□특별기고□

해킹기법의 동향

해커스랩 김창범

1. 서 론

인터넷이 급격히 발전하면서 컴퓨터를 이용한 범죄가 크게 증가하고 있다. 특히 인터넷을 통해서 상대편의 컴퓨터에 침입하는 해킹은 시스템과 네트웍에 대한 전반적인 이해를 바탕으로 이루어진다. 그러나 근래에 들어서 단순히 인터넷에 공개된 프로그램이나 정보를 이용해 공격만을 위한 해킹을 한다던지, 불특정 다수에 대한 생각 없이 단순히 피해를 입히는 사려들이 빈번하게 발생하고 있다. 해킹의 본질인 시스템을 파악하고 그를 통해 시스템과 네트웍에 대한 더 많은 지식을 쌓기 위한 과정으로 이해되던 긍정적인 입장이 퇴색되고 있다는 생각이 든다. 여기에서 소개하는 해킹 기법은 단순히 다른 사람들이 만들 프로그램으로 인식되는 것 보다는 보다 시스템과 네트웍을 잘 이해하기 위한 방안으로 받아들여져야 한다.

해킹에 앞서서 대상이 되는 시스템들에 대한 정보수집은 공격의 첫 번째 단계로 공격대상 네트워크에 대한 정보를 파악하는 것이다. 주로 네트워크 구성, 시스템 운영체제의 종류 및 버전, 네트워크 장비의 종류, 그리고 WWW, FTP 등 공격대상 네트워크가 제공하는 서비스와 그 버전에 대한 정보를 수집한다. 정보수집 방법은 스캔 공격도구를 이용하는 것에서부터, 다양한 네트워크 서버가 제공하는 정보를 수집하는 방법에 이르기까지 상당히 다양하며, 침입차단시스템을 무력화 할 수 있는 방법도 존재한다.

공격 대상 네트워크에 어떤 시스템이 있는지를 파악하기 위하여 간단하게 "Ping"과 같은 간단한 점검 도구에서부터 DNS 서버 조회 및 각종 네트웍 스캐너 및 취약점 분석 도구들을 이용한다. 요즘에 공개로 나온 보안 도구들은 보안 도구이기도 하면서 다른 측면으로 공격에 이용될 수 있는 도구이기 때문에 보안 도구들에 대한 자체, 정기점검이 필수로 필요하다. 공개된 보안 도구들은 시스템 및 네트웍에 대한 점검을 수행하여 문제가 있는 서비스를 찾아내 주며, 망 전체의 많은 대수의 시스템에 대해서 동시적인 적용이 가능하게 되어 있다. 이를 통해서 얻을 수 있는 정보는 망에 있는 시스템의 종류와 그 시스템들이 가지고 외부에 서비스하고 있는 서비스 종류, 그리고 취약점이 존재하는 서비스와 그 문제점에 대한 설명들을 제공한다.

좀 더 세밀한 해킹을 위하여 해당 시스템의 OS 버전에 대한 정보수집을 한다. OS 버전을 탐지하는 기술은 "IP stack fingerprinting" 이라는 점을 이용한다. 시스템에 따라 IP stack의 구현이 조금씩 다르기 때문에 그 특성을 특정 패킷을 만들어 보내서 그 응답의 형태에 따라 시스템을 구별할 수 있다. 이런 것을 지원하는 대표적인 도구로서 queso, nmap 등이 공개되어 있고, 이를 이용해서 다른 여러가지 스캐닝이 가능하다.

네트웍의 구성은 hop count라는 네트웍 거리를 나타내는 IP패킷의 내용을 이용해서 한다. 일반적으로 라우터와 같은 장비나 게이트웨이를 지날 때 마다 이 hop count가 감소하게 되는 이 차이를 이용해서 네트웍에 새로운 뭔가가 있다는 것을 알 수 있다. 방화벽에 대한 규칙을 알아내는 방법도 존재하는데, 여러가지 출발지를 가지는 패킷을 보내봄으로써 방화벽이 어떤 패킷에

대하여 통과를 시키는지를 알 수 있다.

DNS, SNMP, Sendmail, NetBIOS 등 일반 네트워크 서버가 제공하는 정보를 수집하여 공격에 유용하게 사용할 수 있다. DNS의 경우 "zone tranfer" 또는 일반적인 query를 이용하여 등록된 호스트의 정보를 알 수 있으며, 잘못 설정된 SNMP를 통하여 네트워크의 구성 및 각종 네트워크 정보를 알려준다. 또한 라우터를 통하여 중요한 정보를 알아낼 수 있는 방법도 존재한다.

다음으로는 시스템 침입을 시도하는 단계로 정보 수집에서 수집한 정보를 바탕으로 가장 취약한 부분을 공격하게 된다. 일반적으로 버그가 있는 네트워크 서버를 공격하게 되는데 알려진 취약성을 이용하고 가장 많이 이용되는 것은 버퍼 오버플로우라는 방식이다. 그 외에 서버의 설정 오류를 이용하는 방법도 있으며, 파일만을 빼올 수 있는 취약점이 있으면 패스워드를 빼내와서 crack을 거쳐 패스워드를 해독한다.

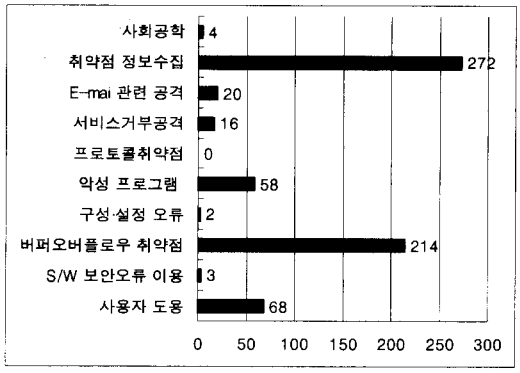


그림 1 1999년 국내에 사용된 해킹 기법

시스템 침입이 성공하고 나면 공격자는 시스템 침입흔적을 제거하게 된다. 또한 정보수집단계로 인하여 남은 흔적도 제거하게 된다. 또한 일반 계정으로 침입한 경우에는 충분한 권한(유닉스의 경우 root 권한)을 갖기 위하여 로컬 시스템의 취약점을 공격하게 되는데, 대부분의 시스템에서 이러한 취약점을 가지고 있다. 또한 제 침입을 위하여 비인가된 접근을 제공해주는 백도어나 트로이 목마를 설치하게 되는데 이러한 백도어는 데몬 서비스 형태, 또는 서비스의 비정상적인 설정 등을 이용하여 특정 포트를 열어놓게 된다.

공격자는 시스템 침입에 성공한 시스템을 이용하여 다른 시스템 공격을 위해서 스니퍼를 설치하여 네트워크상의 Telnet. POP, FTP 등에 대한 트래픽을 감시하며, 사용자 이름과 패스워드를 수집한다. 이외에도 침입한 시스템과 연관된 시스템, 데이터베이스 등에 접근을 시도하여 정보를 찾게 된다. 한번 침입 되어서 백도어나 트로이 목마가 설치된 경우 이를 이용하여 다른 곳을 공격하기 위한 거점으로 쓰이기도 한다. 보통 공격에 흔적이 남더라도 이런 거점을 여러군데 거쳐서 들어오는 경우가 많아서 역추적을 매우 힘들게 만든다.

2. 해킹 기법의 분류

해킹 기법에 대해서 직접 설명하기에 앞서 우선 해킹에 대해서 분류가 필요하다. 크게 보면 해킹 기법은 원격지에서 하는 네트웍 해킹과 시스템에 들어간 다음 시도를 하는 로컬 해킹이 있다. 네트워크에서 하는 해킹은 파일의 변경이나 파일의 획득 및 Unix와 같은 경우는 Shell의 획득을 위해서 시도된다. 로컬 해킹은 주로 시스템에 접근이 성공한 해커가 그 시스템에 root로 불리는 관리자의 권한을 획득하기 위해서 시도된다.

다음 그림은 시간에 따른 해킹 기법의 발전을 보여준다.

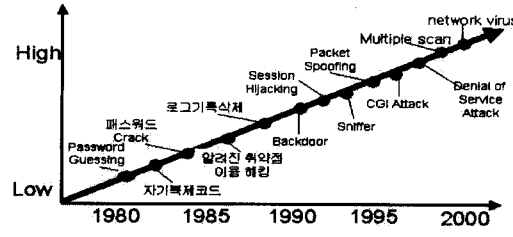


그림 2 크래킹 기술의 발전

해킹 사고는 공격자가 어떤 목적을 가지고 해킹을 시도함으로써 발생된다. 해킹이라는 것은 도구를 가지고 위험성을 가진 목표 대상에 대해 해킹 시도를 하는 것을 말하고 이 과정에서 발생하는 대상에 대한 해킹 시도는 사건으로 남겨지게 된다. 해킹 시도라는 것은 탐지, 스캐닝, Flooding, 인증, 통과, 속임, 정보의 읽기, 복사, 삭제, 수정 등의 행위를 하는 것이다. 다음 그림

은 해킹 사고에 대한 분류를 나타낸다.

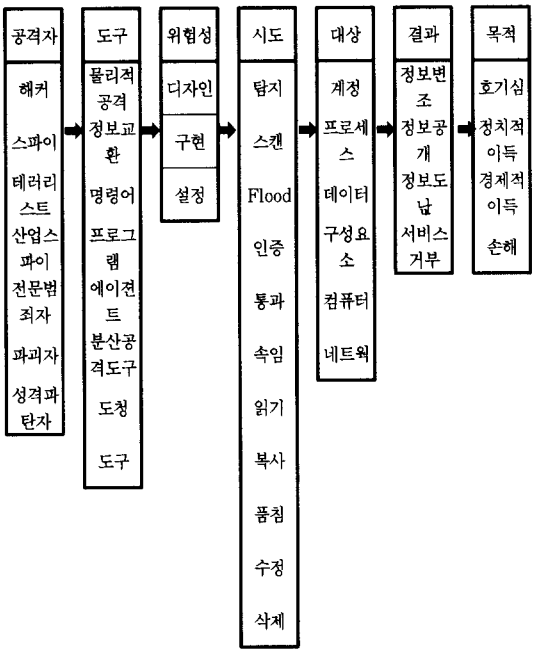


그림 3 해킹의 분류

해킹 시도의 측면에서 아주 많은 적용 가능한 방법들이 있는데, 여기에서는 해킹 기법의 원리와 위험성, 해결 방안에 대해서 설명하고자 한다.

3. 대표적인 해킹 기법들

3.1 DOS(Denial of Service)

올해 초 Yahoo, ebay 등의 해킹으로 아주 유명해진 공격 방법으로 간단하게는 한 개의 시스템에서 많은양의 패킷을 집중적으로 보내서 시스템을 마비시키는 것부터 시작해서 분산된 수백, 수천의 시스템에서 한 개의 시스템을 공격하는 분산 DOS공격까지 다양한 것이 있다. 실제로 이 해킹 방법은 시스템의 서비스를 중단시키거나 서비스의 방해를 목적으로 하는 것이지 해킹을 통한 정보나 자료의 획득, 시스템 자원의 획득 등의 고난도의 해킹 방법은 아니다. 위의 그림 4는 DDOS공격을 위한 시스템 구성도이다. 사용자는 Clinet에서 명령을 내려서 아래에 있는 Agent들이 최종적으로 수많은 패킷을 공격대상에 뿌리게

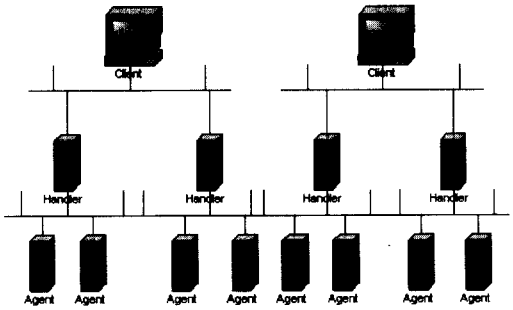


그림 4 DDOS

하여 대상 시스템이 제대로 된 서비스를 못하게 하고 시스템을 마비 시킨다.

3.2 취약성을 이용한 공격

소프트웨어를 개발하는 사람이라면 누구나 알고 있듯이 프로그램에는 버그라는 것이 있다. 버그는 간단한 것에서 시스템을 죽인다던지 하는 아주 심각한 것까지 다양하게 있는데, 재미있는 사실은 대부분의 사람들이 같은 부분에서 실수를 한다는 것이다. 취약성을 이용한 공격은 바로 이런 특징을 이용한 것이다. CERT 등의 많은 기관에서 매일 새로운 취약성이 발표하고 있다.

취약성에 대처하기 위해서 시스템 개발 회사라던지, 발표시에 취약성을 해결하기 위한 방안이 제시되고는 있지만 실제로 이를 적용하는 시스템들은 그렇게 많지 않기 때문에 대부분의 시스템들이 취약성을 가지고 있다고 할 수 있다. 만약 네트웍을 통한 접근을 허용하는 취약성이 있다면 이것은 바로 해킹을 시도하여 성공할 수 있는 확률을 아주 높여주는 것으로 아주 위험하다고 할 수 있다.

보안 소프트웨어도 이런 취약성이 존재하기 때문에 보안 소프트웨어로 시스템을 안전하게 보호하는 장치를 하였다고 자신하여도 시스템들이 지속적으로 해킹의 대상이 되고 해킹당해서 결과가 계속 나오는 이유도 바로 이런 취약성을 지속적으로 해결하기 위한 노력을 게을리하기 때문에 발생된다고 할 수 있다.

3.3 Sniffing

스니핑은 네크워크상의 한 호스트에서 그 주위를 지나 다니는 패킷들을 엿보는 것으로 다른 사

람의 계정과 패스워드를 알아내기 위해 자주 쓰이는 방법이다. 이 방법은 근래에 스위칭 허브가 보급되면서 역할도 많이 줄기는 하였지만 주요한 시스템을 해킹한 이후라면 많은 사람의 계정과 패스워드를 손쉽게 알아낼 수 있다.

그림 5 스니핑

스니핑 프로그램은 네트워크 디바이스를 열어서 Promiscuous mode로 만들어서 지나가는 모든 패킷을 읽는다. 실제로 대부분의 시스템들이 아직 암호화된 통신을 안하고 있기 때문에 스니핑에 의해 빼낼 수 있는 정보는 더 많다고 할 수 있다. 일단 암호화를 하는 통신을 하는 경우에는 스니핑에 의해 빼낸 정보를 복호화하지 못하는 이상 내용을 볼 수 없기 때문에 안전하다고 할 수 있다.

스니핑을 찾는 것은 대부분의 Unix 시스템에는 ifconfig란 명령어가 있는데 이를 통해서 스니핑을 하고 있는지 알 수 있다. 하지만 Solaris와 같은 시스템은 이 방법으로 확인할 수 없다. 이럴때는 lsof(ftp://vic.cc.purdue.edu/pub/tools/unix/lsof))란 도구를 깔아서 확인한다. Lsof는 원래 현재 사용되고 있는 파일의 목록을 보여주는 것인데, 스니핑을 할 때 파일로 결과를 저장하기 때문에 스니핑 여부를 알 수 있게 되는 것이다. 이외에 CPM(Check Promiscuous Mode: ftp://ftp.coast.purdue.edu/)란 프로그램이 있는데, 이를 구해다 컴파일해서 점검이 가능하다.

3.4 스푸핑

스푸핑은 속이는 방법을 통해서 해킹을 하는 것으로 이해를 쉽게 하기 위해 로그인 스푸핑에 대해서 설명하면 마치 로그인 화면 같은 프로그램을 통해서 사용자로 하여금 패스워드와 계정을 치게해서 패스워드를 알아내는 방식이다. 이런 형태의 스푸핑으로 대표적인 것이 Connection Hijacking과 IP Spoofing, Sequence Number Prediction 등 IP(Internet Protocol)레벨에서 속이는 방법을 이용한다. 예를 들어서 대상이 되는 호스트의 IP를 똑같이 주게 되면 Duplicate IP Address라는 에러가 발생하고 서버 시스템은 잠시 네트웍이 멈추게 되는데, 이 순간에 가짜 IP를 다시 한번 이용해 다른 시스템으로 하여금 자신이 대상이 되는 호스트로 보이게 하는 착각을 일으키게 한다. 이런 방식은 IP뿐만 아니라 DNS와 같은 곳에도 똑같이 적용된다.

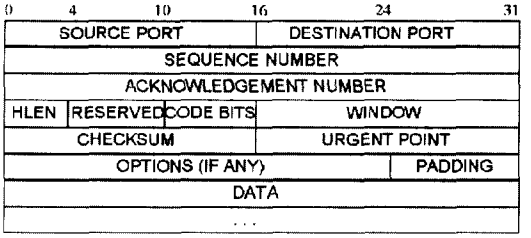


그림 6 TCP Packet의 구조

Connection Hijacking은 TCP 프로토콜 구조에 대해서 잘 알아야 가능하다. 위의 그림은 TCP의 패킷 구조를 나타낸다. Connection Hijacking은 TCP 연결에서 클라이언트와 서버 사이에 확인하는 것이 Sequence Number와 Acknowledgement Number만이기 때문에 가능하다.

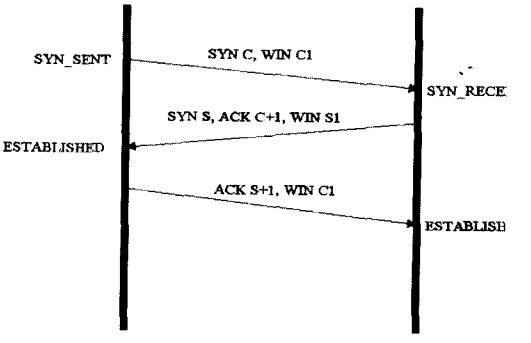


그림 7 일반적인 TCP 연결시작

일반적인 TCP의 연결은 위의 그림과 같이 3-phase handshake에 의해 이루어진다. Clinet는 서버에 SYN이란 code를 보내면 서버는 여기에 대한 SYN/ACK를 보낸다. 마지막으로 클라이언트는 Ack를 보냄으로서 연결이 형성된다. 이때 Sequence Number, Acknowledgment number는 규약에 의해서 서로간에 맞춰서 진행된다.

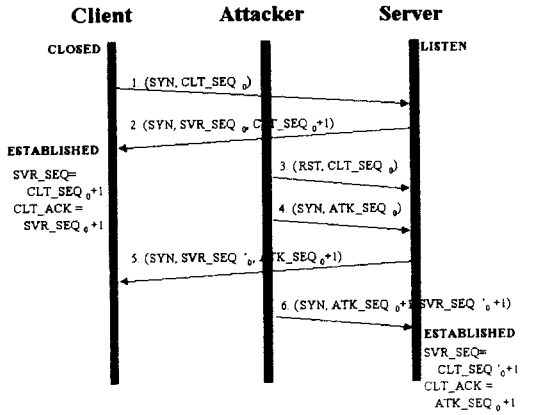


그림 8 Connection Hijacking

일반적인 연결이 이루어지는 것에 비해 Connection Hijacking은 공격자에 의해 전송되는 패킷을 모니터링할 수 있는 위치에서 가능하다. 위의 그림에서와 같이 일반적인 연결이 되는 도중에 공격자는 RST를 서버에 보내면서 다시 바로 SYN을 다시 보냄으로서 마치 연결이 자연스럽게 계속되는 것 같이 만든다. 그리고 서버가 새로운 sequence number를 생성해서 보내면 Client는 이미 연결이 되 있는 상태에 있기 때문에 이를 무시하게 되고 공격자는 다시 서버 Sequence Number에 맞추어 Ack를 보냄으로써 서버와 공격자 사이에 연결이 이루어진다.

3.5 버퍼 오버플로우

버퍼 오버플로우는 시스템에 실행되고 있는 프로그램에서 메모리 버퍼를 넘치게 해서 프로그램을 이상 동작하게 만드는 기법이다. 시스템에서 프로그램이 실행되어지고 있다는 의미는 그 프로세스가 이용하고 있는 메모리 영역이 존재한다는 뜻이며, 이 영역은 원칙적으로 보호되는 구역과 보호되지 않는 구역으로 구분된다. 문제는 이 보호되지 않는 영역이 존재하고 이 때문에 이 영역을 잘 활용하면 프로그램이 원래의 목적을 벗어난 이상 동작을 할 수 있다는 것이다.

메모리는 다음과 같은 구조로 구성되어 있다.

text는 프로그램의 본체라고 할 수 있는 명령어들과 Data들을 담고 있다. Data는 C 언어에서 사용되는 광역 변수, 정적 변수 등으로 선언되는 이 저장되어 있는 곳이다. Stack은 C가 함수를

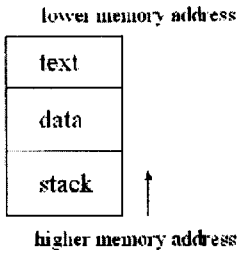


그림 9 메모리 구조

부를 때 사용하는 영역으로 프로그램의 수행 중 함수를 부를 경우 이 새로운 함수에서 사용할 지역변수, 인자값, 함수가 끝났을 경우 다시 되돌아갈 text 영역의 주소인 Return Address 등을 Push하여 스택에 저장하게 된다. 이 후 함수가 끝났을 경우 위의 값들을 Pop하고 Return Address로 돌아가게 된다.

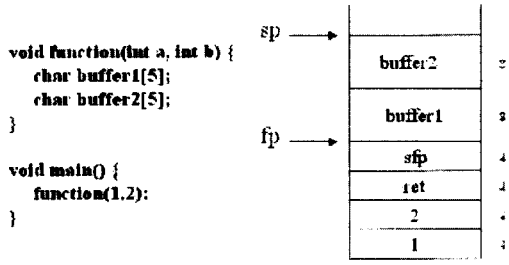


그림 10 스택의 실행 1

그림 10에서 보듯이 버퍼는 아래에서 위로 메모리가 쌓이는 구조로 되어 있고, 각 메모리의 포인터는 가장 위를 가리키고 있다. 위의 예에서 main함수에서 function을 call하면 이때 들어간 인자 1, 2가 스택에 저장되고 ret(Return Address)가 저장되고 sfp(stack frame pointer)가 저장되고 그리고 지역변수인 buffer1, buffer2가 저장되어 있다. Fp(frame pointer)는 실행중인 함수의 위치를 나타내고 sp(stack pointer)는 메모리의 끝을 나타낸다. 버퍼 오버플로우는 여기에서 buffer1이나 buffer2에 strcpy와 같은 경계를 명확하게 점검하지 않는 시스템 함수가 불리워질때 여기에 buffer1, buffer2의 크기보다 훨씬 큰 데이타를 집어넣어서 ret(return address)를 프로그램이 원하는 위치로 변경하는 것이다.

그리고 그 위치에 shell을 실행하는 코드를 집어 놓음으로서 프로그램의 실행과 함께 shell을 얻을 수 있게 된다.

버퍼 오버플로우를 막기 위해서는 프로그램 개발시에 경계를 점검하는 자체 함수나 상용함수를 쓰거나 stack guard라는 컴파일러로 컴파일하는 것이다. 하지만 stack guard라는 컴파일러는 아직 성능이 않좋다는 이유로 많은 이용이 없다는 것이 현재로서는 문제점이다. 버퍼 오버플로우에 이용되는 프로그램이 Setuid가 설정되어 있는 파일이면 이 공격을 통해서 root의 권한으로 도는 shell을 얻을 수 있다.

3.6 백도어와 트로이 목마

트로이 목마는 정상적인 기능을 하는 프로그램으로 가장하여 프로그램 내에 숨어서 의도하지 않은 기능을 수행하는 프로그램의 코드 조각이다. 이는 바이러스나 웜에서 주로 사용하는 메커니즘이다. 최근의 컴퓨터 세계에서 말하는 트로이 목마 는 유용하고 흥미있는 기능을 수행하거나 그런 기능을 하는 것처럼 보이는 독립된 프로그램에 숨어 있다. 이러한 표면적으로 드러나는 기능과 함께 트로이 목마 프로그램은 어떤 다른 비인가된 기능을 수행한다. 전형적인 트로이 목마는 유용한 것으로 가장하여 사용자가 그 프로그램을 실행하도록 속인다. 사용자가 의심하지 않고 그 프로그램을 실행하게 되면 실제 기대했던 기능이 수행된다. 하지만 실제 목적은 사용자의 합법적인 권한을 사용하여 시스템의 방어 체제를 침해하고 공격자는 접근이 허락되지 않는 정보를 획득하는 것이다. 트로이 목마는 새로운 시스템 기능에 대한 정보를 보여주거나 새로운 게임이라고 하는 프로그램들에 숨어 있는 경우가 많다.

전통적인 트로이 목마 공격은 Dennis M. Ritchie에 의해 소개되었다.

공격자는 정상적인 login 기능을 하는 것 같은 패스워드 수집기(Password grabber)프로그램을 작성한다. 의심 없는 사용자가 로그인 프롬프트(login :)를 보면 로그인하려고 하고, 프로그램은 정상적인 로그인 순서로 사용자가 평범한 방법으로 로그인하고 있다고 생각하게 한다. 하지만 Trojan Horse를 가진 그 프로그램은 로그인 ID와 패스워드를 받으면 이 정보를 공격자 소유의 파일에 복사하거나 메일로 공격자에게 전송한다. 그리고 "login incorrect"라는 오류 메시지를 보낸다. 사용자는 자신이 잘못 쳤다고 생각하고 로그인 ID와 패스워드를 다시 친다. 그 동안 Trojan Horse를 가진 프로그램은 빠져 나오고 실제 login 프로그램에게 제어권을 넘겨준다. 다음에 사용자는 성공적으로 로그인하게 되고 자신의 로그인 ID와 패스워드 정보가 유출되었다는 사실을 전혀 의심치 않는다.

어떤 이는 이러한 종류의 트로이 목마를 "Trojan mule"이라고 부르는데 그 이유는 전형적인 트로이 목마처럼 유용한 시스템 정보를 보여주는 프로그램이나 게임 프로그램에 숨어 있는 것과는 다르게, 유용한 기능을 가장하지 않고 단순한 눈속임이기 때문이다. 어떤 트로이 목마는 자기 존재의 흔적을 남기지 않아 발견될 염려가 없고, 의심받지 않는 소프트웨어에 숨어 있다. 또 발견되기 전에 스스로를 파괴하도록 프로그래밍될 수도 있다.

Back Door는 원래 시스템 설계자나 관리자에 의해 고의로 남겨진 시스템의 보안 헛점으로 응용 프로그램이나 운영체제에 삽입된 프로그램 코드이다. 즉 Back Door는 시스템 접근에 대한 사용자 인증 등 정상적인 절차를 거치지 않고 응용 프로그램 또는 시스템에 접근할 수 있도록 한다. 이러한 보안 헛점을 남겨두는 이유가 항상 악의적인 것은 아니다. 경우에 따라서는 현장 서비스 기술자나 시스템 공급자의 유지보수 프로그래머가 사용할 목적으로 특수 계정을 허용하는 코드를 운영체제나 응용프로그램에 넣을 수도 있다. 이러한 Back Door는 디버깅 시 개발자에게 인증 및 셋업시간 등을 단축하기 위한 뒷문으로 사용된다. 하지만 이러한 Back Door가 비양심적인 프로그래머가 비인가된 접근을 시도하거나 개발이 완료된 후 삭제되지 않은 Back Door가 다른 사용자에 의해 발견될 경우 대단히 위험할 수도 있다.

1983년 Ken Thompson이 ACM에서의 강연에서 초기의 유닉스 버전에는 Back Door가 존재하고 뛰어난 해커라면 이를 공격할 수도 있다고 말했다. 이러한 스킴으로 'login' 프로그램이 재 컴파일되어 특정 패스워드가 입력되었을 경우 접근을 허락하도록 하는 코드가 숨겨져 있을 수

도 있다. 그 결과 시스템에 계정이 있든지 없든지 간에 시스템으로의 접근이 허용될 수 있다.

컴퓨터 시스템에 침입하려는 공격자들은 시스템에 비정상적인 방법으로 시스템에 접근하고자 Back Door 기술을 개발하였다. 침입을 위한 Back Door 프로그램들의 주요 특징은 다음과 같다.

* 모든 패스워드들을 바꾸는 등 관리자가 안전하게 관리하려고 함에도 불구하고 시스템에 침입할 수 있다.
* 발견되지 않고 시스템에 침입할 수 있다. 대부분의 Back Door 프로그램은 로그를 남기지 않고, 온라인 으로 들어 왔음에도 불구하고 이를 발견할 수 없다.
* 시스템에 최단 시간에 침입할 수 있다.

3.7 Race Condition

Race Condition이란 것은 어떤 프로그램이 임시파일을 만드는 특성을 이용하는 것이다. 예를 들어서 /tmp 디렉토리에 어떤 지정된 파일을 만드는 경우 자신이 원하는 파일로 미리 이 파일을 Symbolic link를 만들어 두면 이 파일이 실행되면서 임시파일을 만들면서 이 Symbolic link된 파일을 변경하게 된다. 보통 root로 실행되는 프로그램이 이런 행동을 하게 되면 해커가 만들어 둔 파일의 권한이 해커가 변경하거나 지울 수 있는 파일로 될 수 있기 때문에 이를 이용해서 해킹이 가능하다. 이 방식은 직접 권한을 root로 주어지지는 않지만 나름대로 해커가 root권한의 파일을 마음대로 운영할 수 있는 권한을 줄 수 있다는 특징이 있다.

3.8 Windows 해킹

Windows 시스템도 여러가지 해킹 방법이 존재한다. 스니핑, 스푸핑을 비롯하여 Netbios를 이용한 해킹, 공유, IIS 버그로 인해 위험그리고 다양한 도구를 이용한 로칼, 원격 트로이 목마들이 나와있다. 특히 사용자들이 유의하여야 하는 부분은 공유설정에 대한 부분이다. 일반적으로 컴퓨터를 처음 사용한 사용자가 공유를 하게 되면 패스워드에 대한 설정등을 하지 않는 경우가 많은데 이것은 굉장히 위험한다. 인터넷에 연결된 모든 기계의 공유에 이렇게 패스워드가 걸려있지 않으면 어디에서도 접근이 가능하다. 또한 공유에 대한 사용자의 설정이 중요하다. NT의 경우 특히 공유 사용자가 Everybody로 기본 설정되는데 반드시 유효한 사용자를 만들고 그 사용자만이 접근하도록 하는 것이 중요하다.

Windows를 가지고 서비스를 하고 있으면 반드시 최신의 서비스 팩을 깔도록 하고 가장 최신의 소프트웨어를 쓰도록 하는 것이 중요하다. Unix와 마찬가지로 네트웍 서비스 쪽에서 Windows도 많은 버그로 인한 위험성이 존재하고 이를 지속적으로 보고 문제를 해결할 수 있어야 한다.

3.9 Back Orifice

백 오리피스 클라이언트/서버 모델로 설계되었으며 원격 공격자 클라이언트는 해당 Back Orifice 백도어 서버를 공격대상 시스템에 설치해야 하고 공격자 호스트에 클라이언트 프로그램을 설치해야 한다. 설치는 자동적으로 이루어지며 초보자도 쉽게 설치 가능하도록 되어있다.

Back Orifice 백도어 서버는 자체적으로 다음과 같은 주요기능을 가진다.

* HTTP 서버의 기능
* 패킷 스니퍼링 기능
* 키보드 모니터링 기능
* 연결재지정(connection redirect) 기능
* 시스템 명령어 및 네트웍 프로그램 실행기능
* 파일 공유 설정 변경
* 프로세스 상태 변경(list, kill, start new)
* 레지스트리 변경 기능
* 서버/클라이언트간에 암호화 통신 기능

클라이언트측에서는 서버의 이러한 기능을 이용하여 파일시스템의 모든 파일들에 대하여 접근이 가능하고, 프로세스의 생성/삭제도 원격 조정된다. 그리고 시스템 패스워드 유출, 키보드 모니터링, 사용자의 현재 화면 캡쳐도 가능하고 네트워크자원의 공유지정, 네트워크접속 재지정, 파일조작, 레지스트리 조작도 가능하다. 이외에도 여러 가지 기능을 이용하여 원격사용자는 마치 자신의 시스템처럼 사용할 수 있다. 클라이언트는 유닉스용과 일반 윈도우용이 있으며 명령형 실행

방식과 GUI를 이용한 실행방식을 모두 제공한다.

또한 서버가 새로 버전업 되었을 경우 원격지에서 업로드하여 업그래이드 될 수 있는 재설치 구조를 가지고 있다. 그리고 플러그인 구조를 채택하여 세계각지에서 만들어진 플러그인을 추가시켜 수행시킬 수 있는 기능도 포함하고 있다.

백 오리시스는 바이러스와 같이 메일에 붙어서 전파되거나, 인터넷에서 다운로드 받은 파일에 덧붙여 올수도 있고, 다른 사람에 의해 고의적으로 설치될 수 있다. 설치되있지 여부를 알기 위해서는 다음과 같은 방법들이 쓰인다.

* 백 오리피스는 기본적으로 31337 포트를 사용하는데 이포트가 열려있는지 netstat -a를 통해서 알아본다. 실제로 이 포트는 공격자에 의해 바뀔 수 있기 때문에 유의하여야 한다.

* HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices\Default의 값을 regedit 명령으로 확인하여 Blank가 아니고 특정값이 들어있으면 백도어가 있다고 판단하여 해당 값을 변경시킨 후 시스템을 재부팅 시킨다.

* Bodetect 등 여러가지 detection하는 도구를 이용한다.

백 오리피스를 제거하기 위해서는 가장 좋은 방법이 하드를 포맷하고 시스템을 새로 깔고 V3나 nobo와 같은 프로그램을 설치하는 것이고 아니면 nobo나 bospy와 같은 프로그램으로 제거하도록 해야한다.

3.10 기타 해킹 방법

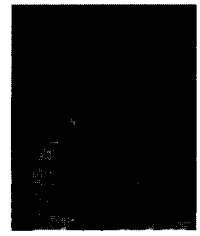
해킹 방법은 날로 새로운 방식이 등장하고 있고, 시스템의 버그로 위험한 취약성은 매일 매일 새로운 것들이 발표되고 있다. 대표적으로 작년에 처음 발견되었지만 아직 분석이 제대로 이루어지지 않고 있는 Ring 0라는 해킹 방법은 Windows 시스템에 ITS.exe와 PTS.exe란 프로그램을 깔고 내부에서 외부로 연결을 시도한다. 이런 특징은 방화벽을 완전하게 무력하게 하는 특징들이다. 이 해킹 방법은 포트 스캔 및 자신이 얻은 정보의 전송을 한다는 특징이 있다는 것이 기존의 해킹 방법과는 아주 다른 방식이라고 할 수 있다. 이 방식이 전송되는 방식은 현재 바이러스와 같은 형태라고 분석되고 있지만 아직까지 어떤 증거도 찾지 못한 상태에 있다.

이외에도 Phreaking이라는 해킹이 있다. 이것은 컴퓨터나 네트웍을 기반으로 한다기 보다는 Real World에 있는 전화라던지, 위성, 이동 전화, 열쇠등 다양한 것을 대상으로 하고 있고 심지어는 커피 자판기까지 대상으로 한다. 이것의 출발은 2600Hz라는 주파수가 전화기 교환기를 교란시켜 공짜로 전화를 걸 수 있게 한다는 발견으로부터 시작되었다. 근래에 들어서는 위성까지 대상으로 하려는 노력들이 있다.

4. 결 론

본 지면에서는 각각의 해킹 기법들에 대해서 대략적인 설명을 하였다. 각각의 방법에 대해서 구체적으로 이해하기 위해서는 인터넷에 공개된 문서들을 살펴봐야 한다. 또한 해킹 방법을 배웠다고 해서 다른 사람의 서버에 테스트를 하는 것은 불법이기 때문에 자신의 서버에 환경을 갖추고 실험을 하거나, www.hackerslab.org 같은 해킹을 연습할 수 있는 사이트에서 실험을 하여야 한다.

앞서 설명한 해킹 기법들은 이미 발표된 것들이다. 그러므로 충분한 주의를 기울인다면 이러한 방법을 사용한 해킹에 대해서 대처할 수 있다. 그리고 실제적으로 이런 정도의 해킹을 막을 수 있다면 자신의 시스템을 안전하게 지키고 있다고 할 수 있다. 실제로 정보보안 사고가 발생하는 것을 보면 대부분 앞서 설명한 방법들을 사용한 것이다. 본 지면에서 해킹 기술들을 소개하는 것도 해킹 기법들을 배워서 자신의 시스템을 보호하는데 활용하도록 하기 위함이다.



김 창 범

1990 한국과학기술원 전산학과 학사
1993 한국과학기술원 전산학과 석사
1996 미국 PSI Research staff
2000 해커스랩 부사장
2001 한국과학기술원 전산학과 박사예정
E-mail:cbkim@hackerslab.com