데이터가 누설되고 있다 … 철저한 보안책 시급

권한 · 인증 · 암호기술이 열쇠, 스마트 · PC 카드 활용 도모

인터넷의 발전과 함께 시스템 보안에 대한 절대적 필요성이 날로 증가하고 있다. 외부에서 HTML 페이지로 액세스하는 것을 제한하는 방법은 무엇일까? 인터넷을 통한 트랜잭션을 보호할 수 있을까? 해커의 침입을 막을 수 있는 방법은? 심지어 인터넷에 노출돼 있는 금융권에서도 그들의 가장 큰 위험 요인은 내부에 잠재해 있다고 밝힌다. 대규모 피해를 가져오는 원인에는 꼭 악의적 의도가 요구되지는 않는다. 단순한 실수라도 상당한 피해를 불러일으킬 수 있기 때문이다. 삭제된 데이터베이스 파일이 사고나 나쁜 의도에 의해 바로 거대한 문제로 이어질 수도 있다. 특정 시스템에 대해 사용자를 인증하고 행위에 권한을 부여하는 방법이 악의적인 행동을 막는 것보다는 훨씬 현실적인 대안이 될 것이다. 이같은 보안책은 각종 사고로 인한 피해를 막을 수 있도록 도와준다. 어떤 보안 솔루션이 내외부적 위험을 막을 수 있는지가 중요한 관건이다. 〈편집자〉

지금까지 대부분의 기업들은 잘 관리된 데이터 센타를 통해 그들의 시스템을 안전하게 보호하는 물리적 보안방법에 의존해왔다. 그러나 분산시스템 세계에서는 보다 확실한 보안책이 필요하다. 시스템의 상당 부분이 외부에 위치하고 있어 외부로부터 차단된 데이터 센타내에서 보호되던 때와는 상황이 변했기 때문이다. 예를 들어, 제조업의 경우 공급자들의 컴퓨터 시스템이 제작자의 시스템과 직접 연결돼 있는 경우가 많다. 각 시스템이 인터넷을 통해 액세스될 경우 엔드유저는 세계 어느 곳에나 존재할 수 있는 것이다.

분산시스템 환경에서 보안에 관한 희소식은 분산시스템을 보호하기 위한 관련기술들이 이미 존재한다는 사실이다. 반면, 보안시스템을 분산된 시스템에 쉽게 어드레스할 수 없다는 단점이 있다. 만일 곳곳에 문제의 요소가 있다면 조직의 구성은 허술해질 수 밖에 없기 때문이다. 또한 제품들의 단순한 조합만으로는 상업적, 분산적 오픈시스템 환경에 완벽한 보안책을 제공할 수 없다. 암호나 권한을 위한 솔루션을 제공하는 것이 상대적으로 쉽기 때문이다.

다른 한편으로는 사용자 인증을 위한 핵심 솔루션을 제공함으로써 분산 환경에서 관리될 수 없는 부분까지 빠르게 관리할 수 있다. 분산시스템 사용자들은 그들 스스로가 여러 시스템에서 동일하게 수행할 수 있기를 원한다. 여기서 착안한 것이 싱글 사인-온(single sign-on) 개념이다.

싱글 사인-온은 사용자가 스스로 인증할 수 있어 데이터베이스를 포함한 여러 시스템에 액세스할 수 있다. 이것은 점차 중요해지고 있는 개념으

로 단순히 편리한데 그치는 것이 아니라 패스워드 사용의 수를 감소시켜 패스워드가 재사용되고, 어딘가에 적어 놓는 등 다른 방법으로 노출되는 일을 최소화시킴으로써 보안을 대폭 향상시킬 수 있다. 분산시스템을 위한 필요 요소들이 존재함에도 불구하고 표준, 오픈 아키텍처, 완벽한 보안 솔루션을 상호 운영할 수 있는 제품들과는 거리가 멀다.

보안이란 무엇인가

보안에 대한 정의는 저마다 다르게 내려지고 있다. 이것이 보안의 기본 정의를 내리는 좋은 전초전이 될 것이다. 어떤 사람은 보안이 컴퓨터 바이러스 또는 해커가 인터넷을 통해 서비스 이외의 범위를 침해하는 것이라고 말한다. 다른 이는 사적 영역(privacy) 및 암호(encry-ption) 또는 인증(authentication) 및 권한(authorization)이라고 말한다. 보안시스템은 어드레스되는 모든 영역이 해당된다. 각 보안 요소들은 다음 4가지 기준을 따른다.

● 인증

- 그가 누구인지 말할 수 있는가?

- 그 객체(프로그램 같은)가 무엇인지 말할 수 있는가?

- 메시지는 어디서부터 오는가?

- 사람들이 어떤 것을 부인(거절)할 수 있는가?

● 권한

- 어떤 사람이나 그룹이 허용되는가?

- 어떤 프로그램이 허용되는가?

● 암호

- 누가 정보를 보도록 허락되어 졌는가?

● 시스템 보호(System Protection)

- 시스템 피해에 대해서는 바이러스 대처, 방화벽, 프록시 등이 보호를 하고 있다. 이들은 각 서비스 침해에 대응하고, 우연히 발생할 수 있는 시스템 오류를 최소화하기 위한 조치를 취한다. 이 보호는 인증, 권한, 암호기술 등을 통해 부분적으로 어드레스될 수 있다. 이밖에도 수많은 바이러스 감지 및 제거 소프트웨어들이 있으며, 자바와 같은 바이러스 방지 시스템 환경도 같은 종류에 포함될 수 있다.

이 글은 특히 분산 환경에서의 권한, 인증, 암호에 대해 집중적으로 설명할 것이다. 더불어 싱글 사인-온의 문제점도 함께 살펴보겠다.

현재 이들 영역들을 포괄하는 솔루션들이 다양하게 출시돼 있으나 아직까지 완벽한 보안 아키텍처를 구축할 수 있는 제품은 찾을 수가 없다. 예를 들어, SSL (Secure Socket Layer) 구축을 통해 인터넷 서버 인증 및 암호화를 하려고 할 때 SSL의 최신 버전(현재 버전 3.0)임에도 불구하고 사용자 인증까지는 어드레스하지 못한다. 다양한 커버로스(Kerberos) 기반 개인(private) 키 메카니즘은 이 3가지 기능을 모두 제공한다. 그러나 이 경우도 수천명의 사용자 키가 관리될 때에만 사용할 수 있다.

최근에 새롭게 부각되는 몇가지 보안 표준이 있다. 보안 서비스에 프로그램을 인터페이스할 수 있는 GSS-API(Generic Security Service Appli-cation Program Interface)와 그밖에 다수의 암호 알고리즘인 DES 및 RSA가 있다. 이들은 메시지 암호방식을 지원하고 있으며, MD5와 SHA 해싱 알고리즘은 디지탈 '지문'을 만들어 준다.

보안의 올바른 인식

보안에 대해 논할 때 많은 사람들은 암호 키의 길이와 암호의 기술적 문제에 대해 흥분하곤 한다. 이러한 문제들도 중요하지만 진짜 위험에 처해있는 곳에 대해서는 간과하고 있는 경향이 있다. 보안에는 2가지 기본 규칙이 있다.

먼저 세상에는 완벽한 보안이란 없다는 점이다. 보안에는 항상 비용과 사용의 난이성을 감수해야 하는 한 측면과 위험이 있는 다른 측면이 있다. 기술에 투자한 비용에 따라 위험을 증가시킬 수도 있고, 감소시킬 수도 있다. 따라서 양 측면의 적절한 조화가 이루어져야 한다. 그것은 보안 관리에 있어 매우 중요한 요소로 위험도를 평가하고, 보안에 투자할 수 있는 비용 및 사용의 난이성에 대한 균형을 맞춰야 한다. 이 문제에 대해서는 뒤에서 다시 거론하겠다.

두번째 기본 규칙은 가장 큰 보안 노출이 항상 사람들의 작업 방식에서 이루어 진다는 점이다. 아무리 우수한 기술이라도 사람들이 안전하게 사용하지 않는다면 시스템의 안전성은 보장될 수 없다. 따라서 성공적인 기술적 솔루션은 안전한 작업 방식과 불안전한 작업 방식을 찾아낼 수 있는 능력에 따라 그 성과를 발휘할 수 있다고 할 수 있다.

보안의 가장 큰 문제중 하나는 시스템의 확장에서 비롯된다. 필자가 개인적으로 내부 및 인터넷상의 수백여 사이트에 액세스했을 때 많은 사이트들이 액세스에 필요한 패스워드를 요구했다. 이것은 딜레마에 빠지게 한다. 과연 각 사이트별로 다른 패스워

드를 이용하고 그 패스워드를 잊어버릴 위험을 안고 있어야 하는가, 아니면 그것들을 어딘가에 적어둬야 하는가? 또는 패스워드 몇개를 재사용하기로 하고, 한 웹사이트에 사용했던 패스워드를 다른 곳에 또 사용하는 것이 적합하겠는가?

만약 패스워드가 재사용된다면 패스워드가 도용될 위험이 있다. 예를 들어, 우수 웹사이트의 자료를 무료로 사용할 수 있는데, 단 패스워드를 사용해야만 들어갈 수 있다고 해보자. 그 사이트에서는 접속할 때마다 자주 패스워드를 바꿀 것을 요구할 것이다. 사용자들은 시간이 지날수록 그동안 사용했던 패스워드를 잊게될 것이고, 사용자는 계속 다른 패스워드를 만들어낼 것이다. 그런 사이트에서는 아마도 여러 패스워드들을 수집할 수 있게 되고, 이를 다른 곳에 도용할 가능성을 갖게 되는 것이다.

이것이 물리적 토큰(token)이나 생물측정(biometric) 기반의 인증, 싱글 사인-온 등이 사용되는 중요한 이유임을 시사한다. 그것은 사람들의 작업 방식으로부터 보안이 노출되는 위험을 줄이는데 도움을 줄 것이다.

특정 문제 영역

분산시스템 세계에는 수많은 특수 영역이 있으며, 이 곳에는 일반적으로 사용되는 보안체계보다 더 안전한 기능의 필요성이 제기되고 있다. 이 경우 다음 기능들이 포함된다.

- 웹 페이지로의 제한된 액세스 자격
- 분산 객체, 특히 코바/IIOP를 포함하는 커뮤니케이션을 안전하게 하는 자격
- 인터넷/인트라넷/엑스트라넷 환경에서의 싱글 사인-온으로 사용자 스스로를 동일화해 단 한번 액세스함으로써 여러 시스템으로 액세스할 수 있도록 허락하는 것
- 분산시스템에서 다른 시스템으로 가도 스스로 동일화할 수 있는 시스템 자격(사용자만이 아니라)
- 회사간(엑스트라넷 포함) 안전한 커뮤니케이션을 제공하는 자격
- 수많은 보안시스템을 조합하기 위한 자격, 예를 들어 운영체제 보안 및 데이터베이스 보안을 포함하는 웹 보안의 동기화
- 분산 환경내 레거시 애플리케이션으로의 안전한 액세스

필자는 이상의 특정 문제 영역에 대한 기준 항목을 중심으로 분산시스템 보안 구조의 청사진을 논하고자 한다.

분산 아키텍처

많은 조직들은 3/4계층의 웹 기반 분산 아키텍처로 이동하고 있다. 이 아키텍처는 가벼운 HTML 또는 자바 기반 클라이언트(티어 1)를 웹 서버로 HTTP(티어 2)를 통해 액세스한다. 웹 서버는 비지니스 객체 계층이나 티어 3(역시 애플리케이션 서버라고 불린다)로 코바나 메시지 기반 미들웨어를 통해 액세스된다.

티어 3는 티어 4로 액세스되는데, 티어 4에는 RDBMS들이 SQL 기반 미들웨어를 통해 액세스되거나 메인프레임 레거시 시스템이 메시지 기반 미들웨어 또는 분산 TP모니터를 통해 액세스된다. 메시지 기반 미들웨어로는 IBM의 MQ시리즈를, 분산 TP 모니터로는 BEA시스템의 텍시도를 예로 들 수 있다. 〈그림〉

3계층 구조를 예로 들면, 트랜잭션이 비지니스 객체 계층에서 IIOP를 통해 직접 제공된다. 3계층은 레거시 시스템이나 데이터베이스 모두가 해당되며 4계층의 예와 같다.

최근 대부분의 아키텍처들은 몇가지 다른 제품들을 설치하면 안전하게 프로세스를 진행할 수 있다. 단순히 클라이언트에서 웹으로(client-to-web) 가는 서버 암호를 위해서는 SSL이 일반적으로 사용된다. 사용자 인증은 일반적으로 단순한 패스워드 검사 소프트웨어를 웹 서버상에서 돌리는데 그칠 뿐이다. 비지니스 룰층의 보안은 보통 벤더 전용이다. 예를 들어, 텍시도 버전 6.3은 링크-레벨 보안을 위해 공용 키 암호(public key encryption)를 설치했다. DCE 기반 분산시스템을 위해 보안 메카니즘은 커버로스에 준한다.

현재 클라이언트와 트랜잭션 서버간의 안전한 코바/IIOP 트래픽을 위해서는 몇가지 옵션이 있다. 코바는 DCE를 전송층으로 이용했고, DCE 보안이 일반적으로 이용된다. OMG는 안전한 IIOP를 위한 표준을 제안해왔는데, 현재까지 어느정도 수용되고는 있지만 아직 어느곳에서나 완전히 적용되지는 못하고 있다.

웹 보안에도 새로운 세대교체가 이루어지고 있는데, 이것은 웹페이지의 단순한 패스워드 보호를 넘어선 것이다. 새로운 보안 기능은 2가지 클래스로 요약된다.

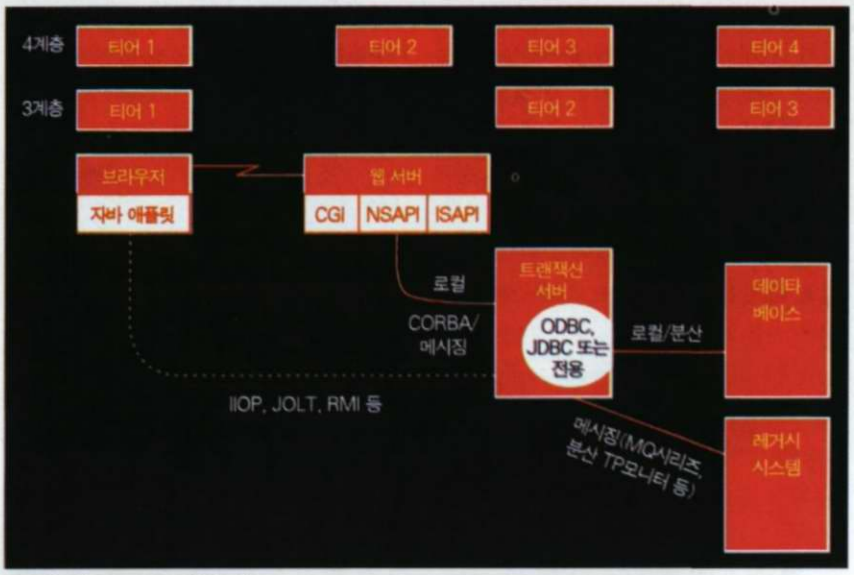
첫번째 접근법은 방화벽이다. 보안 서버가 웹 서버의 앞에 위치하고, 다른 서버들은 주요 웹 서버로부터 인증되고 권한을 부여받은 사용자에게 해

당되는 보안 방법을 이용한다(넷액세스사가 이 접근을 하고 있다). 예를 들어, 넷스케이프 커뮤니케이션은 넷스케이프 시큐리티 API를 인텔의 CDSA(Common Data Security Architecture) 1.2 기반으로 소개했다. CDSA는 표준 보안 아키텍처를 향한 야심을 가진 제품이다. 이 구조는 커먼 시큐리티 서비스 매니저로 구성됐는데, 이는 애플리케이션과 추가되는 모듈의 세트 사이에 위치한다. 이곳에서는 암호표기(cryptographic) 서비스, 증명, 믿을만한 정책 등을 제공한다.

CDSA는 모든 분산시스템 분야의 거대한 보안 아키텍처로 촉망받는 제품이기는 하지만 아직 시기상조다(CDSA 버전 1.2는 올 3월에 발표됐다). 완전한 기능을 제공하지 못하고 있으며, GSS-API가 애플리케이션 인터페이스를 가지고 직접 보안 서비스를 하는 것처럼 커먼 시큐리티 인터페이스 층이 표준 보안 API를 위해 필요하다.

보안 아키텍처

안전한 분산 환경은 항상 물리적 보안 및 논리적 보안(인증, 권한, 암호)의 조합이 이루어져야 한다. 적합한 보안을 제공하기 위해서는 기술적인 문제를 제외하고도 보안 프로토콜과 사용자의 작업 방식이 부합돼야 한다. 아무리 우수한 물리적, 논리적 보안체계라도 몇몇 인증된 사람이 정보를 유출해 간다면 속수무책이기 때문이다. 기술적 보안은 항상 완벽한 보안시스템의 한 부분이 돼야 하며, 증명된 프로토콜과 수동적 프로시저가 통합돼야 한다.



〈그림〉 3/4계층 아키텍처의 예

사람들은 종종 시스템의 특정 부분이 물리적으로 안전하다면 논리적 보안 부분을 간과하는 경향이 있다. 예를 들어, 센서가 물리적으로 안전하지 않거나 아웃풋이 암호화되지 않았다면 당신은 쉽게 생체측정 인증(지문과 같은)에 속을 수 있다. 따라서 메시지가 물리적으로 안전할 필요가 없는 내용이더라도 개인 키의 안전은 철저히 유지해야만 한다.

바로 이 경우 일반적으로 사용 가능한 3가지의 방법이 있다. 첫번째는 데이터센타나 잠겨진 방과 같은 물리적 보안 환경에서 키를 포함하는 하드웨어를 보호하는 것이다. 두번째는 개인 컴퓨터를 안전하게 하기 위해 물리적으로 침입이 불가능하게 만드는 것이다. 예를 들어, 고수준을 요하는 보안 환경의 경우 컴퓨터 내부에 별도의 안전 장치가 설치돼 있는 경우와 같은 것이다. 끝으로 비용 효과적인 방법으로 키와 암호화 엔진을 안전하게 보호하는 것이다.

최근 솔루션들은 관리가 가능한 PCMCIA 카드나 스마트 카드를 이용해 키나 암호화 엔진을 안전하게 할 수 있다. 유럽 휴대폰 산업은 스마트 카드를 GSM(Global System for Mobile Communications) 시스템을 이용해 보안용으로 사용하고 있다.

키는 물리적으로 충분히 안전하지 않다는 사실을 주지해야 한다. 암호화 엔진 또한 키와 함께 안전하게 만들어야 한다.

그렇지 않으면 도용된 키가 키의 안전과 암호화 엔진 사이의 허술한 부분을 틈타 정보를 가로챌 수 있다.

보안 프로토콜

도청, 가로채기, 거부, 메시지 변조 등 다양한 시나리오에 대응하기 위한 다양한 보안 프로토콜이 있다. 하지만 필자는 인증, 권한, 암호에 대한 메카니즘을 설명할 것이다.

개인 키 암호화는 공용 키 기술보다 매우 낮은 컴퓨팅 파워를 요구하며, 속도는 더 빠르다.

그러나 분산돼 있는 구성원들과 비밀을 공유해야만 한다. 소수의 사용자인 경우 키를 물리적으로 분산해 사용

할 수 있다. 예를 들어, 프린팅 기술을 보호하는 것과 같다. 종종 개인 키는 공용 키 기술을 이용해 분산된다.

커버로스는 개인 키 암호로 메사추세츠 기술연구소에서 개발한 방법이다. 이것은 중앙(복제될 수 있는) 보안 서버에 기반하고 있어 모든 사람들이 신뢰하는 방법이다. 상업적인 구현은 사이버세이프사와 같은 회사들을 통해 이용가능하다. 모든 사용자들은 자신만의 개인 키를 가지고 있어 보안 서버와만 정보를 공유할 수 있다. 이 장치는 사용자가 개인 키를 가지고 각각의 사용자/서버와 쌍을 이뤄야 하는 것이다.

커버로스 프로토콜은 사용자가 기계에 액세스할 수 있는 티켓에 유통기한을 두고 있다. 티켓이 가로채이는 것을 방지하기 위해 커버로스는 암호화된 시간 도장을 이용, 각 티켓에 제한된 시간을 부여한다. 커버로스의 기본시간은 5분으로 그 이상이 되면 시간 도장이 거부당하게 된다. 따라서 5분안에 티켓이 클라이언트 머신을 떠나 서버에 도착해야 하는 것이다. 불행히도 이 방법은 모든 클라이언트와 서버가 정확히 동기화된 시계를 가져야 한다는 제약이 있다. 따라서 다수의 사용자나 사용자간 시간차가 있는 지역에 있을 경우 문제의 소지가 된다. 이 경우에는 특별한 기술을 필요로 한다.

커버로스는 비지니스에 있어 안전한 서버 투 서버 커뮤니케이션을 위한 유용한 접근법이다. 당신은 복잡한 키 관리 대신에 싱글 사인-온을 위해 커버로스를 사용할 수 있다.

공용 키 기술도 인기를 끌고 있다. 현재 타원형 알고리즘을 비롯해 다른 알고리즘도 많지만 가장 범용화된 방법은 RSA 알고리즘을 이용하는 것이다. 키 관리는 공용 키만큼 단순화할 수 있다. 가장 일반적인 접근 방법은 소수의 믿을만한 키 분산 센타를 갖는 것이다. 분산 센타는 통신하기 원하는 모든 부분의 공용/개인 키의 공용 부분만 관리하는 곳이다. 그다음 각 부분은 다른 부분의 통신 안전을 위해 공용 키를 되돌려줘야 한다. 하지만 개인 키는 결코 소유자를 떠날 수 없다.

다른 기술 및 알고리즘은 디피-헬만(Diffie-Hellman)으로 공용 키 기술을 이용해 생성된 개인 키와 협상하는 것이다. 이 방법은 오히려 한 분야를 개인 키로 생성하고 다른 곳으로 보낼 수 있도록 하는 것이다. 단, 표준 디피-헬만 프로토콜은 가로채기에 대해 취약하다. 그러나 이것은 메시지를 디지탈로 사인함으로써 어드레스할 수 있다. 이밖에 다른 기술로 MD5와 SHA 등 해시 기능을 이용해 메시지가 운반되는 과정에서 정보가 바뀌지 않도록 하는 방법이 있다.

상업적 세계에 있어 가장 주의할 점은 상대가 안전하다고 생각되는 메시지를 향해 공격해 온다는 점이다. 이것은 분산 환경에서 이해할 수 있는데, 특히 인터넷의 개방성과 연관된다. 대부분의 경우 단순한 패스워드 보호를 통해 암호나 싱글 사인-온 없이 액세스하고 있다. 또 상업용 DBMS를 포함하는 대부분의 데이터베이스 보안 방법은 스탠드얼론이며, 쉽게 다른 보안 메카니즘과 통합될 수 없다.

이것은 놀라운 일이다. 왜냐하면 정보의 취약성 정도에 따라 침입자가 얼마나 오랫동안 그 데이터베이스 안에서 일을 같이할 수 있는 지를 알 수 있기 때문이다. 메시지는 짧게 존재한다. 그러나 데이터베이스에 있는 데이

웹 리소스

사이버세이프	www.cybersafe.com
게이트웨이 투 인포메이션 시큐리티	www.securityserver.com
젬플러스	www.gemplus.com
GSS-API	www.ansi.org, www.css.org
IBM	www.ibm.com/Security
인텔	developer.intel.com/ial/security/cdsa/index.htm(CDSA용)
아이리스캔	www.iriscan.com
자바소프트	썬마이크로시스템즈의 자회사, FAQ:애플릿 시큐리티, www.javasoft.com/sfaq/index.html
메사추세츠 기술연구소	최초 커버로스 개발, www.mit.edu/kerberos/www.index.html
마이크로소프트 보안 상담자	www.microsoft.com/Security
넷티그리티	www.netegrity.com, www.security.com
넷스케이프 커뮤니케이션	웹사이트상의 SSL 스펙과 지원 정보 관련 리스트, www.netscape.com
오브젝트 매니지먼트 그룹	www.omg.org
오픈 시스템 파운데이션	DCE의 근원지, www.opengroup.org/tech/dce/security
프로티그리티	www.protegrity.com
레인보우 테크놀러지	www.rainbow.com
RSA 데이타 시큐리티	www.rsa.com
썬마이크로시스템즈(썬스크린 제품)	www.sun.com/security/overview.html

터는 수년간 머물게된다.

최근 이를 이용해 트랜잭션 메시지를 안전하게 보호하는 것처럼 해놓고 관계된 비보호 데이터베이스에 액세스, 데이터를 바꿔치기하는 수법으로 상대에게 피해를 입히거나 사기 행동을 하는 것이 가장 쉬운 보안 침해 방법이 되고 있다.

DBMS 벤더들은 안전한 데이터베이스 성능을 가지고 있다. 그러나 보통 국방용으로 사용될 경우에는 상업용 데이터베이스에 공용 키 기술을 이용해 암호화하거나 보호하기란 불가능하다. 이 부분에 대한 관심은 앞으로 계속 증가될 것으로 보인다. 프로티그리티(Protegrity)와 같은 기업들은 이미 이 분야에 두각을 나타내기 시작했다. 이러한 벤더의 제품군은 일반적으로 암호와 비암호 데이터를 외부에 있는 DBMS로 필요시 옮기는 것이다.

아마 안전한 분산시스템을 위한 가장 큰 문제는 다른 보안시스템간의 통합이라고 할 수 있다. 당신의 웹사이트는 SSL을 이용하고, 메인프레임은 ACF2 패스워드 프로텍션을 이용하며, 내부 서버에서는 커버로스를 이용한다고 하자. 현재까지는 이같은 문제를 해결해 줄 일반적 솔루션이 없다.

비록 몇가지 전용 솔루션이 IBM과 같은(기업의 시큐어웨이 보안 제품군) 기업에 의해 출시돼 광범위한 환경을 보완해 주고 있기는 하다. 최근 GSS-API나 CDSA와 같은 표준들은 이 문제를 어드레스하는 작업을 시작하고 있다. 하지만 그 사이 대부분의 기업들은 몇몇 주요 솔루션 및 자신을 위해 통합 솔루션을 구매하게 될 것이다. 컨셉트 5(Concept 5)와 같은 기업은 최고의 서비스 접근을 통해 이 분야에 어드레스하고 있다.

인터넷/인트라넷/엑스트라넷

인터넷은 그동안 적대적 공간으로 인식돼 왔고, 동시에 인트라넷과 엑스트라넷 보안을 위한 좋은 테스트 공간이기도 하다. 만일 보안 기술이 인터넷에서 이루어진다면 어디서나 작업할 수 있게 될 것이다.

바로 지금 인터넷에서 강조하는 부분은 안전한 금융 트랜잭션을 제공하는 것이다. 현재로서는 매우 주의깊은 애플리케이션 코딩, 방화벽, SSL 가능 서버, 브라우저 등의 코딩이 필요하다. SSL은 유연한 프로토콜로 다양한 암호화 알고리즘을 지원한다. 여기에는 RSA, DES 등이 포함된다. 불행히도 SSL은 현재까지 HTTP 트래픽의 보안용으로만 어드레스되고 있다. 즉, 일반 웹페이지용으로만 사용된다는 것이다.

이것은 현실적 문제는 아니다. 왜냐하면 대부분의 인터넷 트랜잭션은 폼 기반이며, HTTP를 이용하고 있기 때문이다. 인터넷 트랜잭션 프로토콜인 코바/IIOP가 소개되면서 보안에 대한 또다른 채널이 되고 있다. 현재로서 IIOP는 비암호화돼 있으나 곧 안전한 IIOP가 출시될 예정이다. 안전한 IIOP의 개발은 IIOP가 인터넷을 통한 트랜잭션용으로 이용할 수 있는지를 가름할 중요한 사안이 될 전망이다.

SSL은 상용 브라우저의 표준 버전으로 이용돼 왔다. 공용 키 기술에 준한 서버 인증을 제공할 뿐 공용 키 기반 사용자 인증은 지원하지 않는다. 소프트웨어를 이용해 특정 웹 페이지가 여러 웹 서버상에서 이용 가능하도록 패스워드 보호를 하고 있지만, 이 소프트웨어는 패스워드를 서버에 저장해 이용하도록 하고 있다. 따라서 만일 누군가가 동일한 패스워드를 여러 사이트에서 이용한다면 그것을 가로채기란 어렵지 않을 것이다.

512비트 이상을 사용하는 RSA, 128비트 이상의 DES 알고리즘이 사용될 수 있는 미국에서는 SSL이 효과적이지만 이 방법은 웹 브라우저와 웹 서버간 연결을 안전하게 할 뿐이다. 웹페이지가 서버상에 저장됐을 경우 SSL을 이용하면 그 서버를 안전하게 보호할 수 있다.

분산 환경이 점차 증가되면서 웹 서버는 데이터 소스 뿐 아니라 레거시 시스템과 다른 기기에 위치한 데이터베이스로 액세스를 제공해야 한다. 방화벽을 이용해 웹 서버와 소스 시스템간에 진행되는 통신은 인터넷과 분리돼 있으며, 외부 간섭에 대해 강력하게 작용한다. 그러나 다른 한편으로는 내부 지식을 가진 방화벽 뒤에 있는 직원들에게도 보안 침해 가능성이 있기 때문에 철저한 보호가 필요하다. SSL은 미들웨어를 이용하고 있으며, 특히 서버 투 서버 커뮤니케이션에서 필요하다.

안전한 미들웨어

미들웨어는 물리적 보안 및 기기간의 특수한 커넥션을 제공한다. 미들웨어 커뮤니케이션을 안전하게 하기 위한 가장 간단한 방법 - 그러나 아마도 가장 적은 유연성과 가장 비싼 비용을 지불해야 할 것이다 - 은 하드웨어를 이용하는 것이다. 안전한 환경에서의 서버들은 암호 박스인 듯 보인다. 썬

마이크로시스템즈의 썬스크린 제품의 경우 공공 커넥션을 통해 안전하게 통신할 수 있다. 이 제품은 인트라넷이나 인터넷용으로도 사용할 수 있다.

이 접근은 견고한 포인트 투 포인트 보안을 제공한다. 보다 유연한 접근은 미들웨어 커뮤니케이션을 소프트웨어 레벨에서 안전하게 하는 것이다. 이것은 잠재적으로는 하드웨어적 접근보다 더 취약할 수도 있다. 그러나 새로운 커넥션이 필요할 때 더 싸고 유연하게 설치할 수 있다는 장점을 갖는다. 잠재돼 있는 보안의 취약성은 상업적 유닉스를 적용하거나 윈도우 NT와 같은 안전하지 않은 운영체제 환경과 관련된 소프트웨어 작동의 결과이다. 이러한 환경을 안전하게 하기 위해서는 대규모의 기술 도입이 필요하다.

IBM의 MQ시리즈와 같은 대부분의 미들웨어 제품은 개인 또는 공용 키 암호가 설치될 수 있는 보안 기능을 가지고 있다. 턱시도 5.3과 DCE 같은 제품은 그동안 보안 기능을 지원해 왔다. 그러나 대부분의 기업들은 미들웨어 제품군을 보안 기능에 상관없이 선택했다. 따라서 이것이 종종 보안문제를 야기시키는 원인이 되고 있다.

인증/싱글 사인-온

공용 키 메카니즘(RSA 같은)은 인증 소프트웨어와 분산시스템의 하드웨어 요소를 위해 적합하다. 개인/공용 키 한쌍 중의 개인 키가 물리적으로 안전하게 보호되는 한 하드웨어나 애플리케이션의 안전은 보장될 수 있다. 보안을 보다 철저히 하기 위해서는 개인 키를 물리적으로 안전하게 하고, 암호화 엔진으로 패키지화해야 한다.

최근 PCMCIA와 스마트 카드 기반 하드웨어 암호 디바이스에 대해 몇몇 업체로부터 성능, 가용성 등이 제공되고 있다. 인포메이션 리소스 엔지니어링(IRE), 매크 테크놀러지, 크리살리스-ITS 업체 등으로 HP, 인포믹스, 젬플러스 등과 파트너관계를 맺고 있다.

향후 공용 키와 개인 키 설치, 암호화 엔진을 가진 CPU 칩이 출시될 것으로 기대되고 있다. 그렇게 되면 전자 망원경을 통해 칩을 해킹하려는 사람이 줄어들게 돼 칩을 안전하게 보호할 수 있을 것이다. 인증 기법은 개인 키가 소프트웨어에 열려 있을 때는 확실한 보안이 어렵다. 이 경우 키들은 내부적으로 해커에게 드러나게 된다. 따라서 운영체제 보안에 의존해가고 있는 현실이다.

인증된 사람들의 문제는 한층 더 어렵다. 인증자와 인증된 기기 사이에서 비밀을 공유하는 보안 관리 패스워드가 더 큰 관리 문제를 야기시킬 수 있다. 과거 필자는 많은 패스워드들이 가지고 있는 기본적 문제를 발표해 비밀 유지의 필요성을 역설한 적이 있다. 공용/개인 키의 사용은 키 유지보수 문제를 크게 단순화시켰는데, 많은 개인 키들을 하나의 개인/공용 키로 줄임으로써 해결할 수 있다.

이 접근은 2가지 문제를 발생시킨다. 하나는 사람들이 암호 엔진을 자신의 머리 속에 가지고 있지 않다는 점이다. 이것은 인증할 때 사용될 수 있는 암호 엔진으로 개인 키가 진입하는 것을 의미한다. 그 키는 이 시점에서 노출되기 쉽다.

두번째 문제는 개인 키가 길고 의미가 없어 그들이 기억하기에 매우 어렵다는 점이다. 이를 해결하기 위해서는 토큰이나 생체측정 또는 2가지 모두를 이용하는 것이다. 스마트 카드와 같은 토큰의 인증 방법은 사용자가 물리적이고 독특한 것을 소유하는 것이다. 토큰 기반 보안의 가장 큰 이점 중 하나는 사용자가 공격받기 쉬운 상태가 됐을때 쉽게 알 수 있다는 점이다. 카드를 복사하기가 매우 어렵기 때문에 외부로부터의 침입 가능성은 별로 없다. 순수 토큰 기반 보안은 토큰을 잃어버리거나 도둑맞았을 경우 윈도우를 열 기회를 잃어버린다.

패스워드로 토큰을 보호하는 것은 다음과 같은 단점을 어드레스할 수 있다. 이상적인 방법은 보안을 최대화하기 위해서는 패스워드가 토큰안으로 바로 들어가야 한다(어떤 것은 키패드를 가진다). 만일 카드 판독기를 확신한다면 대부분의 경우 카드 판독 장치에 패스워드를 입력하는 것으로 충분하다. 그러나 더미 카드 판독기를 통해 패스워드를 가로채일 수 있다는 점도 명심해야 한다.

생체측정 기술은 측정 센서가 안전하고 암호가 제대로 출력된다면 훨씬 뛰어나다. 필자가 가장 좋아하는 방법은 홍채(iris) 스캔 기술이다. 이 기술은 최근 상업적으로 성공을 거두고 있는 방법이다. 망막(retinal) 스캔과 달리 홍채 스캔은 눈 앞의 고밀도 카메라를 통해 떨어진 거리에서 스캔돼 사용자의 감정을 자극하지 않는다.

현재 이용되는 검색 기준 중에는 홍채가 지문보다 훨씬 나은 생체측정법으로 알려져 있다. 이것은 적은 수의 오류를 기록하고 있다. 이 방법은 도용될 가능성이 있는 패스워드로 진

입하거나 패스워드를 기억할 필요가 없다. 생체측정의 비용은 일반적인 상업 애플리케이션용으로 수용할만큼 낮아지고 있다.

보안 관리

앞에서 언급했듯이 기술적 보안만으로는 적합한 보안을 적절한 비용으로 제공할 수 없다. 기술적 솔루션을 적용하기 전 당신은 2가지 분야의 보안 관리를 해결해야만 한다.

하나는 설계 및 적용 단계에서 관리하는 것이고, 다른 하나는 조작시 위험을 관리하는 것이다. 보안 구조에 대한 위험 평가는 많은 제품과 프로토콜이 사용되는 분산 환경에서 기술적으로 매우 복잡하다. 여기서 2가지 의문이 생긴다.

먼저 무엇이 가장 최악의 상황이며, 비지니스에서 수용할 수 있는 정도인가? 두번째는 보안 접근이 비용 효과적인가? 아주 대규모 조직을 제외하고는 이러한 질문 내용이 조직내에서 유용하지 않을 것이다. 만일 사용자들이 실수를 저질러 위험에 빠졌다면 필자는 외부 보안 상담을 이용할 것을 추천한다. 적어도 내부적 요인이 무엇인지를 입증할 수는 있을 것이다.

기술적 보안은 적절한 프로세스, 올바른 컨트롤 구조, 보안에 대한 교육된 태도에 의해 지원되지 않는다면 작동하지 않을 것이다. 운영 보안은 이러한 쟁점들을 어드레스한다. 많은 조직에는 잘 정비된 보안 운영 부서가 있기 때문에 필자는 이 문제를 세부적으로 논의하지 않을 것이다.

대부분의 조직에서는 사용자에게 최소 30일에 한번씩은 패스워드를 바꾸고, 당분간은 그 패스워드를 재사용하지 말도록 권장하고 있지만 이에 대해 필자는 걱정이 앞선다. 이러한 방법은 이론적으로 보안을 향상시킬 수는 있지만 실질적으로 보안을 약화시키는 방법이기 때문이다. 패스워드를 발견하는데 걸리는 시간이 짧을수록 패스워드는 취약하다고 볼 수 있다. 비록 이것이 기술적으로 사실일지라도 실제로는 그 방법으로 작동하지 않았다.

만일 토큰 기반 보안이 이용되지 않고 누군가 당신 패스워드를 원한다면 이 사람은 아마 하루도 못돼 쇼울더 서핑(shoulder surfing)하거나 키보드를 조작하여 보안을 침해할 것이다. 만일 패스워드가 강력한 힘에 의해 파손된다 하더라도 딕셔너리 어택(dictionary attack)을 통해 여러가지 경우를 시도함으로써 한시간 내에 당신의 패스워드를 알아낼 수 있을 것이다(딕셔너리 어택은 언어에서의 다양한 단어를 사용해 승인을 시도하는 것이다. 침입자는 숫자를 추가하는 것처럼 단어상에서의 다양한 변수를 시도할 것이다).

이러한 경우는 특히 패스워드가 자주 바뀌는 경우 일어나는데, 그 이유는 사람들이 패스워드를 기억하기 쉽게 만들기 때문이다.

주로 아이들의 이름이나 좋아하는 취미, 컴퓨터 기종 등을 활용하거나 다른 시스템의 패스워드를 다시 사용하는 경향이 있다.

만일 패스워드를 무작위로 사용할 수 밖에 없다면 사용자들은 패스워드 넣는 것을 포기하게 될 것이고, 패스워드를 발견하기는 더욱 어려워질 것이다.

필자는 보안이 중요한 상황에서는 토큰 기반 보안을 선호한다. 만일 이것이 여의치 않다면 무작위로 선택된 키들을 비정기적이나 연간으로 바꾸는 것을 좋아한다. 또 패스워드를 교체할 필요가 있을 때마다 바꿔주곤 한다. 이러한 접근은 쇼울더 서핑을 조금 노출시킬 수 있다. 그러나 주의를 기울이면 오히려 보안 및 이용성을 전체적으로 향상시킬 것이다. 이 예는 조작적 보안 프로시저의 중요성을 설명한 것이다.

예술의 시작

최근 안전한 분산시스템은 몇가지 다른 타입의 제품을 조합하기 원한다. SSL은 인터넷과 몇가지 인증을 위한 연결 암호를 제공한다. 제품군들은 커버로스나 RSA 기술에 기반해 안전한 미들웨어 환경을 이용할 수 있게 한다. 턱시도와 같은 몇가지 미들웨어 제품들은 보안 기능을 제공하며, 로터스 노츠와 같은 네트웍 기반 애플리케이션에도 보안 기능이 있다. 표준은 적어도 분산시스템의 메시징 부분을 보호하기 위해 시작됐고, 안전한 IIOP는 더 중요한 부분중 하나가 되고 있다. 인증 기능, 특히 싱글 사인-온은 최근 한계에 처했다. 많은 환경을 지원하기 위해 서버측에서 실행되는 싱글 사인-온 인증의 광범위한 사용이 있기 전까지 상당한 시간을 요하기 때문이다. 단기적으로 가장 안전한 접근 방법은 많은 키를 보유하고 있는 싱글 사인-온을 제공하는 스마트 카드나 PC 카드 기술을 사용하는 것이다. DC