특집

인터넷 보안

박 성 균
미소테크(주)

들어가는 글

인터넷이 우리나라에 들어오고 최근의 인터넷의 상업화에 힘입어 기업, 정부, 학교 등등 여러 기관에서 인터넷에 접속하고 있으며, 기업체의 상용서비스및 구청과 같은 관공서들도 간단한 민원 업무를 서비스 하는등 관련 서비스의 종류도 다양화되어 가고 있다. 초기의 전화회선을 통한 접속만이 가능했던 PC 통신 서비스도 인터넷에서 접속 가능하여졌고, 여러가지 인터넷 서비스도 제공하고 있는 실정이다.

이러한 국내의 폭발적인 인터넷 사용인구의 증가와 규모가 커져감에 따라서 이에 따른 보안문제 역시 큰 관심사가 되었다.

우리나라의 인터넷에 관련된 보안 사건을 대략 살펴보면, 93년에 있었던 서울대 중앙교육전산원의 LAN에 침입하여 6대의 워크스테이션의 정보를 지운 서울대 중앙교육전산원 침해사건과 당시 HANA망을 운영하던 한국통신 연구센터의 자료를 지운 한국통신연구센터 침해사건, 94년의 인천지역 정보망인 인디텔에 가입한 선배의 아이디를 도용하여 홈뱅킹 계좌이체를 시도하다 붙잡힌 천리안 홈뱅킹 사건, 95년 2명의 부산지역 해커가 부산지역을 비롯하여 전국 주요 대학의 시스템을 해킹하다 붙잡힌 사건과 96년의 한국과학기술원과 포항공대사이의 사건과 PC 통신망을 홈뱅킹을 통한 사건등 널리 알려지고 신문지상에 보도 되었던 몇가지 사건들이 있었다.

그러나, 이렇게 언론을 통하여 겉으로 보도된 사건들 만이 전부는 아닐 것이다. 이러한 인터넷 보안 기술의 미비와 보안에 대한 인식이 제대로 형성되지 않음으로 인하여 그 뒤의 밝혀지지 않거나 심지어 피해를 입었는지에 관한 여부 조차 모르고 지나간 경우도 많을 것이라고 생각된다.

인터넷에 접속하고 어떤 서비스를 제공할때, 그에 대한 보안에 대한 고려도 이제는 매우 중요한 일이 되었다. 따라서, 이러한 보안 마인드의 부족으로 인하여 이러한 점을 대수롭지 않게 생각하다가 피해를 당하였을때는 그때는 이미 늦어버린 것

이다. 이러한 보안 의식과 과연 어떠한 문제점으로 인하여 공격을 받는가에 대한 지식을 통하여 한번 더 인터넷 보안에 관한 의식이 필요한 것이다.

이 글에서는 인터넷에 보안에 관하여 간단한 공격방법들에 대한 설명과 보안성 향상을 위하여 많이 사용되는 툴들에 대한 소개, 마지막으로 요즘 많이 이슈가 되어지는 암호화와 인증에대한 기초적인 설명과 소개를 하도록 하겠다.

I. 인터넷에서의 보안

여기서는 인터넷에 관련된 보안의 기술적인 문제점들 중 널리 알려진 몇가지를 간략히 설명하고자 한다. 인터넷 보안에 가장 크게 관련되는 것들은, 인터넷을 이루고 있는 네트워크와 이러한 네트워크에 연결된 컴퓨터들의 보안 약점이 될것이다. 먼저 이러한 보안약점을 공격하는 해킹 유형을 중심으로 살펴보자.

기술적으로 해킹의 유형을 내부시스템에서의 공격과 외부시스템에서의 공격으로 크게 두가지로 나누어 다루어 볼 수 있다. 물론 이러한 두가지 약점을 동시에 공격하는 유형도 있고, 이 두가지 분류가 절대적이지는 않다. 다만, 설명상의 편의를 위한 구분임을 밝혀 두겠다.

1. 내부 시스템에서의 공격 (local attack)

내부 시스템에서의 해킹의 유형은 대부분 내부에서 시스템을 사용할 수 있는 사용자(시스템을 사용할 수 있는 권한(계정)등을 가진)에 의한 해킹 유형이라고 할 수 있겠다. 이러한 유형은 대부분 시스템의 프로그램이 가지고 있는 문제점(Bug)에 의하여 공격받는 형태라고 하겠다. 또한, 관리자의 보안에 관한 무관심이나 무지에 의한 잘못된 시스템의 설정과 관리 등등에 의해 쉽게 공격을 받을 수 있다.

인터넷을 이루고 있는 컴퓨터들에서, 서비스를 하고 있는 대부분의 컴퓨터는 UNIX기반의 운영체제를 사용하고 있다. 최근 들어 MicroSoft Windows NT 또한 점점 많이 사용되어가는 추세이다. 그러나, 아직까지는 UNIX기반의 컴퓨터들이 대부분을 차지하고 있고, 내부 공격이라고 할 수 있는 대부분의 문제점들은 여러 UNIX의 문제점과 그리고 그 운영체제위에서 동작되는 응용 프로그램 내부의 문제점에 의하여 공격받는 형태라고 할 수 있다.

먼저 응용 프로그램상의 문제점을 이용한 공격을 살펴보자, 이러한 대부분의 문제점들은 각 단체의 보안 권고문(advisory)들에 의하여 보고 되어지고, 관련된 UNIX업체에서 이에 관련된 patch를 제공하고 있다. 그러나, 이러한 보안 권고문과 관련 patch들이 발표 되더라도 시스템의 관리자가 보안의식이 부족하다면, 쉽게 지나치기 쉽고 이러한 것들에 대하여 관심을 갖지 않는다면 이런 약점에 대해 노출되었다고 할 수 있겠다. 물론, 이러한 권고문에 관심을 갖고 즉각적인 처리를 한다고 하더라도 권고문에 발표되지 않은 문제점에 대하여는 방비책이 없는 것이다. 또한, 관리자의 보안에 관련한 무지에 의한 잘못된 시스템 설정과 관리에 의하여 시스템이 약점을 가질 수 있다.

그렇다면, 널리 알려진 대표적인 문제점들을 살펴보자.

1) IFS attack

이 문제점은 매우 잘 알려진 것들 중의 하나이다. 환경변수중 IFS의 값은 파일간의 구분을 짓는 구분자(delimiter)이다. UNIX system call들 중에 system(), popen()과 같은 call등이 이러한 IFS값을 참조하게 되는데, 이러한 IFS값을 임의로 변경하여 이러한 call등이 들어있는 프로그램을 실행시키게 되어 문제가 발생하는 것이다.

예를 들어, 프로그램의 코드(code)중에 system ("/bin/sh");과 같은 명령이 들어있다면 IFS의 값을 "/"로 변경하여 " bin sh"을 실행시키는 엉뚱한 일이 발생할 수 있는 것이다.

2) Race Condition

race condition에 대하여 설명하기 전에 먼저 symbolic link에 관한 문제점을 설명하도록 하겠다. 어떠한 관리자(root) 권한으로 실행되는 프로그램이 내부에서 임시적인 파일을 생성하여 자료

를 쓰게 되는 경우, 그러한 파일이 생성되는 장소가 /tmp 디렉토리같은 내부 사용자에 의하여 쓰기 권한이 있는 곳일 경우 내부 사용자가 그러한 파일이름으로 다른 시스템 파일들에 symbolic link를 설정해 놓았을 경우 symbolic link 에 설정된 파일들이 간단히 조작되어지는 문제점을 가지게 되었다.

이러한 문제점을 방비하기 위하여, 이런 문제점을 가진 프로그램들이 임시파일을 만들 때 symbolic link 의 여부를 살피도록 하는 code를 넣었는데 이것 또한 race condition을 고려하지 않은 생각이었다.

race condition 이라 함은, 여러 프로세스가 한 자원을 놓고 경쟁하는 경우에 있어서 그 결과는 프로세스 스케쥴링에 의해 여러가지 상황이 나올 수 있는 것이다. 이러한 문제점은 유닉스 시스템 설계 당시에도 있었고, 이를 해결하기 위하여 atomic function과 같은 것들을 도입하였다.

따라서, 위의 해결책이 완벽한 것은 아님을 알 수 있다. 즉, 한 프로그램이 link와 unlink를 반복한다면, 링크체크와 파일에 쓰는 작업 사이에 이러한 race condition에 의하여 결과가 어떻게 나올지 예상할 수 없는 것이다.

3) Buffer Overflow attack

최근의 각종 보안 단체의 권고문을 살펴보면 대부분이 overflow 에 의한 약점에 관한 내용임을 알 수 있다. 물론, 상황이 심각한 만큼 각각의 관련 업체에서도 이에 대하여 급히 대책을 내놓았다.

Buffer overflow의 공격방법은 프로그램이 관련 옵션이나 어떠한 스트링 값을 input으로 받을때, 이 정보가 프로그램 실행시 stack에 저장되는 과정에서 그 값의 길이를 체크하지 않음으로 인하여, stack을 overflow 시켜서 return address 값이 저장되는 곳에 임의의 값을 넣을 수 있어 이러한 문제점으로 인하여 임의의 명령을 실행 시킬 수 있는 것이다. 대부분 shell 을 실행시키는 shell code를 삽입하여 시스템 관리자 권한의 shell을 실행시킬 수 있게 된다.

이러한 buffer overflow attack은 system내부의 getopt(), gethostbyname() 등 여러 function call에 문제점을 가지고 있어 이러한 call을 사용하는 root setuid를 가진 프로그램 대부분이 이 문제점을 가지고 있다.

2. Romote attack

remote attack이라 함은 일단, 공격하고자 하는 시스템에 사용권한을 가지고 있지 않은 경우, 다른 곳에서 시스템을 공격하는 방식이다.

이러한 경우 인터넷의 기반이 되는 TCP/IP의 알려진 문제점에 의한 공격을 생각할 수 있다. 그리고, 시스템 내부에서 외부, 외부에서 내부로의 서비스를 제공하는 프로그램들이 가진 문제점에 의하여 공격을 받을 수도 있다.

이러한 문제점들도 어떻게 보면 내부의 문제점이라고도 할 수 있는 것들이 상당부분 포함되어 있다. 다만, 외부에서 이런 문제점을 공격할 수 있기에 remote attack 이라고 소개해 보겠다.

1) sendmail

sendmail 은 거의 모든 UNIX기반의 운영체제에서 사용되는 전자우편 관련 서비스를 제공하는 대몬 프로세스(daemon process)이다. 그 만큼, 이것을 통한 공격 방법이 매우 많이 존재하며 이에 관련된 문제점 또한 여러종류가 보고 되었다. 이에 따라 수시로 업그레이드 되었으며, 주의를 기울여 즉각적인 업그레이드와 관리가 절실히 필요한 것이다.

이에 한가지 문제점으로 버전 8.6.9, 8.6.10에서 메일을 주고 받을때 사용자의 인증을 하기 위한 ident 프로그램을 변조하여 조작된 정보를 보내어 임의의 코드를 수행하는 것이 가능한 문제점을 가지고 있었다. 이 외에도 여러 문제점이 존재하고 초창기에 가장 공격의 대상이 되었던 것이다.

2) wu-ftpd

anonymous ftp service 또한 많이 제공되고 있는 서비스의 한 종류이다. 이 서비스를 제공하는 wu-ftpd 역시, 여러가지 문제점을 가지고 있다. 한가지 예를들어, quote와 site 명령을 이용하여 외부에서 프로그램을 동작시킬 수 있는 문제점이 발견되었다.

이런 anonymous ftp와 같은 서비스를 제공할

때에는 신중히 configuration등을 설정할 필요가 있다. 그리고, 항상 log파일 등을 통해 상황을 살피는 것이 바람직하다.

3) httpd

인터넷 붐을 일으킨 대표적인 서비스로 이에 관련된 문제점을 알아 보면, www 서비스를 제공할 때, httpd의 자체의 결함으로 인한 문제점과 잘못된 설정으로 인한 문제점들이 존재하고, 또한 취약점을 가진 CGI 스크립트 등과 같은 프로그램을 통하여 공격을 받을 수 있다.

CGI(Common Gateway Interface)에서 가장 널리 알려진 문제점들은 phf, nph-test-cgi 등이 있다. 이러한 것들은 주로 code내부에 사용된 system(), popen() 같은 call들에 의하여 원치 않은 명령들이 실행되어 질 수 있는 것이다.

4) TCP/IP의 문제점에 의한 attack

이제 더이상 현재의 TCP/IP(IPv4) 체계를 통한 통신은 안전하다고 볼 수 없다. 이에 관한 몇가지 문제점들은 많은 문서를 통하여 보고되었고, 그를 이용한 해킹 방법들도 많이 등장하였다. 여기서는 현 체계의 TCP/IP 의 취약점들을 알아보고 이런 약점들을 이용한 공격 방법들이 어떤것들이 있는지 살펴 보고자 한다.

(1) TCP/IP의 문제점

먼저 IP(Internet Protocol)의 약점을 살펴보자. 여러가지 문제점이 있으나 여기서는 보안 측면에 관련된 두가지 문제점을 소개하겠다.

첫째, 패킷의 인증과정이 존재하지 않는다. 이것은 어떠한 패킷을 IP 를 통해 보낼때 Source address와 destination address를 설정하여 보내게 된다. 그런데, 이러한 Source, Destination address를 어떠한 호스트에서도 임의로 지정하여 보낼 수 있다. 이러한 문제점은 실제로 패킷을 보내는 호스트가 아니라도 얼마든지 위조하여 보낼 수 있다는 점이다.

둘째로, 전송되는 패킷의 내용이 중간에서 읽혀질 수 있다는 점이다. 중간에서 이러한 패킷을 가로채어 내용을 볼 수 있다는 의미로 암호화가 되지 않은 상황에서 IP를 통해 보내는 모든 패킷들은 안전하다고 볼 수 없다. 실제로 이러한 문제점으로 인하여 sniffing과 같은 방법들이 상당히 문제시되었다.

TCP의 약점을 들어보면, 현 체계의 TCP connetion을 이루는 처음 과정인 3-way handshaking을 들 수 있다. 간단히 설명하면, TCP connection이 이루어 지는 과정은 먼저 Client쪽에서 SYN message를 보내면 Server쪽에서 이에 대한 응답으로 SYN + ACK message를 보내게 되고 마지막으로, client쪽에서 ACK message를 보내서 connection이 이루어 지게 된다.

그런데, 이 과정의 중간인 Server가 SYN + ACK message를 받고 client의 ACK message를 기다리는 과정이 있는데(이러한 상태를 half-open connection 이라 한다.) 이때 Server는 이러한 정보를 자신의 저장장치에 기록해 두었다가, 어느 시간이 지나도 응답이 없는 경우에 이 정보를 삭제하게 되는데, 이러한 경우에 의도적으로 많은 SYN message를 보내고 ACK message 를 받을 수 없도록 하여 시스템을 내부적으로 Overflow 시키는 공격방법이 존재한다. 이러한 공격 방법은 TCP SYN flooding 이라는 방법이다. 이런 공격은 서비스를 마비시키거나 시스템을 crash시키는 것으로 denial of service라고 불린다. 이외에 IP spoofing에 관한 공격 방법이 있는데 이는 아래에서 다시 설명하도록 하겠다.

이제 위의 TCP/IP의 약점을 이용한 공격 패턴을 알아보자.

(2)packet sniffing

네트워크 인터페이스를 promiscuous mode로 설정하여 sniffit 과 같은 프로그램을 통하여 서브넷을 통과하는 모든 패킷의 내용을 알아 볼 수 있다. 실지로 이러한 프로그램을 통하여 login 과정 중의 password등과 같은 통신과정의 모든 중요한 대이터들을 중간에서 볼 수 있는 것이다.

(3)denial of service

이 방법은 위에서 설명한 TCP SYN flooding과 같은 공격을 하여, 상대방의 서비스를 엉망으로 만들어 버리는 방법이다.

이외에서 이러한 서비스를 마비시키는 공격 방

법으로는, Mail bomb, ping of death, ftp signal attacking, socket-connect attack등 다양한 방법이 있다. 그중 간단히 앞의 두가지를 살펴보면, Mail bomb이란 간단히 특정 호스트에 garbage 메일을 계속 보내 그 시스템의 /var 또는 /partition을 full 시켜 system을 마비시키는 방법이다. 또, ping of death라 함은 크기가 큰 ICMP 메시지를 공격하고자 하는 호스트에 보내서 내부적으로 overflow를 일으켜 서비스를 마비시키는 방법이다.

(4)IP spoofing & Hijacking

이 공격 방법은 상당히 높은 수준의 공격이라고 할 수 있는데, 그 대략적인 공격 방법은 다음과 같다.

이것은 TCP의 connection, close, reset과정을 Trust host로 부터 온 것처럼 속여 보내는 방법이다. A 라는 서버 호스트와 B 는 A와 telnet으로 연결하고 있는 client라 하고, C는 공격을 하는 호스트라고 하자. C 는 A 와 B 사이에 위치하고 있어서 A와 B 사이의 모든 패킷을 받아 볼 수 있다고 가정한다면, 다음의 순서를 통하여 이러한 공격이 이루어 진다.

먼저, B라는 호스트에서 A로 접속하여 사용하고 있을때, 이때, C에서 B로 부터 reset을 보낸것처럼 패킷을 보낸다. 그러면, A는 자동으로 connetion을 끊게 되고, B는 A와 연결이 아직도 이루어진 것처럼 생각하게 된다. 그런 다음, C에 B에서 Connection 을 새로 만드는 것처럼 SYN 패킷을 보낸다. 이렇게 되면, A는 B와 connection이 없는 상태에서 다시 연결을 시도하는 것으로 생각하게 된다. 따라서 A는 자동으로 SYN + ACK 패킷을 B에 보내게 된다. 그러나, B는 아직 connetion이 이루어진 상황으로 생각하고 있으므로, 이 SYN + ACK 패킷을 이해하지 못한다. 이때 이 패킷을 가로채서 SYN + ACK 패킷에 같이 날아온 Sequence number에 따라 ACK 를 보내준다. 날아온 Sequence number와 자신이 보냈던 Sequence number를 참조하여 정확한 패킷을 B에서 보내준것 처럼 A 에 보내주게 되면 A는 B 와 connection이 이루어진 것으로 생각하게 된다. 또한, B 도 여전히 connetion이 이루어진것으로 생각하게 되는 것이다. 이제, A에서 B로 가는 패킷을 가로채어 B에 연계를 시켜주고, B에서 A로 가는 패킷 또한 C에서 가로채어 A로 넘겨주게 된다.

이러한 과정에서, A에서 B로 어떤 데이터를 요구하는 경우, C에서 날아오는 패킷을 받아서 B에 연결하여주면 B는 그것에 답하게된다. 그리고, 이러한 방법은 Sequence number에 의하여 이루어지기 때문에 A에서 보내는 request에 대하여는 B는 받아보지 못하는 것이다. 즉, 처음에 A와 B 사이의 통신에 사용하던 Sequence number가 변하기 때문이다. 즉, A에서 보낸 패킷이 B로 가더라도 B가 그 패킷을 받을 수 없고, 그 반대도 마찬가지이다.이러한 방법으로 C는 원하는 패킷을 보낼 수 있게 된다.

II. 보안 도구

지금까지 local, remote 공격 방법의 형태를 간단히 알아보았다. 이렇듯 공격방법은 그야 말로 다양하고, 지속적으로 생겨나고 있다.

CERT, CIAC 등 여러 단체에서 나오는 Advisory들이 이러한 모든 문제점들을 보고하지는 못한다. 어떠한 Advisory가 나오더라도 그 문제점이 이미 Advisory가 나오기 전에 사용된다면 이러한 Advisory에 모든 보안을 의지하는 것이 문제가 있다.

이에 따라, 관리자는 이러한 문제점을 보다 개선하기 위하여 많은 보안 프로그램이나 관리 프로그램을 통하여 좀더 보안성의 향상을 볼 수 있다. 그 대표적인 것들로 요즘은 firewall같은 것들이 있을 것이다. 여기서는 firewall을 비롯 널리 사용되는 보안 툴들에 대하여 소개하고자 한다.

1. security tool

보안 프로그램은 주로 알려진 문제점에 대한 검사와 시스템 파일들에 대한 설정의 문제점들을 주로 검색하여 준다. 크게 local, remote system의

문제점에 대한 scanning tool과 system monitoring tool, 유지관리 tool을 들 수 있다. 각각에 대한 몇가지 널리 사용되는 tool에 대해 간단히 소개하겠다.

(1) local, remote system tool

crack은 주로 패스워드에 대한 관리를 한다. 쉽게 추측 할 수 있는 패스워드를 찾아 주는 tool이라 하겠다.

cops 또한 내부시스템의 안전을 검사해주는 tool로 간단한 패스워드 추측, 잘못된 시스템 파일들을 검사해주는 툴이다. 이에 관한 상세한 정보와 프로그램은 아래 URL에서 얻어 올 수 있다.

ftp : //ftp.cert-kr.or.kr/pub/Security/tool/cops-104.tar.Z

ISS에 만든 SafeSuite는 상용 security scanner로 local, remote 등 전반적인 문제점들을 찾아 준다.

SATAN 역시 유명한 tool로 한때 해킹에 이용되기도 했다. 이 SATAN은 아래 URL을 통하여 얻을 수 있다.

ftp : //ftp.cert-kr.or.kr/security/tools/satan-1.1.1.tar.Z

(2) monitoring 및 유지관리 tool

관리자는 시스템의 보안성 향상을 위하여 시스템을 모니터링할 필요가 있다. 이에 도움이 되는 tool이 ttywatcher로써 사용자의 tty를 직접적으로 모니터링할 수 있는 강력한 tool이다. 또한 netlog, CPM등이 있으며 자세한 정보는 아래의 URL을 참조하기 바란다.

ttywatcher http : //nad.infostructure.com/watcher.html

netlog ftp : //net.tamu.edu/security/netlog-1.2.tar.gz

CPM ftp : //cert.org/tools/

또한, 시스템의 주요파일들의 변조 유무를 관리하는데 도움을 주기위하여 checksum 정보에 의하여 이러한 파일들의 변조 유무를 관리하는 tripwire, MD5등의 tool 이 널리 사용되고 있다. 이들은 아래를 참조하여 얻을 수 있다.

tripwire ftp : //coast.cs.purdue.edu/pub/COAST/Tripwire/tripwire-1.2.tar.Z

MD5 ftp : //ftp.cert-kr.or.kr

2. firewall

요즘들어 이러한 내부 외부적 공격방법에 대하여 내부 네트워크를 보호하기 위한 firewall이 많이 사용되어지고 있다. 파이어월은 내부 네트워크를 외부 네트워크에 대한 격리를 하는 것이다.즉, 네트워크 입구에서 외부의 패킷이 내부 네트워크로 들어오는 것을 막는 것이다.

firewall은 외부 네트워크와 내부 네트워크의 경계에 위치하면서 내부에서 외부, 또는 외부에서 내부로 나가고 들어오는 패킷을 일괄적으로 처리한다. 즉, 내부, 외부간의 통신이 모두 firewall을 거쳐하는 방식이라면 firewall은 이 병목점 같은 지점에서 적당히 packet filtering 하여 내부 네트워크를 보호할 수 있다.

III. 암호화

요즘들어 전자상거래, 전자 결제등 이러한 서비스를 인터넷상에서 제공하면서, 위의 TCP/IP를 통한 packet이 읽혀 질 수 있다는 문제점으로 인하여 이러한 packet 을 암호화하여 보내는 기술등, 암호화에 대한 기술이 상당한 이슈로 부상되고 있다. 여기서는 널리 사용되는 암호화 알고리즘과 기술들에 대한 간략한 소개를 하도록 하겠다.

1. 공개키 암호화와 디지털 서명

전통적인 암호화 방식은 암호화때와 복호화(decryption)때의 암호가 다르도록 되어 있었다. 이러한 암호화 방식을 대칭형 암호화(symmetric cryptography) 또는 비밀키 암호화(secret-key cryptography)라 부른다. 그러나, 송신자와 수신자가 멀리 떨어져 있다면 비밀키를 안전하게 전달하는데 있어서 어려움을 겪는다. 이러한 문제점을 해결하기 위하여 나온 개념이 공개키 방식의 암호화이며 1976년에 Whitfield Diffie와 Martin

Hellman에 의하여 제안되었다.

디지탈 서명이란, 우편으로 말하자면, 보낸 우편의 주소에 쓰여진 사람이 정말로 우편을 보냈는지 확인할 수 있게 하는 방법이다. 단순한 공개키 암호화 방식에서는 A라는 사람이 B에게 우편을 보내지 않았는데, C라는 사람이 중간에 속여서 보낼 수 있는 문제점이 있다. 이러한 문제를 해결하기 위하여, 송신자를 정확하게 확인하기 위한 방법으로 디지탈 서명이 사용된다.

비밀키 방식에의 비밀키를 전달하는 과정에서 노출의 우려가 있지만, 공개키 암호화 방식에서는 비밀키를 누구에게도 가르쳐줄 필요가 없이 개인적으로만 알고 있으면 되기 때문에 위의 문제점을 방지할 수 있다. 다른 또하나의 장점은 공개키 암호화 방식이 디지탈 서명의 방법을 제공한다는 것이다.

(1) RSA 알고리즘

대표적인 공개키 암호화 방식이며 미국내 저작권은 RSADSI사가 소유하고 있다.

RSA알고리즘은 실제 구현에서는 속도가 느리기 때문에 DES와 같은 대칭형 암호체계와 함께 사용되는 경우가 대부분이다. 이런 것을 digital envelop이라고 부른다.

예를 들어 A가 B에게 메시지를 보내고 싶으면 내용을 DES로 암호화하고 그 다음 암호화할때 사용한 키를 RSA를 암호화하여 보내는 것이다. 이렇게 하여 DES의 빠른속도와 RSA의 공개키 방식의 장점을 같이 취할 수 있다.

한편 주의할 점도 있다. RSA 방법의 단점으로는 어떤 메시지를 도청했을때 그것을 풀 수는 없어도 원하는 메시지를 암호화할 수는 있다는 것이다.

이와 같은 문제점도 해소하고 RSA알고리즘 적용의 표준방식을 정하기 위해, RSADSI사에서 개발한 표준이 있다. 그 표준의 이름은 PKCS인데, http : //www.rsa.com/rsalabs/pubs/PKCS/에서 읽어 볼 수 있다.

이제, RSA 알고리즘을 인증에 사용하는 방법에 대하여 알아보자, 쉽게 예를 들어 설명하면, A가 B에게 메시지를 보낸다고 가정하자. 그 메시지 자체의 hash function을 적용하여 얻을 것을 그 메시지의 digital fingerprint라 하면, 그 digital fingerprint를 A의 비밀키(private key)를 이용하여 암호화 한다. 이것을 digital signature라 할 수 있다.

이 digital signature를 받은 B는 그것을 A의 공개키(public key)로 풀어보게 된다. A가 아닌 사람은 A의 비밀키를 알 수 없으므로 만일 풀어서 나온 digital fingerprint가 그 메시지의 fingerprint와 같다면 A가 그 메시지를 보냈다고 확신 할 수 있게 된다.

실제 적용에 있어서 이런 식으로 내용을 암호화하지 않으면 다른 사람도 digital fingerprint의 값을 알 수 있게 되어 A의 비밀키가 풀릴 위험성이 있다. 그점을 해결하기 위해서는 본문의 내용은 A가 사인을 한 다음 암호화를 하면 된다.

이 방식을 적용하기 위해서는 정확하게 각 사람의 public key가 변하지 않아야 한다. 어떤 사람이 자기의 public key를 마음대로 바꿀 수 있다면 사인 해놓고서도 하지 않았다고 할 수 도 있는 것이다. 이때 필요한 것이 믿을 수 있는 제 3자로, key server 같은 것들이 이러한 역할을 한다.

(2) Diffie-Hellman

Diffie-Hellman 공개키 암호화는 사실 RSA보다 먼저 개발된 것이고 현재까지 사용되는 가장 오래된 공개키 암호화 방식이다. 이 방식에서는 내용이나 서명을 암호화(encryption)하지 않아서 RSA보다는 좀 덜 일반적이긴 하지만, 속도 면에서 유리하다. Diffie-Hellman 암호화 방식은 그렇게 많은 기능이 없다. 단지 두 사람이 같은 비밀키를 공유할 수 있도록하는 기능을 가지고 있다.

A와 B가 서로 신문광고면으로 이야기를 주고받는데 공유하는 비밀키를 가지고 싶다고 하자. 먼저 A가 B에게 소수 p와 generator g를 보내기 위해 신문 광고를 낸다. 그 다음 B가 받고 승락을 하면, 각자 난수 하나를 생각한다. A는 a, B는 b라는 난수를 생각했다고 하자. 그러면 신문광고를 통해서 A는 $g \cdot a \pmod p$값을 보내고, Bob은 $g \cdot b \pmod p$ 값을 보낸다. 이제 두 사람이 공유할 수 있는 비밀키는 $g \cdot ab \pmod p$이다. A는 $g \cdot ab = = (g \cdot$

b)·a (mod p)를 통해 구할 수 있고, Bob은 g·ab ==(g·a)·b (mod p)를 통해 구할 수 있다. 다른 사람이 이 메시지를 가로채더라도 실제 g·a, g·b 값만 가지고서는 g·ab를 계산하기 어렵기 때문에 보안성이 유지된다.

실지로 이러한 알고리즘을 사용하는 방법은, 예를 들어 이제 공유된 공개키를 DES의 키로 이용하여 그 다음부터 통신에 그 키를 사용할 수 있게 된다. 그러나, 여기서 A가 아닌 다른 사람이 A인 것처럼 신문에 광고를 낸다면 문제가 생긴다. 이러한 공격 방법을 middle person attack이라고 하는데, 이 문제가 발생하는 원인은 인증기능이 없기 때문이다. 이 문제를 해결하기 위하여 Diffle, van Oorschot, Wiener는 1992년에 인증기능이 있는 Diffie-Hellman 키 공유 프로토콜 (STS protocal)을 개발 하였다.

(3) DSA

DSA(Digital Signature Algorithm)은 NIST (National Institute of Standards and Technology)에 의해 만들어진 표준인 DSS (Digital Signature Standard)에서 인증에 사용되는 알고리즘이다. DSA는 Diffie-Hellman 알고리즘을 개량하여 만들어진 El Gamal에 의해 개발된 알고리즘을 응용한 것이다. DSA는 오직 인증만을 위해서 사용된다.

RSA와 비교하면 몇가지 차이점이 있다. DSA는 서명을 하는 시간이 서명을 검증하는 시간보다 더 빠르지만, RSA는 반대이다. 서명을 검증하는 횟수가 많을 가능성이 있다는 점에서 대부분의 사람들은 RSA가 더 편리하다고 하는데, NIST의 주장은 다르다. 생성하는데 유리하다고 생각한다. 한편, DSA의 경우는 아무런 특허가 걸려 있지 않기 때문에 RSA에 비해서 편하게 쓸 수 있다는 장점은 있다.

하여튼, DSA에 관해서는 많은 비판이 오가고 있다. 왜 NIST가 RSA를 이용하지 않고 El Gamal의 알고리즘을 개량한 것을 사용했는지 의문을 표시하는 사람도 있다.

이외에도, 앞에서 DSA를 소개할 때 이야기 했던 El Gamal 인증 알고리즘이 있다. 그외에도 Elliptic Curve 암호화체계나, LUC 암호화 체계, Knapsack 암호화체계, algebraic coding theory에 기초한 McEliece 암호화 알고리즘, Merkle의 Tree signature 방식, 확률적 암호화 방식, Rabin의 서명 체계등이 있다.

2. 블럭 암호화

블럭 암호화 체계는 어떤 고정된 길이의 평문(블럭)을 암호문의 블럭으로 대응시켜서 암호화를 하는 방법을 말한다. 이때 암호를 풀기 어려우려면 이 대응시키는 함수가 one-way 함수, 즉 한 쪽 방향은 계산이 쉬우나 역함수는 계산하기 상당히 어려운 것을 써야 한다.

(1) DES

대표적인 블럭 암호화 방식 중의 하나이고 1977년 미국 정부가 공식적으로 인정한 암호화 체계이다. DES는 대칭형 암호체계, 즉 암호화 하는 키와 복호화 하는 키가 같다.

DES의 블럭 사이즈는 64비트인데, 이 중 56비트를 암호화에 사용하고 나머지 비트는 패리티 비트로 사용한다. 16라운드의 Feistel cipher를 하는데, 하드웨어 구현에 적합하게 만들어져 있다.

(2) IDEA

International data Encryption Algorithm의 약자로 Xuejia Lai와 James L.Massey에 의해서 개발되었다. IDEA는 소프트웨어로 구현할때 효율적이게 설계되어 있으며, 128비트이다. 비교적 최근인 1991년에 개발되어 아직까지 알려진 중대한 약점은 없다.

(3)GOST

러시아의 국가 표준인 암호화 알고리즘이다. DES와는 달리, classified된 문서의 암호화에도 사용할 수 있다고 규정되어 있다. 알고리즘은 DES와 유사한데, 64비트의 블럭을 사용하고, 32개의 라운드를 이용한다. DES의 키는 56비트이지만, GOST는 256비트의 primary 키와 512비트의 secondary키를 이용한다.

(4) Blowfish Encryption Algorithm

64비트 블럭과 32비트에서 448비트까지 다양한 크기의 키를 이용할 수 있는 대칭형 암호화 알고

리즘이다. 개발자는 Applied Cryptography라는 책을 쓴 Bruce Schneir이며, DES나 IDEA보다 훨씬 빠르다고 한다. 1993년에 개발되었고, ssh등에도 사용되고 있다.

(5) CAST-128

DES와 유사한 암호화 알고리즘으로, rfc 2144에서 관련 자료를 찾을 수 있다. 12 혹은 16라운드 Feistel cipher의 일종이며 블럭의 크기는 64비트이고 키의 크기는 128비트까지 가능하다.

이 외에도 RC-4, SAFER K-64등 몇가지 알고리즘이 있다.

이러한 블럭 암호화는 고정된 길이의 데이타에만 적용할 수 있으므로 임의의 길이의 데이타를 다루려면 여러 블럭으로 나누어야 할 것이다. 이 나누는 방법에 여러가지가 있는데 그 중 널리 쓰이는 4가지로 ECB(Electronic Code Book), CBC (Cipher Block Chaining), CFB(k-Bit Cipher Freedback Mode), OFB(k-Bit Output Feedback Mode)가 있다. 이중 CFB 모드나 CBC모드가 널리 쓰인다.

3. SSL(Secure Socket Layer)

SSL은 secure socket layer의 약자로서 실제로 위에서 언급된 알고리즘을 사용하여 전자결제, 수강신청 등등 오고가는 데이터가 보안이 필요한 WWW 서비스에 실제로 사용되고 있다. 그중 이러한 접속을 하기위한 프로토콜 2개(SHTTP, SSL) 중 하나이다. Netscape사에서 개발하였는데, 주로 SSLeay-8.8.0 을 사용하고 있다. 라이브러리에는 RSA, DES, blowfish, IDEA등을 지원하고 있다.

VI. 맺음말

위에서 언급한 내용들은 대부분이 널리 알려진 것들이고, 이것들을 중심으로 설명하였다. 이외에도 인터넷 보안에 관련된 내용은 많은 부분이 있다. 실제로 인터넷을 서비스하고 있는 기업체라든지 보안성 향상이 필요한 곳들은 이러한 내용을 숙지하고 좀더 보안에 신경을 쓰도록 하여야 한다.

인터넷에 연결하여 서비스를 제공하는 system의 관리자건 사용자건 좀더 이러한 보안 지식과 보안 의식을 가지고 서비스를 제공하고 사용한다면 보안은 그렇게 어려운 문제만은 아니다.

참 고 문 헌

[1] Charlie Kaufman 외, Network Security private communication in a public world, Prentice Hall.

[2] Chapman & Zwicky, Building Internet Firewalls, O'Reilly & Associates, Inc.

[3] W. Richard Stevens, Advanced Programing in the UNIX Environment, Addison-Wesley.

[4] Comer, Internetworking with TCP/IP Volume 1, Prentice Hall.

[5] 한국정보보호센터, 정보시스템 해킹 현황 및 대응, 7 ~ 13 page, 11월, 1996년

[6] 보안 전문 웹진 Login, http://login.misotech.com

[7] 한국정보기술원, 보안강화 대책 및 운영방안, 4월, 1997년

저 자 소 개

朴 盛 均

1958年 3月 26日生
1991年 3月 고려대학교 경영대학원 졸

1992年 3月～1994年 12月 (주)우성/기획조정실
1995年 7月～현재 미소테크(주) 대표이사

주관심 분야 : 통합 Networking