이용 불안감 해결하는 열쇠

사이버 주식거래의 등장과 폭발적 증가는 단순히 기존의 판매망에 새로운 판매채널이 추가된 정도의 의미가 아니며, 또한 인터넷이라는 별도 영역에서의 또 다른 경쟁을 의미하는 것도 아니다. 증권회사가 인터넷의 전략적 활용이라는 선도적 자세를 갖고 어떻게 기존의 사업구조, 상품구조, 마케팅 구조, 영업구조, 전산시스템 구조의 개선을 추진하느냐가 디지털, 글로벌 경제 시대에 선도기업으로 앞서나갈 수 있는 요소가 될 것이다.

■ 성기윤/KAIST 인터넷상거래연구실

연재순서

사이버금융에서의 보안 문제

사이버금융은 전자상거래의 다른 어떠한 분야보다도 보안의 필요성이 강조되고 있는 분야이다. 금융산업의 특성상 실제적인 현금 가치가 이전되는 일이 빈번하게 발생하기 때문이다. 사이버금융은 기본적으로 열린 구조(open architecture)를 갖는 인터넷을 통하여 금융 거래 정보를 전달하기 때문에 거래과정에 있어서 사용자의 금융정보의 비밀 유지에 있어서 취약하다.

개인의 금융정보(계좌 번호, 신용카드 번호, 각종 비밀 번호 등)의 유출은 범죄로 악용될 수 있다. 현실에서의 금융거래에서도 개인의 계좌 번호나 신용카드 번호, 비밀 번호 등의 유출로 인한 금융 사고가 빈번히 발생하고 있다. 인터넷상에서의 금융거래가 위험한 것은 1차적으로 인터넷 자체의 보안의 취약성과 2차적으로 사이버금융 시스템 자체의 보안 취약성으로부터 기인한다.

인터넷의 보안 취약성

해킹 기술은 계속 발전하여 과거에 관리자의 실수나 OS의 실수를 악용하던 초보적인 수준에서 벗어나 Packet sniffing, IP spoofing 등의 고난도 해킹방법이 등장하게 되었다. Packet sniffing과 IP spoofing이 전자상거래와 관련하여 문제가 되는 것은 이 방법들이 OS의 오류나 전자지불시스템의 오류와 상관없이 인터넷의 고유한 구조를 이용한 것이기 때문이다.

Packet Sniffing은 인터넷에서 정보를 송수신하는 방법의 허점을 이용한 해킹 방법이다. 인터넷에서 정보를 송수신할 때 가장 기본이 되는 정보 단위인 패킷(packet)을 이더넷(ethernet) 케이블상에 무작위로 뿌린다. 즉, A라는 호스트가 B라는 호스트로 패

킷을 보내고 싶다면 호스트 A는 호스트 B와 배타적인 연결을 하는 것이 아니라 그 패킷을 이더넷에 뿌리는 것이다.

따라서 B호스트 외의 호스트에도 이 패킷이 도달할 가능성이 높지만 대부분의 호스트는 자신의 주소로 송신되고 있지 않은 패킷은 무시하며 오직 B호스트 만이 자신의 주소로 송신되고 있는 패킷을 받음으로써 통신이 이루어지는 것이다. 그런데, 호스트 B가 아닌 다른 호스트가 그 패킷을 무시하지 않고 그 내용을 해커에게 전달해 준다면 이야기가 달라진다.

아무리 네트워크 보안에 신경을 쓴 호스트라도 주변의 호스트가 스니핑을 위해 사용된다면 무력해질 수 밖에 없다. 스니퍼(sniffer)란 이와 같이 네트워크의 한 호스트에서 실행되어 그 주위를 지나다니는 패킷들을 엿보는 프로그램이다. 따라서 해커는 이더넷상에서 패킷을 수집하여 분석하면 사이버금융 서비스를 이용하여 금융거래를 하는 사용자들의 계좌정보를 알아낼 수도 있는 것이다.

IP spoofing은 해커가 머물러 있는, 또는 단순히 악용하고자 하는 호스트의 IP 어드레스를 바꾸어서 이를 통해 해킹을 하는 것이다. 가령 A 호스트와 B 호스트가 하드디스크를 공유하고 있는데 A 호스트와 B 호스트는 보안이 잘 되어서 해킹하기가 보통 어려운 것이 아니라고 하자. 그리고, 해커는 B 호스트 안에 있는 극비 문서를 훔쳐오고 싶다고 하자. 이때 해킹 방법은 다음과 같다.

우선 해커는 자신이 머물러 있는 호스트의 IP 어드레스를 B의 IP 어드레스로 위장을 한다. 위장을 하면 B의 호스트 화면에는 "duplicated IP address"라는 문장이 찍히게 되고 B 호스트는 네트워크 기능을 잠시 상실하게 된다. 이 때를 놓치지 않고 해커의 호스트는 A 호스트에게 자신이 진짜 B 호스트라는 정보를 보내어 A 호스트와 같이 하드디스크를 공유하도록 시도한다. 성공하게 되면 해커는 A 호스트의 하드디스크에 있는 극비 문서를 A 호스트나 B 호스트에 잠입하지 않고도 얻어낼 수 있게 된다.

사이버금융 서비스의 보안 취약성

사이버금융 서비스에서 고객은 일반적으로 서비스 가입을 하고 비밀번호 등을 이용하여 접속하며, 웹 페이지 상의 Form을 이용하여 자신의 지불 정보를 입력하게 된다. 이 두 가지 방식은 모두 보안상의 취약점을 가지고 있다.

Form을 이용하여 계좌 정보를 전송할 때 평문(cleartext)으로 보내거나 또는 SSL(Secure Socket Layer) 웹 서버를 이용하여 암호화된 정보를 보낼 수 있다. 물론, 평문으로 보내는 방식의 경우는 말할 것도 없고, 미국(U.S.A.) 외의 국가에서는 40 비트에 불과한 길이의 키로 암호화가 되므로 packet sniffing 등에 의해 다른 사람에게 유출되었을 때 보안의 유지가 힘들게 된다.

최근 제한적으로 이러한 규제가 풀렸다고는 하나 아직 웹 브라우저에 적용되지 않고 있다. 따라서 우리나라에서는 128비트 이상의 키 길이를 갖는 암호화 소프트웨어를 사용자들이 설치하도록 사이버금융 서비스가 이루어져 있다.

모든 사이버금융 서비스는 미리 등록한 사람만이 이용할 수 있도록 정책을 세우고, 사용할 사람은 미리 자신의 개인정보와 결제정보를 입력해 놓고 회원번호와 비밀번호를 이용해 거래를 할 수 있다. 이것을 초기의 등록 시 Form을 이용한다면 비록 결제정보가 네트워크를 타고 전송되는 횟수는 한 번으로 제한되기는 하지만, 이 때 정보가 누출되면 여전히 위험하다. 따라서 대부분의 국내의 사이버금융 서비스는 이 과정을 직접 지점 창구에 본인이 나와서 하도록 되어 있다.

또 한 사용자가 여러 개의 회원번호와 비밀번호를 가지는 것도 중대한 보안 허점이 될 수 있다. 보통 사용자들이 많은 회원번호와 비밀번호를 외울 수 없기 때문에 같은 회원번호와 같은 비밀번호를 사용하므로 한 시스템에서 비밀번호를 알게 되면 다른 시스템에서 도용 당할 수 있기 때문이다.

따라서 사이버은행 서비스에서는 기본적인 회원번호와 비밀번호 외에도 계좌비밀번호, 이체비밀번호를 따로 요구하고 있으며, 이 외에도 난수가 적혀있는 보안카드, OTP(One-Time Password), 인증서 등을 활용하고 있다.

앞에서 살펴본 Form 입력방식과 가입자 방식 외의 전자상거래 상의 보안 허점이 또 있다. 인터넷에서 쇼핑을 하고 대금결제를 할 때, 사용자의 계좌 또는 신용카드 정보가 그대로 상인에게 전달된다는 것이다. 상인은 고객의 계좌번호 또는 신용카드 번호와 그 비밀번호를 전달 받기 때문에 마음만 먹는다면 그 정보를 자신이 자신의 상품을 구매하는 데에 사용할 수도

있다. 따라서, 상인에게는 주문정보만 전달되고 계좌정보는 금융회사(은행, 신용카드사 등)에만 전달되어야 하는 것이다.

이것의 해결방법은 SET(Secure Electronic Transaction) 표준 안에서 Dual Signature라는 방법을 통하여 해결하고 있으며, KAIST에서 개발한 은행 계좌이체를 통한 직불 표준인 SDT(Secure Debit Transaction)에서도 Location Header를 이용한 Redirection 방법으로 해결하고 있다.

사이버금융에서 필요한 보안 기능

사이버금융 거래에서의 보안 문제로 인한 금융사고의 가능성이 높은 경우는 다음과 같다.

1. 대금결제나 이체 등의 금융 거래 시에 지불 관련 정보의 전송 중 제3자가 사용자의 계좌 정보를 빼 내어 자신이 계좌 주인인 것처럼 사용하는 경우
2. 위장된 상인 등에게 사기를 당하여 대금결제나 이체를 해 주는 경우
3. 금융 거래의 사실을 거래 당사자가 부인하는 경우
4. 금융기관의 데이터베이스 시스템에 저장된 개인의 중요한 정보를 해커가 침입하여 빼내는 경우

1-3의 경우는 암호화 방법의 응용한 네트워크 보안을 통한 방지가 가능하며, 4의 경우는 방화벽 등의 시스템 보안 도구를 사용하여 방지할 수 있다. 따라서 인터넷 보안은 크게 네트워크 보안과 시스템 보안 두 가지로 분류해 볼 수 있다.

네트워크 보안이라고 하면 네트워크를 타고 전송중인 자료의 보안을 이야기하며 암호화 방법론이나 각종 비밀번호의 사용으로 해결하고 있다.

데이터베이스의 중요한 자료는 암호화 하여 저장하거나 접속통제를 통하여 보호할 수 있다. 또한 시스템 보안을 통해 데이터베이스 자료의 보안이 가능하다.

시스템 보안이란 컴퓨터 시스템의 OS, 응용 프로그램, 서버 등의 보안 허점을 이용해 해커들이 침입해서 컴퓨터 시스템을 임의로 사용하거나 시스템의 기능을 마비시키거나 파괴하는 것, 시스템의 데이터베이스 안에 저장되어 있는 자료를 임의적으로 파괴, 수정하는 것을 방지하는 것을 말한다.

앞에서도 얘기했듯이 인터넷은 열린 구조이기 때문에 누구나 접속할 수 있다는 약점이 있다. 이러한 약점을 보완해 주기 위하여 방화벽(firewall)을 비롯한 여러 가지 시스템 보안 도구가 이용되고 있다.

지금까지 살펴본 인터넷의 보안 취약성, 현존 인터넷 지불 방식의 보안 취약성은 네트워크에서 전송되는 중 그리고 시스템에 저장된 후에 보완되어져야 한다.

네트워크에서 전송되는 중에는 암호화 방법에 의해 해결할 수 있고 시스템에 저장된 후에는 시스템 보안 도구들을 이용하여 해결될 수 있다.

그러면, 그 구체적인 해결방법들을 살펴보기 전에 먼저 이 취약성들을 극복하기 위하여 전자상거래 시스템 내에 구현해야 할 보안상의 기능들이 무엇인지 정의하도록 하자. 이것들은 기밀성(Confidentiality), 인증(Authentication), 무결성(Integrity), 부인방지(Nonrepudiation) 네 가지로 요약할 수 있다. 이들을 각각 설명하면 다음과 같다.

(1) 기밀성(Confidentiality)

기밀성은 전달내용을 제 3 자가 획득하지 못하도록 하는 것이다. 예를 들어 전자결제를 위하여 은행 계좌번호와 그 비밀번호를 인터넷을 통하여 상인에게 전달할 때 암호화 하여 전송함으로써 도청자가 packet sniffing 등에 의하여 그 내용을 얻어내더라도 풀지 못하도록 할 필요가 있다.

(2) 인증(Authentication)

인증은 정보를 보내오는 사람의 신원을 확인하는 것이다. 예를 들어, 상인의 입장에서 볼 때 어떤 고객이 상품의 구매대금으로 신용카드번호를 보내왔을 때 그 고객이 그 신용카드의 실제 소유자인지를 확인할 필요가 있는 것이다.

(3) 무결성(Integrity)

무결성은 정보전달 도중에 정보가 훼손되지 않았는지 확인하는 것이다. 예를 들어, 신용카드 회사의 입장에서 볼 때 은행 고객 갑이 "을에게 100만원을 지불하겠다"는 내용을 보내왔을 때 이 내용이 원래는 "병에게 100만원을 지불하겠다"는 등의 다른 내용이었던 것이 중간에 (이 경우, 아마도 을에 의해서) 변조된 것이 아닌지를 확인할 필요가 있다.

(4) 부인방지(Non-repudiation)

부인방지는 정보제공자가 정보제공 사실을 부인하는 것을 방지하는 것이다. 예를 들어, "갑에게 100만원을 신용카드로 지불하겠다"는 메시지를 보낸 을이 나중에 그런 메시지를 보낸 적이 없으며, 따라서 갑에게 100만원을 지불하지 못하겠다며 그 내용을 부인하는 것을 막을 필요가 있다.

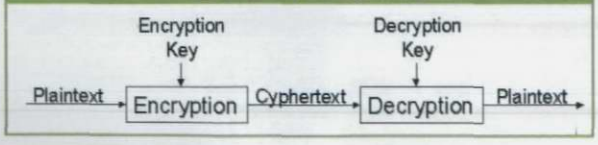
암호화 방법을 통한 네트워크 보안

네트워크 보안이란 네트워크상에서 전송 중에 있는 거래 정보의 보안을 의미한다. 보통은 암호화 방법과 각종 비밀번호를 함께 사용하여 이를 구현한다. 여기서는 특별히 암호화 방법론에 의한 네트워크 보안 구현에 관한 내용을 살펴보도록 하겠다.

암호화 알고리즘은 모두 키(Key)를 가지고 있다. 키는 매우 큰 숫자 중의 하나이다. 암호화와 복호화는 이 키를 이용하여 이루어지며, 키의 값을 제외하고는 모든 사용자가 동일한 암호화 및 복호화 알고리즘을 사용한다. 즉 알고리즘 자체는 공개되어 있고 키의 보안을 유지하는 것이 현대의 암호화 알고리즘의 특징이다. 키를 이용한 암호화와 복호화 과정을 그림으로 도시하면 〈그림 1〉과 같다.

키 기반 암호화 알고리즘은 크게 두 가지로 나누어 볼 수 있다. 하나는 대칭형 알고리즘(Symmetric Algorithm)이고 다른 하나는 공개키 알고리즘(Public-Key Algorithm, Asymmetric Algorithm)이다. 암호화 알고리즘은 아니지만 전달된 정보의 변경 여부(무결성)나 정보를 보낸 사람을 확인(인증)할 때 사용하는 것으로 메시지 다이제스트(Message Digest) 방법이 있다. 이들이 암호화 응용의 요소 기술이며 이들을 각각 설명하면 다음과 같다.

〈그림 1〉 암호화 키(key)의 개념



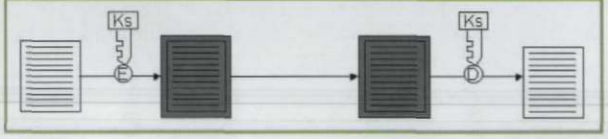
(1) 대칭형 알고리즘(Symmetric Algorithm, Secret-key Algorithm, Single-key Algorithm, One-key Algorithm, 비밀키 암호화 방식)

암호화 키로부터 복호화 키를 계산해 낼 수 있거나, 반대로 복호화 키로부터 암호화 키를 계산해 낼 수 있을 때 이 암호화 알고리즘을 대칭형 알고리즘이라고 부른다. 대부분의 대칭형 알고리즘에서는 암호화 키와 복호화 키가 동일하다. 동일한 암호화 키와 복호화 키를 Ks라고 했을 때, 암호화 및 복호화 과정을 그림으로 도시하면 〈그림 2〉와 같다.

이 방법의 장점은 암호화와 복호화가 빠르다는 점과, 다양한 암호화 기법이 개발되어 있다는 것이다. 그러나, 이 방법의 단점은 복수의 사용자가 관련되어 있을 때 키의 공유 문제가 발생한다는 것과 키 자체를 상대방에게 안전하게 보내는 것이 문제가 된다는 것이다.

예를 들어 A가 B에게 기밀성을 보장하여 문서를 전달하고 싶을 때 비밀키로 암호화 하여 보내면 된다. 그러나 B가 이미 같은 비밀키를 갖고 있지 않다면, A는 B에게 암호화된 문서 뿐 아니라 키 자체도 전달을 해 주어야 한다. 여기서 중간에 키 자체가 함께 가로채기를 당하면 문서의 기밀성은 보장되지 않게 된다. 대칭형 암호화 알고리즘으로는 DES, IDEA, RC2, RC5 등이 있고, 우리나라에서 개발 된 SEED라는 알고리즘이 있다. 금융감독원에서 사이버금융 거래의 보안성 심사를 해 줄 때에는 이 SEED 알고리즘이 적용되어야 한다.

〈그림 2〉 대칭형 암호화 방식



(2) 공개키 알고리즘(Public-key Algorithm, Asymmetric Algorithm, 비대칭형 암호화 방식)

이 방법에서는 암호화 키와 복호화 키가 서로 다르며, 또한 암호화 키로부터 복호화 키를 계산해낼 수 없다. 이 방법이 공개키 방법이라 불리는 이유는 암호화 키가 공개되어도 된다는 것 때문이다. 아무나 암호화 키를 이용하여 어떤 내용을 암호화할 수 있지만, 오직 해당 복호화 키를 가진 사람만이 그 암호문을 복호화할 수 있다. 이 때문에, 이 알고리즘에서는 암호화 키를 공개키(Public Key)라고 부르고, 복호화 키를 개인키(Private Key)라고 부르며, 대칭형 알고리즘에서 키를 비밀키(Secret Key)라고 부르는 것과 구별한다.

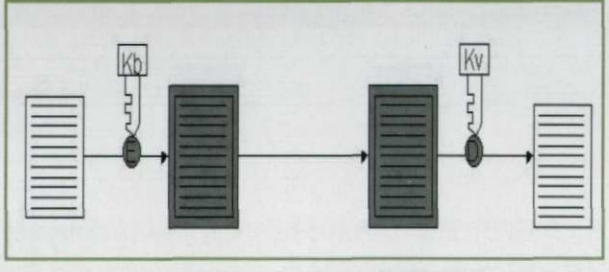
공개키를 KB, 개인키를 KV라고 표시했을 때, 암호화, 복호화 과정을 그림으로 도시하면 〈그림 3〉과 같다.

두개의 키 중에 하나는 공개키로 누구에게나 공개가 된다. 따라서 암호화 된 문서를 받는 상대방에게 전달을 한다. 또는 공개키 센터에 등록을 하거나 하는 방법으로 통신 상대방들에게 공개를 하게 된다. 그리고 하나는 개인키로 아무도 알 수 없도록 잘 보관을 한다.

그렇다면 A가 B에게 기밀성을 보장하면서 문서를 전달하고자 할 때에는 B의 공개키로 문서를 암호화하여 보낸다. 그렇게 되면 이 문서를 복호화할 수 있는 키는 B의 개인키 뿐이므로 B만이 이 문서를 볼 수 있게 된다. 이렇게 하여 대칭형 알고리즘에서 발생하던 키 교환의 문제는 해결이 된다.

이 방식의 장점은 키를 상대방에 보내는 것에 보안상의 허점이 없다는 점과 정보의 기밀 유지 이외에 다른 목적(무결성, 부인방지 등)으로도 사용될 수 있다는 점이지만 단점으로는 암호화, 복호화 속도가 대칭형 알고리즘에 비해 매우(약 1000배) 느리고 많은 양의 자료를 암호화, 복호화하기가 불편하다는 것이다. 공개키 암호화 알고리즘으로는 RSA, LUC, Diffie-Hellman, Elliptic Curve 등이 있다.

〈그림 3〉 공개키 암호화 알고리즘

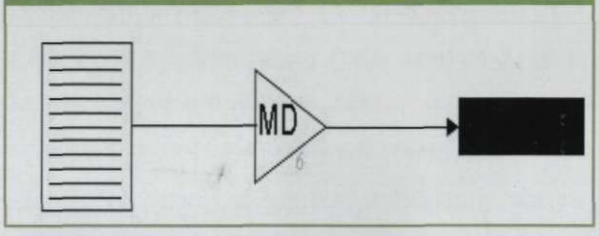


(3) 메시지 다이제스트(Message Digest)

앞에서 언급한 바와 같이 메시지 다이제스트는 암호화 방법은 아니다. 이것은 단방향 해쉬 함수를 이용하여 주어진 정보를 일정한 길이 내의 아주 큰 숫자(해쉬값)로 변환해 주는 것이다. 이 함수는 단방향이기 때문에 주어진 정보로부터 해쉬값을 만들어 낼 수는 있어도, 반대로 이 해쉬값으로부터 원래의 정보를 복구해낼 수는 없다. 다만, 정보와 함께 그 정보의 해쉬값을 받은 사람의 받은 정보의 해쉬값을 구한 후, 정보와 함께 전달된 해쉬값을 비교함으로써, 그 값이 같다면 정보의 전달 중에 정보가 변경되지 않았음을 (100%는 아니지만 거의 확실하게) 확인할 수 있으며, 만약 그 값이 다르다면 정보가 전달 중에 어떻게 든 변경되었음을 알 수 있다.

물론, 이 해쉬값은 암호화 알고리즘에 의해 암호화되어 전달되어야 한다. 그렇지 않다면, 정보를 중간에서 변조하는 사람이 정보를 변조한 후 그 변조된 정보의 해쉬값과 함께 보낼 수 있기 때문에 해쉬값이 제대로 기능하지 못하게 된다. 이 과정은 〈그림 4〉와 같다.

〈그림 4〉 메시지 다이제스트

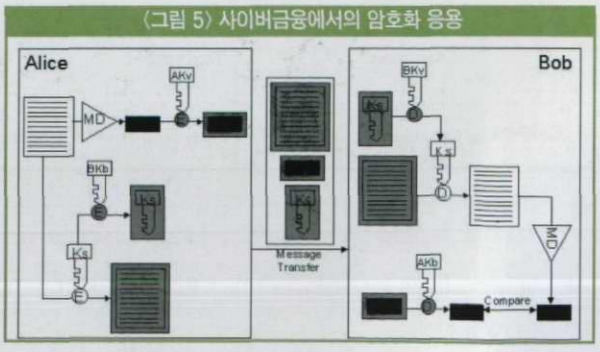


대표적인 메시지 다이제스트 알고리즘으로는 Senfru, CRC-32, CRC-16, MD2, MD4, MD5, SHA, Haval 등이 있다.

일반적으로 앞의 네 가지 보안기능(기밀성, 인증, 무결성, 부인방지)을 이루기 위해서는 전자서명(원문의 메시지 다이제스트를 송신자의 개인키로 암호화 한 것)을 만든 후, 원문과 전자서명을 임의의 비밀키를 이용하여 비밀키 암호화 방식으로 암호화하고 비밀키는 받는 사람의 공개키를 이용하여 공개키 암호화 방식으로 암호화하여 두 암호문을 함께 보내는 방법을 쓴다(〈그림 5〉 참조). 이렇게 하여 암호화를 통한 네트워크상에서의 전송중인 데이터의 보안이 이루어진다.

우리나라에서 사이버금융에 적용 가능한 암호화 소프트웨어 솔루션은 SoftForum사와 Initech사의 두 가지 제품이 있다. 이들은 금융감독원의 승인을 받은 제품으로써 국내에서 사이버

〈그림 5〉 사이버금융에서의 암호화 응용



금융 서비스를 이용하는 사용자들은 금융 서비스에 접속을 하면 이들 두 회사의 제품 중 하나를 다운로드 받게 되며 이후로는 웹 브라우저에서 제공하는 SSL이 아닌 이 제품의 암호화가 거래에 적용된다.

시스템 보안도구의 활용

인터넷을 중심으로 하는 시스템에서의 보안 취약성의 범주는 크게 시스템 자체 버그에서 오는 보안허점, 응용프로그램의 결함 등의 호스트 컴퓨터 내부의 보안취약성과 inetd port같은 네트워크에 연결된 부분에서의 보안 취약성, 그리고 이러한 호스트와 외부 네트워크 사이의 방화벽에서의 보안, 그리고 나머지 네트워크상에서의 보안취약성으로 나누어볼 수 있다.

(1) 결함탐지도구

이러한 보안 사항들은 주로 보안 취약성을 탐지해 내는 도구들을 사용해서 관리하게 되는데 이러한 도구들을 결함탐지도구라고 한다. 이러한 보안결함탐지 도구들은 상용 제품들 외에 인터넷 상의 쉐어웨어도 상당히 많다. 요즘의 추세는 이러한 도구들을 방화벽에 장착하여 방화벽시스템으로 종합적인 보안 시스템을 구축해 가는 상황이다.

인터넷 상에서 많이 사용되고 대표적인 결함탐지 도구들 중에서 각 범주별로 대표적인 것은 Crack, Cops, SATAN 등이 있으며, 보안결함탐지 도구에서 제공하는 주요한 보안 검사 기능은 다음과 같다.

● 계정 보안

- 패스워드 파일 검사: 패스워드 파일의 무결성 검사
- 그룹 파일 검사: 그룹 파일의 무결성 검사
- 패스워드 크래킹: 추측되기 쉬운 패스워드를 파악하여 사용 금지 조치

● 시스템 보안

- 사용자 검사: 사용자 디렉토리의 권한 및 소유권 검사
- 자동 수행 검사: 자동적으로 수행되는 파일들의 권한 및 소유권 검사
- 장치 파일 검사: 장치에 대한 접근의 제한 검사

● 네트워크 서비스 보안

- FTP 서비스 검사
- TFTP 서비스 검사
- 인터네트 서비스 관련 파일 검사
- MAIL 서비스 검사
- NFS 서비스 검사
- NIS 서비스 검사

● 파일 변경 검사

(2) 방화벽(Firewall)

인터넷에서의 방화벽(Firewall)이란 인터넷과 내부호스트(인트라넷) 사이, 즉 네트워크와 네트워크 사이에 위치해서 인증된 정보만이 전송될 수 있게 해주는 시스템을 의미하며 다음과 같은 특징을 가져야 한다.

● 모든 트래픽은 방화벽을 통해야 한다.

● 보안정책에 의해 규정된 인증된 트래픽만이 통과할 수 있다.

● 방화벽 자체는 보안성이 뛰어나야 한다.

방화벽은 자신의 네트워크 사이트들을 인터넷을 통하여 침입하는 해커들로부터 효과적으로 방어할 수 있을 뿐만 아니라, 인터넷 상에 존재하는 유용한 서비스의 이용 및 정보 획득을 용이하게 해줄 수 있어 가장 각광 받는 보안기술로 인정 받고 있다. 요즘의 방화벽은 위의 4가지 보안 검사 범주들을 통합적으로 모두 해결하는 통합보안시스템으로 발전하고 있다.

방화벽이라 하면 네트워크에 보안기능을 제공할 수 있는 라우터, 호스트 시스템 또는 이러한 시스템들의 집합 등만을 의미하는 것은 아니다. 차라리 방화벽이라고 하는 것은 보안이라는 목표에 다가가기 위한 접근방법이라고 할 수 있다. 방화벽은 허용할 서비스들이나 접근을 정의하는 보안정책을 구현할 수 있게 할 뿐만 아니라 네트워크의 구조, 라우터 및 호스트들, 그 외에 보안기법(정적인 패스워드를 대체할 수 있는 진보된 인증 등) 등을 이용한 보안정책의 구현을 의미한다.

방화벽 시스템의 주된 목표는 자신의 네트워크으로부터 다른 네트워크들의 접속을 제어하거나 외부로부터 접속을 규제하는 것이다. 모든 연결 행위를 방화벽 시스템을 통해서만 가능케 함으로써 이러한 모든 연결행위를 모니터링할 수 있게 된다. 방화벽 시스템은 라우터, PC, UNIX호스트 또는 이들의 집합체가 될 수도 있다. 이러한 집합체로서 네트워크 사이트나 서

브넷을 외부의 호스트들에 의해서 자주 사용되는 프로토콜 및 서비스들을 이용한 해커들의 침입으로부터 보호할 목적으로 구성될 수 있다.

방화벽 시스템은 주로 네트워크 사이트를 보호하기 위하여 인터넷과 연결하는 길목에 위치시키거나 이보다는 규모가 작은 서브넷들이나 호스트들을 보호하기 위하여 이들의 게이트웨이에 설치하기도 한다.

방화벽을 설치하는 이유 중에서 첫번째로 꼽을 수 있는 것은, 네트워크 사이트의 호스트들이 사용하고 있는 서비스들 중에 보안 취약점이 있는 NFS 나 NIS 같은 편리한 서비스를 계속적으로 안전하게 이용하면서 동시에 외부로부터 침입을 봉쇄하기 위함일 것이다. 즉, 방화벽이 네트워크 전체를 방어해 주고 그 안에서 자유롭게 시스템들을 사용하고자 하는 것이다.

이러한 방화벽이 설치되어 있지 않을 경우 해당 사이트의 호스트들은 전적으로 호스트 단위 보안에 의존하여야 하며 모든 호스트들을 상당 수준의 보안 상태를 유지하게 하여야 할 것이다. 그러나 호스트의 수가 많아지면 이러한 작업은 용이치 않게 되므로 보안상의 허점을 발생시키게 될 것이다. 방화벽은 이러한 상황에서 해당 사이트 내에 존재하는 많은 호스트들의 보안상태를 전반적으로 향상시켜 줄 방법인 것이다. 다음은 방화벽을 사용하여 얻을 수 있는 보안상의 주요 이점들이다.

- **보안상의 취약점을 갖고 있는 서비스들의 보호**
- **해당 사이트 시스템들에 대한 접속 제어**
- **집중화된 보안**
- **암호화를 통한 강화된 프라이버시**
- **네트워크 사용 또는 오용에 관한 통계 자료 축적**
- **네트워크 접속 정책 강화**

이상에서 언급한 사항들이 방화벽을 구축함으로 얻을 수 있는 보안상의 이점들이다. 그러나 완벽한 보안이란 불가능하다. 방화벽에 의한 보안도 예외가 될 수는 없다. 다음 열거하는 사항들이 방화벽을 설치함으로써 발생할 수 있는 사용자들이 느낄 수 있는 사용상의 불편과 보안상의 문제점들이다.

- **유용한 서비스들에 대한 제한된 접속**
- **뒷문(back door)들에 대한 무방비**
- **내부 해커에 대한 무방비**
- **바이러스 문제**
- **네트워크 성능 저하 문제**
- **기타 문제들(WWW, gopher, MBONE등 서비스 프로토콜 접속 문제)**

사용자 신원 확인 문제

기본적으로 인터넷은 상호 확인이 불가능하다고 할 수 있다. 그러나 금융 거래와 같이 신원 확인이 중요한 분야에서는 이에 대한 해결책이 절실한 상황이다. 국내의 사이버금융 서비스는 사용자는 반드시 금융기관 지점에 가서 사용자 등록을 신원 확인과 함께 하는 절차를 갖고 있다. 그리고 사용자는 복잡한 ID와 비밀번호를 갖게 된다. 그러나 금융기관의 입장에서는 ID와 비밀번호를 알고 있다고 해서 정말 그 사용자가 사용하고 있다고 믿기 어렵다.

금융사고에 대한 책임 문제 등이 무겁기 때문이다. 따라서 사이버은행의 예를 들면, 이미 사용하고 있는 계좌 비밀번호와 PC뱅킹, 폰뱅킹 등에서 이미 사용하고 있는 계좌이체비밀번호를 이중, 삼중의 신원 확인 도구로 제시를 요구하고 있다.

그러나 이러한 비밀번호들은 정적(static)이기 때문에 충분히 만족할 수 없다는 입장이다. 이러한 정적인 비밀번호 여러 개보다 동적(dynamic)인 비밀번호 한 개가 더 안전하다는 것이 일반적인 입장이다. 동적인 비밀번호의 응용 예로는 보안카드, OTP(One-Time Password)가 있다. 보안카드는 전화카드 크기의 카드에 30~35개의 난수가 적혀있다. 사용자는 계좌 이체 서비스를 사용할 때마다 랜덤하게 질문되는 번호에 해당하는 난수를 입력해야 한다. 사이버금융 시스템은 고객별로 보안카드의 내용을 저장하고 있으면서 비교를 해서 맞는지 틀리는지 검사한다.

OTP는 전자계산기와 같은 크기의 일종의 난수 발생기로서, 질문 된 난수를 입력하면 일정한 알고리즘에 의해 계산되어 나오는 다른 난수를 입력한다. 사이버금융 시스템은 OTP 알고리즘과 고객마다의 다른 키 값을 저장하고 있다가 자신이 낸 문제에 해당하는 OTP를 계산하여 비교해서 맞는지 틀리는지 검사한다. 보안카드는 비용이 적게 들고 이미 PC뱅킹, 폰뱅킹에 적용되고 있으나, 아무래도 30여 가지의 경우의 수 뿐이므로

보안성이 떨어질 우려가 있다.

OTP는 백만가지 이상의 경우의 수를 갖기 때문에 보안성이 우수하다고 할 수 있지만, OTP카드의 가격이 비싸 사용자에게 배포가 어렵다는 단점이 있다. 따라서 일반 개인 사용자들은 보안카드를, 거래 규모가 아주 크거나 기업 사용자들은 OTP를 사용하도록 하는 것이 효과적인 방법이라고 할 수 있다.

또 다른 사용자 확인 방법으로는 인증서를 사용하는 방법이 있다. 전자서명의 기능을 설명할 때 인증, 무결성, 부인방지를 확인하기 위해 중요한 역할을 한 것이 정보를 확인하는 데 쓰인 공개키였다. 전자서명을 받은 사람은 그것을 보낸 사람의 공개키로 복호화함으로써 그러한 내용들을 확인할 수 있었던 것이다. 그런데, 상호 확인이 원칙적으로 불가능한 가상공간 내에서 다른 사람의 공개키를 어떻게 알 수 있는가 하는 것이 문제가 된다. 미리 상대방의 공개키를 모르는 상태에서, 상대방이라고 자처하는 사람이 전자서명을 보내면서 자신의 공개키를 함께 보내온다면(그 사람이 가짜이고, 또 똑똑하다면) 그 전자서명은 함께 보내온 공개키에 맞게 생성될 것이다.

보내는 가짜의 입장에서는 하나의 공개키, 개인키 짝을 생성한 후 그 개인키를 이용하여 전자서명을 만들고 그 전자서명과 함께 앞에서 준비한 공개키를 보내면 되기 때문이다. 따라서, 상대방의 공개키가 그 사람의 것이 맞다는 것을 확인하기 위한 방법이 마련되어야 한다.

이를 위하여 제시된 것이 인증기관(CA, Certificate Authority)이다. 인증기관은 한 사람의 공개키를 자신의 개인키로 암호화함으로써 그 사람의 공개키를 인증한다. 물론, 인증기관은 인증하기 전에 그 사람을 실제로 확인한 후 그 사람이 제시한 공개키를 인증한다. 인증기관의 개인키로 암호화한 공개키를 전자인증서(Digital Certificate)라고 부른다.

이 전자인증서를 받은 사람은 인증기관의 공개키로 전자인증서를 풀어서 나온 공개키를 상대방의 공개키라고 믿을 수 있다. 왜냐하면, 그러한 전자인증서를 만들 수 있는 사람은 그 개인키를 알고 있는 인증기관뿐이며, 인증기관은 믿을 수 있기 때문이다. 물론, 이 확인을 위한 전제조건은 그 인증기관의 공개키는 미리 널리 유포되어 모두가 알고 있어야 하며, 그 공개키는 누군가에 의해 조작되어 있지 않아야 한다. 그렇지 않다면 전자인증서의 확인은 틀린 것이 되기 때문이다.

이와 같은 CA 계층구조는 아직 없다. 그러나 이것은 전자상거래의 기간구조(Infrastructure)로 언젠가는 반드시 마련되어야 할 구조이다. 우리나라는 1999년 7월 전자서명법이 시행되면서 KISA(한국정보보호센터)에서 CA계층구조에 대한 안을 제시하고 있다. 은행의 상위 CA는 금융결제원이, 증권회사의 상위 CA는 증권전산이 운영하도록 제시되어 있다.

인증서는 암호화를 이용한다는 점에서 보안성이 우수하다고 인정되지만 필요한 기간구조가 방대하고 비용도 비싸며, 사용방법이 어렵고 불편하기 때문에 사용자에 대한 교육이 많이 필요하며, 모든 개인 사용자가 공개키 암호화 알고리즘이 적용되기 때문에 속도가 느리다. 무엇보다도, 인증서를 간편하게 저장할 수단이 없다면 사용자가 여러 다른 PC에서 서비스를 사용할 수 없고, 현재 인증서를 발급 받았던 PC에서만 서비스를 사용할 수 있다는 단점이 있다.

현재 플로피 디스크에 저장하는 안이 제시되고 있지만 안전성이 떨어지고 사용자의 교육이 어렵기 때문에 거의 사용되지 않고 있으며, PC를 이동할 경우 다시 인증서를 발급 받도록 하고 있다. 이 것은 사실 비밀번호만 알면 누구나 인증서를 발급 받을 수 있다는 면에서 오히려 보안성이 떨어진다고 할 수 있다. 가장 효과적이 인증서 저장 수단으로 생각되는 것은 IC카드이다. 우리나라의 사이버은행 서비스들도 인증서를 적용하고 있지만 사용자들의 불편이 많은 실정이다.

결론

이상으로 사이버금융 서비스에 있어서 보안 취약성과 그에 대한 대책에 대해서 살펴보았다. 사이버금융 서비스는 일반 고객들에게 편리한 서비스를 제공하면서 동시에 강력한 보안을 유지해야 하는 양면성을 가지고 있다. 보안성의 강화는 필연적으로 사용상의 불편함과 속도의 저하를 가져올 수 밖에 없다. 이는 사이버금융 서비스의 목표와 상치되는 것이다. 그러나 안전하지 않다면 아무도 그 서비스를 이용하지 않을 것이다.

앞으로 기술의 발전과 함께 좀 더 편리한 보안 도구들이 개발될 것이다. 그 때 까지는 사이버금융 서비스 업체는 현재의 기술들을 최대한 보안성이 강화되면서 효율적으로 편리성과 속도를 갖는 시스템으로 설계하여야 할 것이다.