「經營情報硏究」 第4號
大韓經營情報學會
2000年 6月

네트워크 2중 보안을 위한 전문가시스템 시뮬레이션의 구현

이 창 조*

〈목 차〉

Ⅰ. 서 론

인터넷이 추구하는 정보의 공유성은 사업 또는 연구분야에서 경비 절감뿐만 아니라 뛰어난 선전 매체 작업의 효율성 등을 제공할 수 있다. 다른 기술이나 그룹 활동처럼 인터넷은 놀라운 능력과 함께 위험부담을 함께 가져온다. 그러나 위협을 이해하고 데이터를 지키는 기술적인 행동을 취한다면 인터넷에 관련된 보안 허점의 위험을 최소화할 수 있다. 네트워크 보안에서 납득할 정도로 안전한 유일한 컴퓨터는 네트워크에 연결되지 않은 컴퓨터라는 점이다. 또한 네트워크상에는 사용자의 컴퓨터에 침입해서 정보를 몰래 훔쳐 가는 해커들이 있다. 인터넷이라는 사이버 공간을 활용한 전 세계인의 무대라는 장점을 이용하여 컴퓨터에 초대한 사람마저도 신뢰할 수 없는 현실에 부딪쳐 있다. 네트워크 보안을 열기를 좋아하는 사람, 특별한 정보를 훔

* 성심외국어대학 정보통신학부 경영정보시스템전공 조교수

쳐 가는 사람, 데이터를 파괴하는 사람 등을 소위 '해커'로 부른다. 해커는 단순한 취미로 여러 사이트를 크랙하는 사람일 수도 있고 경제적인 이유로 사이트를 크랙하려는 자기 경쟁자의 고용인일 수도 있다.

특히 인터넷에 연결된 시스템은 항상 보안의 위협에 노출되어 있다. 사용자는 공격으로부터 시스템을 보호하려면 침입자가 데이터 약점, 소프트웨어 약점, 물리적-시스템 약점, 전송 약점 등과 같은 경로를 통해서 공격할 것임을 예상해야 한다.

인터넷 해킹사고로 접수되는 대부분의 사건들은 바로 Intrusion의 형태를 띤다. 즉, 불법으로 시스템 자원을 사용한다든지 다른 해킹을 위해 거쳐가는 경로를 사용함으로써 해커들은 다른 시스템을 침입하는 것이다. 인터넷상에서 특정 호스트 또는 네트워크로 하여금 제역할을 수행하지 못하도록 각종 서비스들을 정지시키는 형태의 해킹방법을 바로 Denial of Service라고 하며 실제로 어느 회사의 공식 홈페이지가 이 공격에 의해 웹서비스를 지원하지 못한 일이 발생하기도 하였다. 정보의 가치를 경제적인 측면에서 평가되는 시대가 도래함에 따라 해커들의 정보를 대하던 시각은 급격히 바뀌어 가게 되었다. 즉, 훔쳐온 정보가 곧 돈이라는 의식아래 라이벌 기업의 정보를 훔쳐낸다든지 훔쳐낸 기밀로 흥정을 하는 등의 새로운 해킹유형이 생겨났다. 해커들이 특정 시스템을 타깃으로 잡은 뒤 어떠한 기술적인 것들을 이용하여 해킹을 시도하는 방법을 분류해보면 아래와 같은 종류로 나눌 수 있다.

1. Local attack

UNIX 시스템은 멀티유저/멀티프로세싱 환경을 지원하는 일환으로 사용자들에게 쉘이란 도구를 지원한다. 즉, 사용자들은 쉘과 커널과의 통신을 통해 원하는 프로그래밍이나 기타 응용 프로그램들을 사용할 수 있는 것이다. 하지만 이 쉘을 이용해 시스템내의 각종 버그나 홀을 이용해 root 권한을 도용할 수 있게되며 이를 가리켜 소위 Local attack이라 한다.

이런 유형으로는 race condition, IFS 쉘변수를 이용한 해킹, Buffer Overflow 공격 등이 존재한다.

2. Remote attack

타깃시스템에 계정을 갖고 있지 않을 경우에도 해커들은 타깃시스템의 정보를 빼 내려고 할 것이다. 즉 계정으로 접근할 수 없는 시스템에 접근하기 위해 가능한 모든 방법을 이용하는 해킹유형을 가리켜 Remote attack이라 한다.

3. Denial of Service

타깃시스템이 지원하는 서비스 자체를 망가뜨리거나 타깃시스템 자체를 다운시키는 등의 일이 주요 목표가 되는 경우도 있다. 네트워크 보안의 주된 관점은 외부 사용자로부터 내부 시스템을 보호하는 것으로 인터넷을 사용하는 경우 외부 네트워크에서 내부 접근을 제어하여 내부 네트워크상의 시스템을 보호하는 것이다.

Inter-Networking 환경하에서 보안은 네트워크 보안, 시스템 보안 등으로 분류된다.

3.1 네트워크 보안(Firewall)

초기에 FireWall은 네트워크 보안의 개념으로 도입되었는데 최근 FireWall의 발전 추세는 시스템보안, 자료보안의 기능도 일부를 포함하여 운영되고 있다. 네트워크 보안의 일반적인 방식은 다음과 같다.

1) Packet Filtering 방식

Packet Filtering 방식은 OSI 7 Layer상의 Network Layer와 Transport Layer 사이의 Packet의 정보를 검증하는 방식으로 IP-address와 전송Protocol(TCP, UDP)을 기준으로 내부 네트워크에 대한 접근을 제어한다. 주로 Router에서 많이 사용되며 단점으로는 log, audit, authentication 기능이 제공되지 않고 access 제어를 위한 Rule(access list)을 많이 만들면 시스템 performance가 현저히 감소되어 UDP와 같은 connectionless traffic에 대한 완벽한 검증이 되지 않는 경우가 발생한다. 몇몇의 인터넷 프로토콜 라우트는 패킷을 필터링 할 능력을 가지고 있다. 패킷 필터링은 트래픽이 소스 IP address, Destination IP address, 프로토콜, TCP나 UDP 등과 같은 몇몇 영역에 기초한 사용자의 네트워크에서나 네트워크로부터 통과할 수 있는지 없는지를 사용자가 결정하기 위해 라우터를 프로그램 하도록 허락한다.

2) Application Level Gateway방식

Application Level Gateway방식은 7 Layer의 Application Layer에 해당되는 data 영역의 Packet만을 검증하는 방식을 지원한다. 즉 client에서 service Request(Telnet, Http, Ftp)가 들어오면 firewall에서 Application의 검증(Telnet Gateway, FtpGateway 등의 검증을 통함)을 거쳐 목적 시스템으로 접속되는 방식이다. Application gateway의 장점으로는 내부와 외부 Network이 완전 분리되어 High Level의 보안수준을 제공하며 log, authentication, Audit 기능 등을 지원한다. 단점으로는 Performance가 좋지 않으며 네트워

크 Traffic이 많은 경우 Firewall로서 정상적인 기능을 지원하지 못하는 경우가 발생할 수 있다.

3) Firewall S/W의 종류

① TCP Wrapper

접근 제어 리스트(Access Control List)를 통하여 관리자가 네트워크 접근을 필터링하고 기록하는 유닉스 기반의 방화벽 툴이며 TCP_WRAPPER툴은 네트워크 서비스의 통제와 모니터링을 제공한다. 기본적으로 무슨 일이 일어나느냐 하는 것은 서비스가 접속될 때마다 TCP_WRAPPER소프트웨어를 구동하기 위해 사용자의 dual-homed 게이트웨이에 사용자가 어떻게 구성했느냐 하는 것이다. 사용자가 TCP_WRAPPER를 어떻게 구성했느냐 하는 것은 그때의 접근에 대한 로그정보가 되며 실제로 의도된 서버 프로그램이 시작된다. 사용자는 소스를 가지고 있고 사용자가 무엇을 필요로 하느냐에 따라 더 좋게 수정할 수 있다. 예를 들어 사용자는 실제프로그램(actual program) 대신에 프록시 서버에 접속하기 위해서 TCP_WRAPPER를 원할 지 모른다. 이때 사용자의 보안요구가 요청하는 어떤 방법이든 간에 이 처리를 사용자의 프록시 소프트웨어가 다루도록 하여야 한다. 이것은 몇몇 소스들로부터 유용하다. 그러나 CERT가 입증했던 가장 최신의 카피를 가져야 한다는 것을 유념해야 한다.

② TIS Toolkit : 인터넷 방화벽 공개 소프트웨어이다.

③ TCPR : 방화벽을 통하여 나가는 Telnet과 FTP에 투명성 있는 프록시 기능을 제공하는 perl로 작성된 툴이다.

④ SunScreen : Sun Microsystems사의 Application level의 방화벽 제품이다.

3.2 시스템 보안

① COPS : UNIX 시스템의 보안상 문제점을 검사하는 툴이다.

② TIGER : COPS와 비슷한 툴로 사용이 더 편리하고 파일 변조 유무를 검사하는 기능도 있다.

③ NPASSWD : 추측 가능한 패스워드를 검사하는 패스워드 변환 툴이다.

④ PASSWD+ : 패스워드 변환 툴이다.

Ⅱ. 네트워크 2중 보안의 구현방법

본 연구는 네트워크 2중 보안 툴을 구현해 보고자 파워빌더를 이용하여 Simulation해 보았다. 또한 프로토타입의 실행과 그 시뮬레이션의 결과들을 구현해 봄으로써 네트워크상에서도 확장, 사용 가능하게 하여 이미 존재했던 보안의 개념에서 각각의 방화벽 계층을 두껍게 하여 철저한 개인 정보 보안을 유지해 보고자 함에 그 목적을 두었다.

위의 각 요소들에 대한 언급을 시작하기 전에 먼저 보안정책을 개발하기 위해 필요한 부분들을 설명해야 할 것 같다. 즉 각 조직이나 사이트들은 효과적인 보안정책을 가지고 있을 필요성이 있다. 여기에는 많은 도구들과 보안 통제를 수행 가능하게 하는 기술들이 있다. 그러나 사용자는 먼저 필요한 것이 무엇인가에 대한 면밀한 분석을 수행해야만 하고 네트워크 서비스나 특징들에 대한 필요사항이 무엇인지 결정해야 한다. 또한 사용자가 요구하는 보안의 등급 정도와 사용자가 받아들일 수 있는 위험이 무엇인지 결정해야 한다. 그러나 이런 결정은 각 조직에 따라서 다르고 보안을 확장시키는 다양한 방법들뿐만 아니라 사용자의 지역 네트워크와 외부간의 보안 억제 포인트를 생성하는데 대한 방화벽의 개념은 모든 사람들에게는 적합하지 않을 수 있다.

이런 툴에 대한 어떤 기능은 중복되어 있다. 사용자는 이런 툴에 대한 소스를 가지고 있기 때문에 새로운 특성을 첨가하기 위해서 최적화 하거나 이들을 수정할 수 있다. 네트워크에 대한 접근방식은 다음과 같이 나누어 볼 수 있다.

1. 네트워크상의 호스트에 대한 물리적 접근

1.1 SOCKS Library and sockd

SOCKS Library and sockd 는 "TCP wrapper"를 수행하기 위한 또 다른 방법을 제공한다. 이것은 보안상에서 시스템이 이것을 작동하게 하도록 의도된 것은 아니다. 그러나 모든 외부 인터넷 서비스를 방화벽으로 집중시키도록 되어있다. 이 sockd처리는 서버상에서 요청될 때마다 inetd에 의해 시작되어 진다. 그리고 이때 인증된 호스트에게만 접속이 허용된다. 또한 접속에 대한 LOG 정보가 될 것이고 사용자는 외부로 나가는 접속에 대해 sockd을 직접적으로 이용하기 위해서 클라이언트 소프트웨어를 수정하는 용도로 Socks Library를 사용할 수 있다. 물론 사용자는 이런 클라이언트 프로그램의 소스를 가지고 있어야 한다.

1.2 공유 라이브러리를 통한 SunOS, RPC를 위한 Kernel_Wrap

본래 이것은 portmap, ypserv, ypbind, ypupdated, mountd, pwdauthd 등과 같은 RPC를 사용하는 SunOS 데몬을 위한 wrapper이다. 이것을 이용하기 위해서 사용자는 SunOS 4.1 이상을 가지고 있어야 하며 사용자의 공유 라이브러리를 개조할 능력을 가지고 있어야 한다. 기본적으로 일어나는 것은 사용자 accept(), recvfrom(), 그리고 recmsg()와 같은 RPC 접속을 만들기 위해서 커널이 사용하는 함수호출을 수정하는 것이다. 이런 호출은 공유 라이브러리에서 유지되기 때문에 사용자는 커널을 재작성하지 않고 그것들을 수정할 액세스를 가지게 된다.

1.3 SWATCH

간단한 SWATCH는 실제로 두 가지다. 그것은 다양한 보안 프로그램에 의해 생성되는 "sysolg" 같은 LOG 데이터를 해석하기 위해서 사용되는 하나의 프로그램이다. 그러나 이것은 trigger 형태로 구성되어 있으며 실시간으로 LOG를 모니터링하고 있다. 사용자가 관심을 보이는 우선 순위의 이벤트에 기초해서 동작할 수 있다. 이것을 전부 사용하기 위해서 사용자는 ftpd나 telnetd와 같은 사용자의 네트워크 서비스 데몬을 수정할 필요성이 있고 확장된 접속(logging)이 SWATCH를 만족시키기 위해 "syslog"에 추가되어진다.

1.4 Controlled Access Point(CAP)

이것은 프로토콜 정의나 방법적인 것이다. CAP는 위험을 줄이기 위해 계획된 네트워크 메커니즘을 제공한다. 패스워드 추측이나 초기 패스워드를 가진 잘 알려진 계정에 대한 조사, 인증된 호스트 rlogin, 그리고 네트워크 스누핑(snooping)에 의한 패스워드 포착 같은 것이다. 이것은 두 개나 그 이상의 네트워크 접속을 위한 일반적인 방화벽 접근에 대한 변화나 강화를 위해 설계되어졌다. 이것을 설명하는 문서에서 두 로컬 net의 예가 있다. 하나의 소스는 인증 서비스를 구축하고 또 다른 것은 보증되지 않은 세그먼트이다. 둘다 CAP를 통해 서로 통신하고 여기에는 CAP의 비인증 면에 접속된 공공 네트워크에 대한 통신을 위해 라우터가 존재한다. CAP는 본래 들어오는 접근요구를 탐지하고 인증서버를 불러내며 사용자 인증처리를 가로채는 등 추가적인 기능을 가진 라우터이다.

1.5 Mail Gateway

이것은 소프트웨어 패키지보다 외부 접속을 요구하는 모든 네트워크 서버에 적용되어야 한다. 가장 간단한 실행상태에서 사용자는 패킷들을 필드하기 위해서 사용자의 라우터를 구성해

야 한다. 그래서 모든 메일 트래픽(SMTP프로토콜)이 "메일 게이트웨이," 하나의 호스트에 허용되어야 한다. 마찬가지로 사용자의 DNS나 MTA 소프트웨어는 이것을 위해서 잘 구성되어 있을 필요성이 있다.

2. 네트워크상의 호스트에 대한 논리적 접근

2.1 Computer Oracle and Password System(COPS)

COPS는 Unix 보안 상황을 체크하는 것이다. 이것이 하는 일은 파일이 손상되었다면 그것을 보기 위해 소프트웨어 구성이나 다양한 파일을 체크하는 것이다. 그리고 파일이 사용자의 보안 레벨을 최적화하기 위한 적합한 모드나 접근허용 설정을 가졌는지를 보기 위해 체크한다. COPS는 소프트웨어에서 버그를 검출하지 않으며 보안상 문제를 일으킨다. 그리고 발견되는 어떤 에러를 정정하지도 않는다.

2.2 Chkacct

Chkacct는 일반사용자를 위한 COPS이다. 이 툴은 사용자가 작동하기에 유용하도록 만들어졌으며 하루에 한 번 그들을 실행한다. 이것은 그들 자신의 계정에 있는 파일의 상황에 대한 완전한 거사를 수행한다. 그런 다음 그들에게 그 결과를 메일로 보낸다. 이 패키지는 사용자들이 빈틈없는 보안체제를 할 수 있도록 도와주며 프로그램에서의 참여 수준을 향상시킨다.

2.3 CRACK

CRACK는 보안관리자가 암호화하거나 많은 약한 부분을 검사함으로써 약한 패스워드를 확인하도록 도와준다. 만약 CRACK이 사용자의 패스워드를 알 수 있다면 사용자의 패스워드는 더 나은 것으로 바꾸어야 한다. 이것은 유추할 수 있는 침입자가 사용자의 패스워드를 획득할 수 있다는 가정과 같다.

2.4 SHADOW

프로그램의 쉐도우 패스워드는 /etc/passwd파일로부터 암호화된 패스워드를 제거하기 위해서 사용자의 시스템상의 일반적인 패스워드 제어 메커니즘에 해당한다. 그리고 한 장소에 숨겨져 있는 파일을 단지 읽을 수만 있는 권한을 가진다. 그것은 옵션으로 설정 컴포넌트로 구성되며 사용자가 패스워드를 바꿀 수 있도록 다시 패스워드를 제공한다. 또한 강화된 syslog logging을 추가하고 사용자가 60자까지 패스워드를 설정하도록 허용한다.

2.5 PASSWD+

PASSWD+는 사용자의 시스템 /bin/passwd에 위치한 사전실행 패스워드 검사기이다. 이것은 규칙 기반(rule-based) 시스템이고 쉽게 설정할 수도 있다. 또한 사용자들이 빈약한 패스워드를 선택하는 것을 막아 "CRACK"과 같은 프로그램이 추측할 수 없게 한다. 그리고 강화된 syslog, logging을 제공한다.

2.6 AUDIT

AUDIT는 다른 기종의 환경에 대한 정책위주의 보안 검사기이다. 이것은 사용자의 사이트 보안 정책에 정확히 맞게 설정할 수 있도록 충분히 구성할 수 있다. 이 프로그램은 기능적으로 COPS가 하도록 의도된 것들을 수행한다. 그러나 COPS가 하는 방법을 위한 사용자의 정책 결정용 hard-code는 아니다. 서버 시스템을 감시하는 대부분은 IMHP 벤더를 선택하고 이 정보를 사용하는데 대한 안내가 없는 다량의 원래 데이터를 그대로 두어야 한다는 것이다.

2.7 MIRO

miro는 COPS나 AUDIT와 같은 보안을 체크하거나 확인하기 위한 툴이다. 이것은 특정 OS에 묶여있지 않은 일반적인 형태이다. 이것은 보안 관리자가 형식적인 언어를 통해서 사이트 정책을 표현하기 때문에 유연성이 매우 높다. 또한 현재 정책의 명세를 변경하거나 쉽게 증가시킴으로써 정책을 수정하거나 확장하기가 쉽다

3. 추가적인 보안

위에 설명된 툴들은 일반적인 보안관리에 있어 기본적인 툴이며 기능적인 요구사항이라고 생각하는 부분들이다. 여기에서 설명되는 툴과 방법은 사용자의 전체 보안 프로그램에 추가될 수도 있고 응용하여 합해질 수도 있다.

3.1 One-time password

재사용 될 수 있는 패스워드가 침입자에 의해 확보되거나 도난 되어지기 때문에 일반적인 방법으로는 "one-time password"가 있다. one-time password는 소프트웨어만의 솔루션이나 소프트웨어/하드웨어 솔루션을 사용하면서 수행될 수 있다. 각각의 사용자는 "Digital Pathways" Key-card가 할당된다. 사용자의 ID 코드를 넣었을 때 이것은 한 차례 유효한 패스워드를 줄 것이다. 이것에 대한 또 다른 것은 사용자의 방화벽 서버상에 로그인 쉘을 위치

시키는 소프트웨어이다.

3.2 Privacy Enhanced Mail(PEM)

PEM은 아주 중요한 정보를 암호화하는 RSA기반의 암호화 스키마이다. 그러나 메시지에 대한 무결성을 체크하는 것이 더 나을 수 있고 원본의 메시지를 보내는 것이 거부되지 않게 한다. PEM은 실제로 symmetric(private-key)과 asymmetric(public-key)의 암호기법의 사용을 허용하도록 설계된 프로토콜이다.

3.3 Kerberos

Kerberos는 패스워드 같이 클라이언트로부터 서버 데몬 처리로 네트워크를 통해서 보내지는 중요한 정보를 암호화하기 위한 PEM 기반의 암호화 스키마이다. 이 네트워크 서비스는 "tickets" 권한을 위해서 Kerberos 서버에 자동적으로 응답을 만든다. 사용자는 클라이언트/서버 프로그램을 가질 필요가 있으며 이는 새로운 애플리케이션을 만들고 kerneros 라이브러리를 사용할 수 있기 때문이다. Kerberos tickets은 /tmp에서 로컬로 획득되기 때문에 주어진 워크스테이션에 한 명 이상의 사용자가 있으면 충돌의 가능성이 항상 존재한다. Kerberos는 또한 작동을 위해 시스템 시간에 의존하기 때문에 그것은 보안 시간 서버를 포함하기 위해서는 장차 확장되어야 한다.

3.4 Private-Key Certificates

이것은 실제 구현된 것이 아니라 메일과 같은 애플리케이션에 네트워크 보안을 추가하기 위한 PEM의 대안으로의 디자인 제시라고 할 수 있다. 간단히 올려만 놓으면 private-key 암호기법을 가진 public-key 스타일 수행을 사용한다. 이것은 서로 다른 타입의 애플리케이션이 적용될 수 있다. 또한 이것은 어떤 암호화 알고리즘도 플러그인 할 수 있다. 이것은 더 이상 public-key 프로토콜이 public-key 암호화에 의존하지 않게 하기 위해서 디자인되어진 것이다.

3.5 Multi-Level Security(MLS)

사용자는 네트워크 보안을 만들기 위해서 모든 것을 시험해 본 후에 그때 MLS는 다음의 논리적인 단계가 될 것이란 것을 인지하고 있다. 그러나 이것은 MLS를 실행하기 전에 모든 것을 하기까지 기다려야 한다는 것은 아니다. 단지 기본적인 것을 바꾸기 전 사용자가 n번째로 가기까지 사용자의 시간을 낭비하고 있을지 모른다는 것이다. 많은 Unix 벤더는 MLS 버전을

채용하기 위해서 연습하거나 채용하고 있는 중이다. 바로 그 예는 AT&T 시스템의 V/Release 4/MLS이다. 기본적으로 사용자 OS의 일부분으로 MLS를 구입할 수도 있다. 그러나 이것은 MLS 보안프로그램을 만들기 위한 도구들이다.

3.6 File Encryption

사용자는 그들이 공공 통신망을 통해서 데이터를 보내거나 공공 저장장치에 저장할 때마다 파일을 암호화하는 습관을 가져야 한다. 파일 암호화는 방탄조끼 같은 것이 아니다. 그러나 중요한 정보를 위해서는 명확한 텍스트보다는 훨씬 보안적이며 Unix 암호화 유틸리티는 이런 도구들의 최소한의 보안이다. 그것은 잘 알려진 암호화 기술을 사용해서 깨뜨릴 수 있기 때문에 UNIX DES 유틸리티는 더 보안적이다.

3.7 Secure Programming Methods

프로그래머는 잠재적인 침입자가 실수나 버그를 이용할 기회를 줄이는 방향으로 보안에 많이 신경 쓰고 있다. 여기에는 다음과 같은 권고사항이 있다.

1) 결코 하나의 SETUID 쉘 스크립트를 만들지 마라. 루트에서 동작하는 쉘 프로그램에 접근할 권한을 가지도록 침입자에게 허용할 수 있는 기술이 있기 때문이다.
2) 어떤 시스템에서의 완전한 루트를 포함해서 파일 이름을 기록하라.
3) 사용자가 SETUID에 접근해서 읽을 이유가 없기 때문에 패스워드를 4자리나 3자리로 잡아두어라.

3.8 Counter Intelligence

침입자를 알아내고 입증하기 위한 프로그램을 확장하기 위해서 사용자는 침입자가 흔적을 남기도록 하기 위해서 보안 도구들의 어떤 것을 수정하길 원할지 모른다. 그리고 침입이 시도된 것은 로그를 남기도록 하고 싶을지 모른다. 이런 정보는 침입자를 찾아내는데 도움이 되고 보안 침입에 대한 공격적인 자세를 가지는데 아주 중요하다. Compartmented Mode Workstations(CMW)와 같은 특별한 솔루션을 조사할 지 모르는 사용자의 요구사항에 의존하기 위해서는 End-to-end Data Link Encryption과 TEMPEST 등을 사용해야 한다.

4. 보안정책

사용자의 "보안정책"은 결합력이 있어야 하고 효과적이고 철저한 보안 프로그램을 한데 묶

는 것이다. 사용자의 정책을 조직적으로 세우기 위해서 고려해야 할 몇 가지 이유가 있다. 본 논문에서는 다음과 같은 정책을 가지고 구현하는데 초점을 두었다. 이것은 사용자가 가지고 있어야만 하는 가장 큰 이유 중의 하나이다.

① 사용자가 필요한 보안을 어떻게 구성할 것인가?
② 방어를 위해 꼭 필요한 것이 무엇이며 개인 ID는 어떻게 설정 할 것인가?
③ 부수적인 결합력과 수행을 어떻게 할 것인가?
④ 사용자가 해야 하는 가장 낮은 순위의 필요사항이 무엇인가?
⑤ 1차 보안에 통과한 후 2차 보안은 어떻게 설정 할 것인가?
⑥ 최종통과후의 패스워드를 재시도할 것인가?
⑦ 사용자들을 위해서 채택되어야 할 고려사항은 무엇인가?
⑧ 사용자가 허용될 수 있는 일과 그렇지 않은 것이 무엇인가를 판별할 수 있는가?
⑨ 패스워드 허가를 위해서 Crack을 사용하는가?
⑩ 동료에게 자기 계정을 주는 것을 허용하는가?

이러한 질문들에 의해서 사용자는 수행하기를 원하는 방법과 그 패키지가 무엇인지 또한 그것들을 구성하고 수정하기 원하는 방법을 결정하게 된다. 보안정책이라고 하는 것은 그것의 리소스와 컴퓨터에 접근할 규칙의 공식적인 명세이다. 그것을 모니터하고 네트워크 보안을 유지하도록 돕기 위해서 설치하는 툴이 무엇이건 간에 사용자는 "사용자의 정책"을 수행하기 위해서 구성되어져야 한다. 그렇지 않으면 수행되어질 필요가 있는 모든 업무를 수행하기가 어려워지기 때문이다. 그러므로 사용자는 먼저 하나의 정책을 반드시 가지고 있어야 한다.

Ⅲ. 2중 보안 Simulation의 구현

보안의 위험에 대한 이러한 가정을 토대로 보안정책을 종합적으로 판단할 수 있다. 그것의 요점은 어떻게 패스워드를 까다롭게 설정할 것이며 타인에게 유출되지 않을까 하는 것이다. 이 패스워드는 결코 명확하게 전송되어 질 수 없을 것이다. 왜냐하면 패스워드가 보안되지 않은 양식으로 전송되어져야 한다면 One-Time password가 사용되어 져야 하기 때문이다. 본 논문에서는 이러한 패스워드의 단점을 해소하기 위한 방안으로서 파워빌더를 이용하여 시뮬레이션 해 봄으로써 실제 네트워크상에 적용할 수 있도록 그 방법론을 제시하였으며 특히 수시로

변하는 개인 ID를 사용함으로써 좀더 보안이 유지되는 암호화 기법에 대해 연구하였다.

또한 본 논문은 프로토타입 시스템을 정립한 것과 같은 각각의 방어 메커니즘에 대해 고찰하고 있다.

1. 사용자 인증

보안에서 요구되는 신원의 타당성을 수립하기 위해서 개인 네트워크의 타당성을 수립함으로써 속이기 위한 transaction에 대한 방어를 제공하기 위해 국가가 컴퓨터 보안센터의 "Red Book"에 정의되어 있는 것으로 인증을 사용한다. 사용자의 신원은 사용자 이름이나 패스워드의 사용을 통해서 컴퓨터 상에서 이루어진다. 패스워드는 그 비밀을 지키고 추측하기 어려워야 한다. 단지 사용자는 시스템을 이용하기 위해서 사용자 이름과 패스워드를 알면 접근이 가능하게 된다.

사실 패스워드는 추측하기 쉬운 취약점을 가지기도 한다. 또한 외부 연결망에 대한 사용자를 인증하는 경우에는 사용자 이름이나 패스워드에 대한 정보를 획득할 수도 있다. 비록 패스워드가 에코(echoed)되지 않았어도 분명한 것은 통신연결을 통해서 전송된다. 따라서 사용자 이름과 패스워드가 좋은 신원판단 기준을 구성하고 있지 않을 수 있어 쉽게 포착되거나 추측될 수도 있다. 프로토타입 시스템에서 사용자 인증은 신원에 대한 좋은 신뢰를 제공할 수 있는 형태로 이루어진다. 이것은 한 번의 패스워드나 인증 디바이스, Digital Pathways, SecureNet, Security Dynamics SecurID와 같은 것으로 완성될 수 있다. 여기에는 강력한 사용자 인증을 이루는 방법들이 존재하는데 one-time password, digital pathway card 방법, Secure ID Card방법 등이 있는데 본 논문에서는 Secure ID Card 방법을 응용하여 수시로 변하는 개인 ID 질의어 형태로 구현하여 보았다.

2. Secure ID와 개인 ID의 질의어 방법

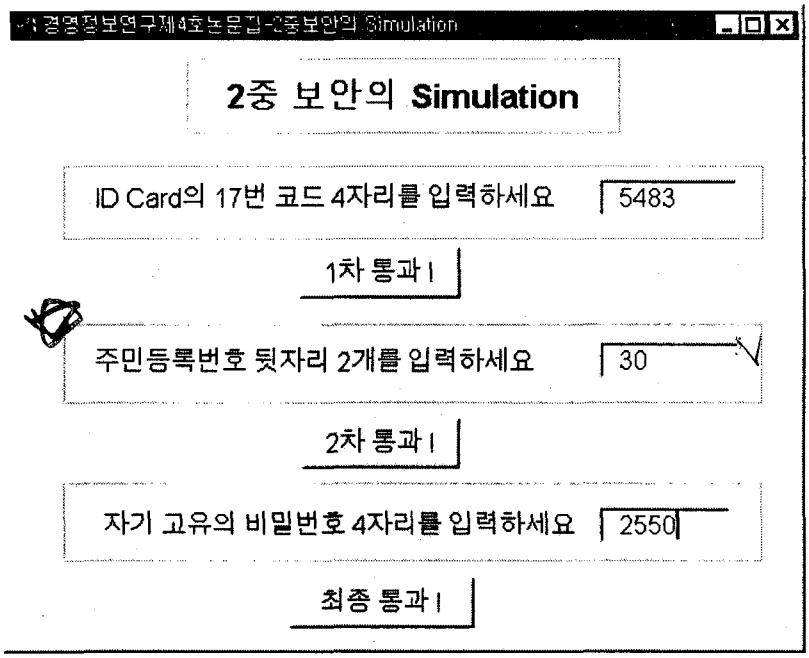
SD(Security Dynamics)로부터의 Secure ID는 유사한 방식에서 작동한다. 각각의 Secure ID 카드는 암호에 대해 사용되어지는 유일한 원천을 가졌다. 매 30초마다 그 카드는 다른 수치를 보여주는데 이는 원천과 데이터와 시간에 의존한다. 서버 소프트웨어는 사용되는 로그인에 근거한 같은 알고리즘을 적용한다. 또한 시도할 단계를 인증과정으로부터 이동시키지만 사용자 인증에 대한 다른 방법들과 유사하다. 개인적인 신원번호도 이 장치 또한 사용된다. Secure ID가 PIN에 들어가는 것을 필요로 하지 않는 장치 버전을 가졌지만 그 카드에 사용

자 신원을 의뢰하는데 이것은 위에서 S/Key와 함께 언급된 위험을 피하기 위해서이다. Secure ID의 손상이라고 하는 것은 같은 반응이 일어날 때 고정된 시간주기가 나타날 때이다. 이러한 것은 이 시점마다 단지 하나의 접속을 허용하면서 소프트웨어에 잘 맞을 수 있다.

본 논문에서는 Secure ID 카드와 개인신상에 관한 질의어를 사용하는 2중 보안에 관한 방법론을 제시하고자 한다. 그렇게 함으로써 조금은 불편하지만 타인이 패스워드를 도용하였다 해도 매 30초 간격으로 바뀌는 개인 신상에 관한 질의어에 응답할 수 없을 것이므로 2중 보안을 강력히 하고자 한다.

3. 2중 보안의 Simulation

다음 〈그림 1〉은 그 예로서 2중 보안에 관한 Simulation을 구현해 본 결과이다. 물론 간단한 파워빌더 DB를 이용하여 작성되었지만 네트워크와 사이버 공간, 실생활에서도 용이하게 사용되어져 보다 철저한 개인의 암호화에 도움이 되었으면 한다.



〈그림 1〉 2중 보안의 Simulation 구현 예

위의 그림에서 첫 번째 ID Card는 1번부터 30번까지 부여되어 있으며(현재 일부 국내 은

행에서 사용중) 항시 고정되어 있고 두 번째 질의어는 매 30초 간격으로 다음과 같은 질문들이 계속해서 무작위로 바뀌게 되어있다.

1. 주민등록번호 뒷자리 2개를 입력하세요(880124-195873<u>0</u>).
2. 당신의 나이를 입력하세요.
3. 당신의 집 번지를 입력하세요.
4. 주민등록번호 앞자리 두 개를 입력하세요(<u>88</u>0124-1958730).
5. 당신의 성씨를 입력하세요.
6. 주민등록번호 앞자리 중 뒤에서 2개를 입력하세요(880<u>12</u>4-1958730).
7. 주민등록번호 뒷자리 중 앞에서 2개를 입력하세요(880124-<u>1</u>958730).

세 번째 질의어인 자기 고유의 비밀번호 4자리는 일반적으로 사용되어지는 4자리의 숫자들로 구성된 조합의 형태이다. 세 번째 질의어가 통과되어야만 다음의 진행과정을 따라 갈 수가 있다. 본 논문에서 구현된 핵심은 두 번째 질의어로서 타인이 도용했다 할 지라도 그 다음의 질문에 응답하기란 매우 힘들다는 점이다. 물론 개인의 데이터베이스는 위의 질의어에 알맞게 작성되어져 있다는 가정 하에서이다.

Ⅳ. 결 론

인터넷에 연결된 시스템은 항상 보안의 위협에 노출되어 있다. 사용자는 공격으로부터 시스템을 보호하려면 침입자가 데이터 약점, 소프트웨어 약점, 물리적-시스템 약점, 전송 약점 등과 같은 경로를 통해서 공격할 것임을 예상해야 한다.

본 연구는 네트워크 2중보안 툴을 구현해 보고자 파워빌더를 이용하여 Simulation해 보았다. 또한 프로토타입의 실행과 그 시뮬레이션의 결과들을 구현해 봄으로써 네트워크상에서도 확장, 사용 가능하게 하여 이미 존재했던 보안의 개념에서 각각의 방화벽 계층을 두껍게 하여 철저한 개인 정보 보안을 유지해 보고자 함에 그 목적을 두었다. 네트워크영역에서 보안을 유지한다는 것은 tradeoffs에서 흥미로운 일이다. 많은 툴들이 기성품에서 유용하지 못하기 때문에 일정량의 소프트웨어 개발이 병행되어야만 한다. 전체 범위의 정보 상호운용은 문제를 일으키고 있으며 극복되어져야만 하고 패스워드의 선택은 코드를 재위치 시키거나 구성을 바꾸어 봄으로써 또는 사용자의 습관을 바꾸도록 독려함으로써 문제를 푸는 방법이 강구되어졌으나 이

제는 다이나믹한 개인의 정보를 이용하여 본인만이 알 수 있는 코드의 설계가 필요한 시기이다.

이에 즈음하여 본 논문에서는 4-GL 언어 중에서 특히 파워빌더를 이용하여 Secure ID를 병행한 개인의 질의어를 패스워드로 이용함으로써 보다 신중한 보안개념의 패스워드를 구현하여 네트워크와 사이버공간, 실생활에서도 응용될 수 있기를 바라는 의미에서 시뮬레이션 해 보았다.

아직까지 DB가 완벽히 설계되지는 않았지만 점차적으로 늘려 간다면 개인의 패스워드를 도용 당하는 해킹의 사례는 없을 것이다. 더 좋은 방법론들이 많이 있겠지만 본 논문에서는 실제로 사이버공간에서 Simulation해 보았고 앞으로 더욱 철저한 보안의 방법으로 설계된 보안 관련 소프트웨어를 수행하는 것은 보안의 결과에 대한 확신을 가지기 위한 필요충분조건을 제기하게 한다.

참 고 문 헌

1. http://www.microsoft.com/intdev/security/security.html, Microsoft Internet Security Framework.
2. 한국정보보호센터, "정보시스템 해킹 현황 및 대응," 1996.
3. CERTCC-KR-TR-97001, "전산망 해킹 침해사고시 어떻게 처리하나," 1997.
4. Defense Science Board, "Report of the Defense Science Board Task Force on Information Warfare-Defense," Nov. 1996.
5. Garfinkel & Spafford, "Practical UNIX & Internet Security," 2nd Edition, O'Reilly Association, 1996.
6. Hans Husman, "Introduction to Denial of Service," Feb 9, 1997.
7. Imprimerie Nationale : Projet de Loi de Finances pour 1995, Etat de la Recherche et du Developpeonent Technologique, Paris, 1994.
8. Marcus J. Ranum, "Thinking About Firewalls," *Proceedings of Second International Conference on Systems and Network Security and Management*, April, 1993.
9. N. haller and C. Metz, "A One-Time Password System," *IETF RFC* 1938, 1996.
10. Phil Karn, Neil M.Haller, and John S. Walden, Bellcore, S/Key software kit, available via anonymous FTP from thumper.bellcore.com: /pub/nmh/skey/*.
11. Stephen T. Kent, "Internet Privacy Enhanced Mail(PEM)," *Communications of the ACM*, August, 1993.
12. Winfield Treese and Alec Wolman, "X Through the Firewall, and Other Application Relays," Cambridge Research Lab. Technical Report 93/10, Digital Equipment Corporation, May 3, 1993.

Abstract

Implementation of Expert System Simulation based on 2th Security of Network

Lee, Chang-jo

Organizations rely on Secure ID resources today to handle vast amounts of information. Because the data can vary widely in type and in degree of sensitivity, employees need to be able to exercise flexibility in handling and protecting it. It would not be practical or cost-effective to require that all data be handled in the same manner or be subject to the same protection requirements. Without some degree of standardization, however inconsistencies can develop that introduce risks.

Policy formulation is an important step toward standardization of security activities for ID resources. ID security policy is generally formulated from the input of many members of an organization, including security officials, line managers, and ID resource specialists. However, policy is ultimately approved and issued by the organization's senior management. In environments where employees feel inundated with policies, directives, guidelines and procedures, an ID security policy should be introduced in a manner that ensures that management's unqualified support is clear. The organization's policy is management's vehicle for emphasizing the commitment to ID security and making clear the expectations for employee involvement and accountability.

This paper will discuss ID security policy in terms of the different types (program-level and issue-specific), components, and Implementation of Expert System Simulation based on 4GL, PowerBuilder.