언트의 IP 주소, 클라이언트의 포트번호 그리고 서버의 포트 번호와 초기 시퀀스 번호를 XOR연산을 하여 MD5로 해쉬 함수를 적용한 128bit 해쉬값이다.



< 그림 4. 서버의 키 생성 과정 >

4.2 평가

본 프로토콜을 이용하여 서버 프로그램과 클라이언트 프로그램을 제작하여 테스트를 진행하였고 서버의 큐의 수는 7개로 설정하였다.

클라이언트 접속방식 및 동시 접속수		기존 TCP	확장 TCP
IP spoofing 사용	10	정상	정상
	50	다운	정상
	100	다운	정상
IP spoofing 사용 안 함	10	정상	정상
	50	정상	정상
	100	정상	정상

< 표 1. 기존 TCP와 확장 TCP의 효과 비교 >

5. 결론 및 향후 계획

<표 1>에서 볼 수 있듯이 기존의 TCP방식에서는 IP spoofing을 이용한 DoS공격에서는 쉽게 서버스 거부 상태로 되었지만 확장된 TCP는 많은 공격을 받더라도 정상적인 서비스를 할 수 있었다. 또한 IP spoofing을 하지 않는 정상적인 접속에 대해서도 정상 작동하였다.

본 논문에서 제시한 확장 TCP 프로토콜은 IP spoofing을 이용한 서비스 거부 공격을 원천적으로 예방할 수 있었다. 또한 다른 침입탐지 프로그램과 함께 사용한다면 IP spoofing을 사용하지 않는 DDoS공격에 대해서도 안전한 서비스를 할 수 있을 것이다.

향후 계획으로는 확장된 TCP가 Hash 함수를 이용함으로 해서 서비스 제공에 소요하는 시간을 최소화하도록 할 것이며 리눅스 커널을 수정하여 다양한 서버 프로그램에 대해 테스트를 진행할 것이다. 또한 수정된 리눅스 커널을 공개하여 많은 사용자로 하여금 성능 평가를 할 수 있게 진행할 계획이다.

6. 참고 문헌

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

[2] Kanta Matsuura and Hideki Imai " *Resolution of ISAKMP/Oakley Key-Agreement Protocol Resistant against Denial-of-Service Attack*" IEEE Internet Workshop, 1999

[3] Stamatis Karnouskos " *Dealing with Denial-of-service Attacksin Agent-enabled Active and Programmable Infrastructures*" IEEE COMPSAC 2001

[4] Andrian Piskozub " *Denial of service and distributed denial of service attacks*" IEEE International Conference 2002

[5] XINU *http://public.ise.canberra.edu.au/~chrisc/xinu.html* online documents