TCP 순서번호 암호화 방식 제안

* 서대희 *강희조
* 동신대학교 전기전자공학과

Suggest of TCP sequence number Encryption

* Dea Hee Seo, *Heau Jo Kang
*Dept. of Electircal & Electronic Eng, Dongshin Univ

요 약

본 논문에서는 TCP(Transmission Control Protocol) 프로토콜의 약점을 이용해 최근 대두되고 있는 IP Spoofing공격, SYN Flooding공격, DoS(Denial of Service)공격에 대응하기 위해 TCP 프로토콜에서 순서번호 생성의 단순함을 극복하고자 하였다. 이러한 방안으로 난수, 순서번호, 접속 요구 IP의 공개키를 이용하여 일회용 키를 생산·분배 하였다. 여기서 발생된 일회용 패스워드를 접속 요구 IP에 할당함으로써 순서번호에 암호화 및 IP 인증을 통한 시스템의 보안 수준을 높이고자 하였다.

Ⅰ. 서 론

최근 PC 통신의 대중적 성공으로 인하여 대부분의 컴퓨터 시스템이 네트워크에 접속되고 유닉스 시스템에 대한 관심이 높아져감에 따라 프로토콜의 약점을 이용한 공격 형태들이 증가하고 있는 실정이다. 프로토콜과 관련된 보안문제를 기술함에 있어서 프로토콜의 의미를 파악하는 것은 중요한 출발점이 될 수 있다. 특히 IP의 인증 체계는 아주 단순하며 출발지와 목적지 IP주소 두 가지 요소를 가지고 데이터그램을 구분한다. 따라서 IP위에 올라가는 TCP 및 UDP의 경우 부가적인 인증 체계를 가지지 않는 한 위험 부담을 가지게 된다.

즉, 침입자가 IP 주소를 도용하여 임의로 데이터그램을 만들어 목적지 호스트(target host)에 전송시키면 목적지 호스트에서는 이를 알아 낼 방법이 없다. 즉, 도용한 IP 데이터그램의 출발지 IP 주소를 목적지 호스트가 신뢰하고 있는 IP 주소로 도용하였다면 해당 데이터 그램이 실제로 도용된 것인지 아닌지를 구분할 수 없게 되는 것이다. 이를 이용한 공격 방법이 IP Spoofing과 SYN Flooding이라고 할 수 있다[1]. 본 논문에서는 IP의 문제점을 이용한 공격을 TCP 순서번호를 암호화하여 접속 할당 IP에 일회용 키를 할당함으로서 IP의 인증 및 순서번호 암호화를 통해 시스템의 보안수준을 높이고자 하였다.

본 논문의 구성은 2장에서 기본 이론을 설명하였으며 3장은 제안된 내용을 기술하였으며 4장에서 결론을 맺도록 한다.

II. 기본 이론

2.1 IPv4와 IPv6

현재 인터넷을 구동시키고 있는 프로토콜은 TCP/IP이다. 이중 데이터의 실제적인 전송 및 경로 배정을 담당하고 있는 IPv4로서 라우팅에 의한 경로 배정, best-effort, 32비트 주소, TOS 필드에 의한 QoS 지원 고려 등이 그의 특징이다. 그러나 인터넷의 사용자가 기하급수적으로 늘어

남에 따라 주소 공간의 부족이 심각한 문제로 대두되었고 주소 고갈의 주요 원인 중의 하나인 주소 할당방식 역시 개선되어야 할 필요성을 가지게 되었다. 그리고 보안에 대한 대비가 전무하다는 것도 IPv4의 취약점이라고 할 수 있다. IPv6는 현재 IPv4에서 진화된 버전으로 향후 인터넷 각 장치의 IP프로토콜로서 쓰일 IP이다. 현재 쓰이고 있는 IPv4와 호환성이 있으며 점진적으로 확대되어 가고 있는 차세대 인터넷 망 프로토콜로서 쓰이게 되는 프로토콜이다. IPv6는 ATM과 같은 고속망 뿐만 아니라 고속망과 같은 저속의 망에서도 효율적으로 동작하도록 설계되었으며 앞으로 인터넷에서의 기본인 인터넷망에서 제공되는 기능 외에 다양한 기능으로 활용될 것이다 [2].

표 1은 IPv4와 IPv6에 대한 비교를 하였으며 그림 1은 IPv4 PDU를 계층별로 보여주고 있다.

	IPv4	IPv6
주소크기	32bit	128bit
Flow Label	No	Yes
Header Checksum	Yes	No
Fragmentation 정보	All datagrams	As an option

표 1. IPv4와 IPv6의 비교

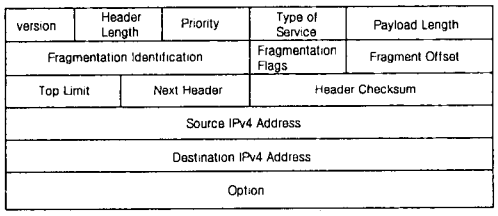

그림 1. IPv4 Protocol Data Unit(PDU)

2.2 DoS(Denial of Service)공격

Denial of Service란 multi-tasking을 지원하는 운영체제에서 발생할 수 있는 공격 방법으로서 구체적으로 한 사용자가 시스템의 리소스를 독점(hogging)하거나, 모두 사용해 버리거나, 파괴하여서 이 시스템이 다른 사용자들에게 올바른 서비스를 제공하지 못하게 만드는 것을 말한다. 그러므로 시스템의 정상적인 수행에 문제를 야기시키는 모든 행위를 Denial of service(DoS) 공격이라고 부르기 때문에 이를 위해서는 매우 다양한 방법이 존재할 수 있다. 여기서 한가지 재미있는 사실은 이러한 DoS가 고의적으로 발생할 수도 있지만 사용자의 의도와는 상관없이 실수로 발생 할 수도 있다는 사실이다. 비록 Denial of service는 시스템에 치명적인 문제(루트 권한의 획득, 시스템이나 사용자 데이터의 파괴나 변조)를 끼치지는 못하지만 시스템의 정상적인 수행에 문제(네트웍이나 시스템 서비스등의 마비)를 야기시킴으로써 사용자들의 많은 불편을 주게 된다. 그러므로 시스템 관리자들은 Denial of service가 고의적으로 발생했든 실수로 발생하게 되었든 이를 재빨리 감지하여 신속히 문제를 해결함으로써 사용자들에게 적절한 서비스를 제공할 수 있게 해주어야 한다. 하지만 대부분의 Denial of service공격의 특징이 공격을 감지하여 이를 막기가 매우 어렵다는 것이므로 이 공격의 심각성를 인식할 수 있다[3].

2.3 IP Spoofing

IP Spoofing은 TCP/IP 프로토콜의 구조적 결함, 즉 TCP 시퀀스 번호(sequence number), 소스 라우팅(routing), 소스 주소를 이용한 인증(authentication) 메카니즘 등을 이용한 방법으로서 인증 (authentication) 기능을 가지고 있는 시스템에 침입하기 위해 침입자가 사용하는 시스템을 신뢰성 있는 호스트(trusted host)로 위장하는 방법이다.

IP Spoofing을 이용한 침입은 패킷의 내용을 변경하여 스크리닝 라우터와 방화벽 시스템을 통과하는 단계, 자신을 신뢰성있는 호스트로 인식하도록 하는 단계, 그리고 트로이 목마 프로그램을 설치하는 등의 목적한 작업을 하기 위해 호스트의 접근 권한 혹은 가능한 루트 권한을 획득하는 단계 등의 세 단계로 구성된다.

IP Spoofing에 대한 예방책은 크게 탐지(dete-ction), 예방 (prevention), 그리고 복구(recovery) 등으로 나눌 수 있다. IP Spoofing의 탐지는 네트워크 모니터링 소프트웨어를 이용하여 외부에서 들어오는 패킷의 소스 IP 주소와 목적 IP 주소를 검사함으로서 발견할 수 있다. 만일, 특정 패킷이 로컬 도메인의 소스 IP 주소를 포함하고 있다면, 내부 네트워크는 침입자에 의해 침입을 당하고 있는 중일 것이다.

또 다른 탐지 방법은 내부 네트워크 내의 모든

시스템간에 이루어진 프로세스 로그(log)를 비교하는 것이다. 이렇게 IP Spoofing에 의한 침입 여부가 탐지되면, 대상 시스템 내에 설치된 트로이 목마 프로그램 혹은 백도어(backdoor)가 설치되어 있는지를 검사해야 한다. 혹은 침입자가 이미 사용하고 있는 커넥션을 가로채려고 하는 경우 터미널에 사용자가 입력하지 않는 명령이 출력되거나 윈도우 상에 입력한 명령에 대한 응답이 출력되지 않는다면 일단 침입을 당한 것으로 판단해야 한다[4]. 그림 2는 IP Spoofing 공격에 대한 형식을 보여주고 있다.

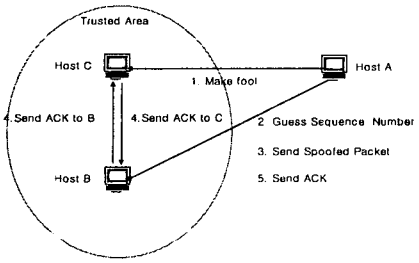

그림 2. IP Spoofing 공격

2.4 SYN Flooding

SYN Flooding 공격은 특정 시스템에 불법적인 권한을 얻는 능동적인 방법이 아니라 시스템이 정상적으로 동작을 할 수 없게 만드는 다소 수동적인 방법으로 일정의 서비스 거부 공격 중의 하나이다. 이것은 TCP가 데이터를 보내기 전에 연결을 맺어야 하는 연결 지향(connection-oriented) 방식이라는 점에 착안하여 많은 수의 SYN 비트가 설정되어 있는 즉, 연결을 요청하는 TCP 패킷을 호스트의 특정 포트에 보내어 이 포트의 대기 큐(backlog-queue)를 가득 차게 하여 이 포트에 들어오는 연결 요청을 큐가 빌 때까지(connection time out이 될 때까지) 무시하도록 하게끔 되어 있는 것이다. 큐의 크기는 시스템마다 다르지만 대략 5에서 10까지의 연결 대기 상태를 저장할 수 있다. 그러므로 실제 SYN Flooding 공격에서는 UDP storm, ping flood와 같은 다른 종류의 서비스 거부 공격과 같이 대량의 패킷을 보내지 않아도 되므로 공격이 쉽게 노출되어 있지 않는다. 또한 출발지 IP주소를 임의의 주소로 만들어서 보내므로 어디서 이러한 패킷이 오는지 알아내는 것도 매우 어렵다[4].

그림3은 네트웍에서 SYN Flooding 공격을 보여주고 있다.

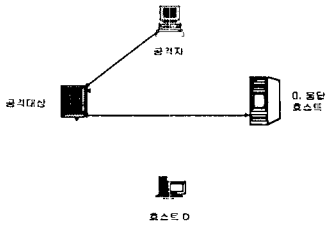

그림 3. SYN Flooding 공격

Ⅲ. 순서번호 암호화 방식 제안 및 검토

이러한 IP Spoofing공격이나 SYN Flood공격과 같은 것은 'Denial of Service' 공격의 기반이 되는 공격이라 볼 수 있으며 이는 하드웨어적인 시스템의 파괴가 아니더라도 훨씬 더 많은 금전적·시간적 피해를 입힐 수 있음을 확인할 수 있었다[6].

따라서 본 논문에서는 공개키 기반 구조하에서 접속 요구 IP가 가지는 공개키를 일방향 해쉬 함수 암고리즘으로 이용하여 접속요구 IP에 하나의 일회용 키를 분배하는 방법을 제안하였다.

파라미터

- A : 피 접속 요구 IP
- B : 접속 요구 IP
- K : 접속 요구 IP에 할당될 일회용 키
- r : 생성된 난수
- SYN : 순서번호
- Q : 접속요구 IP의 공개키
- G : 일회용 키 증명을 위한 증명키

양 사용자 A, B는 기존 TCP 체계에서 접속을 실시하고자 하는 두 객체이며 B는 이미 할당된 공개키를 이용하여 접속을 요구한다.

① 사용자 A는 사용자 B와 통신을 하기 위해 B 의 공개키를 요구한다.

② 사용자 A는 사용자 B의 공개키에 해당하는 난수를 발생 시킨다. 그림 4는 넓은 범위에서 작은 난수를 발생키고자 난수를 설정하는 그림이며 그림 5는 난수 설정에 따른 난수 발생을 보여주고 있다.

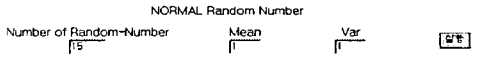

그림 4. 난수 설정 부분

```
NORMAL Random-Number
0.6476634514351613
0.6859401805761118
1.3166048250143238
0.7263550954715635
-0.3829449097253088
1.449008476881794
2.013215386736092
0.7220005434034089
0.9060782077196511
0.9067724702451453
-0.0866107100559370 1
0.2715900532485902
1.3462223776726514
1.2367799147796507
0.6401363536988447

Mean=0.8265874478067831
(1)
Var=0.3466068990044364(1)
SD=0.5887333004038725(1)
```

그림 5. 난수에 대한 결과값

③ 사용자 A는 G를 계산하여 이를 DB형태로 저장시킨다.

$$G = SYN^r \bmod Q$$

④ 사용자 A는 해쉬함수와 B의 공개키를 이용하여 B에 할당할 K를 생성 분배한다.

$$K = f(Q, r)\ SYN^r \bmod Q$$

⑤ 사용자 B가 접속 종료를 할 때 K는 소멸된다. (단. G는 사용자 A, B 사이에 분쟁시 참조를 위해 보관)

사용자 A, B 사이에 분쟁이 발생할 경우 사용자 A는 보관중인 G를 통해 사용자 B를 증명할 수 있었다. 이는 일회용 키 K를 증명할 수 있으며 사용자 A, B는 이를 통해 분쟁을 해결할 수 있음을 알 수 있다. 제시된 일회용 키를 이용하면 기존 TCP 순서번호의 암호화 방법뿐만 아니라 사용자간의 분쟁이 발생시에도 이를 증명할 수 있었다. 또한 접속이 종료 되었을 때 일회용 키는 삭제되어 키의 분실 뿐만 아니라 시스템의 부하 또한 줄일 수 있음을 알 수 있었다.

Ⅳ 결 론

공개키 기반 네트웍 시스템에서 공개키를 난수와 결합할 경우 기존의 생성번호 전체를 취급할 경우 많은 부하를 일으킬 수 있다. 사실상 초당 2,500,000번씩 순서번호가 증가하는 현재의 TCP 프로토콜에서 순서번호를 유추할 수도 있으며, 64K Rule이라 하여 SYN 패킷을 하나 하나 받을 때마다 순서번호가 64000씩 증가하는 운영 체제도 있다.

본 논문에서는 난수 발생으로 인한 시스템의 부하를 줄이기 위해 넓은 범위에서 작은 난수를 선택하여 이를 접속 요구 IP의 공개키, 순서번호와 결합 일방향 해쉬함수를 이용 일회용 키를 분배함으로써 순서 번호의 유추를 어렵게 하였으며 TCP 프로토콜 순서번호의 유추를 어렵게 할뿐만 아니라 IP에 대한 인증 효과까지 볼 수 있음을 알 수 있었다. 이 방식은 공개키 기반 구조에서 이루어지는 전자상거래, 인증 시스템에 이용가능하며 현재 시험중인 IPv4와 IPv6 연동망 및 IPv6에서 활용될 수 있으리라 사료된다.

Ⅴ. 참고문헌

[1] PLUS(포항공대 유닉스 보안 연구회)저, "Security PLUS for UNIX", 영진.COM, pp. 201-227 2000년 7월.

[2] 주식회사 니츠 편저, "인터넷 보안기술." 도서출판 동서, pp. 12-13, 2000년 1월.

[3] http://baram.chungnam.ac.kr/~yhjin/Management/SecurityPLUS-2nd/node7.html

[4] http://ns.yaknamu.co.kr/i&t/boan2.html

[5] 한남대학교, "네트워크 취약성 분석 및 개발 S/W 개발", 한국정보보호센터, 연구보고서, pp. 12-15 1999년 12월.

[6] http://www.hani.co.kr/section-010000000/2000/p010000000200002091337478.html

[7] 이만영 · 김지홍 · 류재철 · 송유진 · 염흥열 · 이임영, "전자상거래 보안기술", 생능출판사, 1999년 8월.

[8] 이임영 · 송유진 공역, "현대암호", 생능출판사, 1999년 2월.

[9] 이재호 · 구창회 · 김장권 · 이상훈, "정보통신망과 프로토콜", 복두출판사, 1998년 8월.