방화벽을 위한 인증 시스템의 설계

문 순 일°, 이 필 중
포항공과대학교 전자전기공학과

Design of an Authentication System for Firewalls

Sun Il Mun , Pil Joong Lee
Dept. of Electrical and Electronic Engineering, POSTECH

요약

최근에 사설망의 네트웍 보안 수준을 향상시키기 위한 방법으로 많은 주목을 받고 있는 방화벽에 대해서 알아본다. 그리고 현재의 방화벽에서 사용자 인증과 네트웍 주소 인증 방법에 대한 문제점을 알아보고, 이를 해결하기 위한 인증 시스템을 설계하고자 한다. 사용자 인증을 위해서는 스마트카드를 이용하면서 공개키 암호 알고리듬을 이용하는 강한 사용자 인증 프로토콜을 설계하였으며, 네트웍 주소 인증을 위한 암호학적 기법을 제안했다. 또한 신뢰도에 따른 각 호스트에 적용할 접근 제어 정책에 대해서 알아본다.

1 서론

최근에 사람들이 네트웍을 이용해서 많은 유용한 서비스들을 받으면서부터, 네트웍에 대한 중요성과 그 가치가 높게 평가되고 있다. 한편 네트웍 때문에 발생하는 보안 사고로 인한 사용자들의 피해가 늘고 있다. 이런 네트웍 보안 사고를 줄이기 위해서 많은 노력들이 이루어지고 있다. 그러나 관리하고자 하는 사설망의 규모가 매우 큰 경우에, 사설망 내부에 있는 모든 호스트들의 보안을 항상 시키기란 그렇게 쉬운 일만은 아니다. 전반적으로 보안을 강화시켰다고 하더라도, 몇몇 보안이 취약한 호스트가 공격을 당했다면 그 주위의 호스트들도 연쇄적으로 비교적 쉽게 공격을 당할 수 있다. 따라서 보호하고자 하는 사설망 전체의 보안 정도를 일정 수준 이상으로 높이는 것이 매우 중요하다. 이런 이유로 해서 등장하게된 방화벽은 사설망을 외부의 공중망으로부터 안전하게 보호하는데 아주 효과적인 시스템이다.

방화벽[1][2]은 보호하고자 하는 사설망을 외부의 공중망과 차단을 시킨 후, 외부와 접속할 수 있는 통로를 오직 하나로만 제한을 한다. 그리고 이 통로에 보안 장치를 마련해서 공중망으로부터 시도되는 공격들을 차단하게 된다. 그러나 현재의 방화벽 기능만으로는 내부망을 완벽하게 보호하기에는 무리가 있다. 가장 큰 문제점으로는 현재의 방화벽에서는 인증 기능이 미흡하다는 것이다. 방화벽에서 필요로 하는 인증에는 사용자 인증과 네트웍 주소 인증이 있다. 현재 간단한 패스워드만으로 사용자를 인증하는 방법은 결코 안전하지 못하며[3], 단순한 방법으로 네트웍 주소를 인증하는 방법도 공격자가 네트웍 주소를 속일 수 있기 때문에[4] 보다 확실하게 네트웍 주소를 인증하는 방법이 필요하다.

본 논문에서는 안전한 방화벽을 구축하기 위해서 사용자 인증과 네트웍 주소 인증을 강화하고자 한다. 사용자 인증을 위해서는 스마트카드를 사용하며, 암호학적 기법을 이용한 강한 사용자 인증 프로토콜을 설계한다. 또한 네트웍 주소 인증을 위해서도 암호학적 기법을 사용하고자 한다.

다음의 2절에서는 방화벽의 기본 개념과 구성 요소, 그리고 기능에 대해서 설명하며, 3절에서는 본

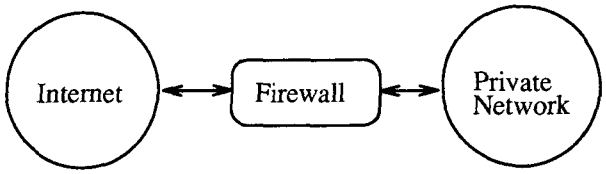

그림 1: 방화벽의 기본 개념

논문에서 고려하는 네트웍 모델과 인증 시스템의 운영에 대해서 알아본다. 4절에서는 공개키 암호 알고리듬을 사용해서 사용자들을 인증하기 위한 인증 프로토콜을 설계한다. 5절에서는 각 호스트의 신뢰도에 따른 접근 제어 정책에 대해 알아보고, 네트웍 주소를 속이는 공격을 막기 위한 네트웍 주소의 인증 방법을 제시한다.

2 방화벽

방화벽(Firewall)은 그림 1에서와 같이 사설망(Private Network)과 외부 공중망(Internet)을 연결하는 유일한 통로이며, 방화벽을 거치지 않고 사설망과 공중망을 직접 연결할 수는 없다. 방화벽은 사설망으로 향하는 모든 서비스들을 통제함으로써 사설망의 보안 수준을 전체적으로 높일 수 있다.

2.1 구성 요소

방화벽[1]은 다양한 방법으로 구현될 수 있으며, 그 기능과 보안의 정도가 각각 다르다. 방화벽은 다음과 같은 기본적인 구성 요소로 이루어 진다.

- 차폐 라우터(Screening Router)
  차폐 라우터는 방화벽의 기본적인 구성 요소로 패킷 필터링[5]을 수행한다. 즉 모든 데이타 패킷의 헤더에 있는 정보인 패킷의 발신지 주소와 목적지 주소, 그리고 응용 서비스의 종류를 가지고 패킷을 통과시킬 것인지 막을 것인지를 판단한다. 이는 일반 호스트[6]나 상용 라우터[7]로 구현할 수 있다.

- 요새 호스트(Bastion Host)
  요새 호스트는 사설망과 외부 공중망을 연결시켜주는 유일한 호스트이다. 요새 호스트는 외부에 노출이 되어 있으므로 보안을 매우 강화시켜서 운영하여야 한다. 이를 위해서 필요한 최소한의 서비스만을 제공하며, 불필요한 사용자 계정은 가지고 있지 않아야 한다. 요새 호스트에서는 사용자 인증이나 사설망으로의 접근 제어, 그리고 모든 데이타 흐름을 기록하는 로깅 등의 보안 서비스를 수행한다.

- 응용 게이트웨이(Application gateway or Proxy gateway)
  응용 게이트웨이는 주로 요새 호스트에서 운영되는 응용 프로그램들이다. 방화벽이라는 특수한 환경에서 사용자들이 사설망과 공중망을 연결하는 네트웍 서비스를 받기 위해서는 응용 게이트웨이의 개발이 필수적이다.

2.2 방화벽의 기능

방화벽은 2.1절에서의 기본적인 구성 요소를 가지고 다음의 기능들을 수행한다.

- 패킷의 발신지 주소, 목적지 주소, 응용 서비스에 근거한 패킷 필터링
- 네트웍 주소와 사용자에 근거한 접근 제어
- 사설망에 접근하고자 하는 사용자의 인증
- 모니터링과 로깅

2.3 강한 인증의 필요성

방화벽에서는 기본적으로 네트웍 서비스를 요청하는 호스트의 네트웍 주소를 가지고, 서비스를 허락할 것인지 거절할 것인지를 판단한다. 그러나 공격자는 자신의 패킷을 조작해서 패킷 필터링을 하는 라우터를 속일 수 있기 때문에[4], 보다 확실한 방법으로 네트웍 주소를 확인하는 것이 필요하다.

방화벽에서는 주로 공중망의 공격자들에 대한 방어만을 생각하기 때문에, 사설망 내부의 사용자들에 의한 보안 문제에 대한 고려는 많이 하지 않는다. 사설망 내부에서 사용자를 인증할 때에 기존의 간단한 패스워드만을 사용하기 때문에 내부 사용자에 의해서 많은 보안 문제가 발생할 수 있다. 또한 내부 사용자가 공중망으로 나가서 사설망의 네트웍 서비스를 받고자 할 때에 먼저 방화벽에서 사용자 인증을 수행하는데, 이때에도 암호화되지 않은 평문으로 사용자의 패스워드를 전달함으로서 사용자의 패스워드가 쉽게 노출될 수 있다.

따라서 이런 문제들을 해결하기 위해서는 암호학적 방법을 이용한 강한 인증이 필요하다.

3 방화벽을 위한 인증 시스템의 운영

먼저 3.1절에서는 본 논문에서 고려하는 네트웍 모델에 대해서 알아보고, 3.2절에서는 방화벽을 위한 인증 시스템의 운영에 대해서 알아본다. 본 논문에서는 강한 사용자 인증을 위해서 공개키 암호 알고리듬을 이용한다. 또한 사설망 내부의 사용자들은 강한 인증을 수행할 수 있도록 스마트카드를 사용한다.

3.1 네트웍 모델

그림 2는 본 논문에서 설계하고자 하는 인증 시스템을 위한 네트웍 모델이다. 사설망(Private Network)을 보호하기 위한 방화벽을 설치해서 공중망(Internet)으로부터의 공격을 차단한다. 방화벽의 모델로는 두 장의 네트웍 인터페이스를 갖는 요새 호스트(Bastion Host B)를 사용하는 Dual-homed 게이트웨이[1]를 가정한다. 요새 호스트 B는 공중망과 사설망을 연결하는 유일한 통로이며, 패킷 필터링과 접근 제어, 그리고 사용자를 위한 응용 서비스 등을 제공한다.

여기에서 사용자들과 각 호스트들은 그 환경에 따라서 각각 다음과 같이 나누어 진다. 먼저 사용자들은

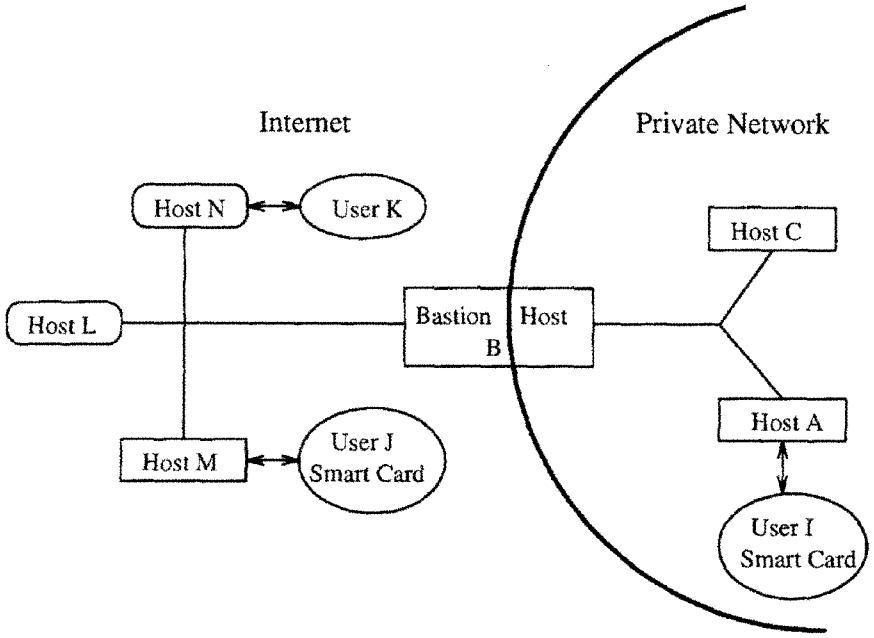

그림 2: 네트웍 모델

- 스마트카드를 소유하고 있는 내부 사용자(User I, User J)
- 스마트카드를 소유하지 못한 외부 사용자(User K)

로 나누어진다. 또한 인증 시스템을 운영하기 위한 조건들(응용 프로그램들과 카드 reader의 유무 등)에 따라서 호스트들은

- 조건을 모두 갖춘 내부 호스트(Host A, Host C)
- 조건을 모두 갖춘 외부 호스트(Host M)
- 조건을 갖추지 못한 외부 호스트(Host L, Host N)

로 나누어 진다. Host M은 사설망 외부에서 사설망의 네트웍 서비스를 받고자 하는 내부 사용자들을 위해서 특별히 마련된 호스트이다.

3.2 인증 시스템의 운영

- 사용자 및 호스트 등록
  시스템을 사용할 권리가 있는 사람은 등록기관(CA, Certificate Authority)에서 자신의 신분을 밝히고 사용자 등록을 신청한다. CA에서는 개인의 신분 확인을 한 후 그 사람에게 스마트카드를 발급해 준다. 각 개인의 스마트카드에는 사용자 I에 대해서 사용자의 이름(I), 공개키(P_I), 유효

기간($lifetime$)과 기타 정보들이 CA의 비공개키(S_{CA})로 서명된 신분 인증서와 사용자의 비공개키(S_I)가 들어 있다.

$$CertI = S_{CA}\{I, P_I, lifetime, etc, ...\}$$

사용자와 마찬가지로 각 호스트들도 CA로부터 등록된 호스트임을 인증하는 인증서를 받아야 한다. 새로운 호스트를 도입하면 담당자는 각 호스트에 대해서 CA로부터 인증서를 받아야 하며, 호스트의 비공개키는 절대 안전한 영역에 보관하여야 한다.

- 각 호스트들의 사용자 관리
  각 호스트들은 각 사용자들에 대해서 독립적으로 권한을 부여하고 관리를 하도록 한다. 각 사용자들은 자신이 원하는 호스트로부터 서비스를 받기 위해서 각 호스트에 자신의 이름을 등록시켜야 한다. 각 호스트는 사용자의 서비스 요청을 받은 후 사용자와 상호 신분 인증을 거친 후 정당한 사용자에 대해서 별도로 자신이 관리하고 있는 DB를 확인한 후 사용자에게 미리 허락된 서비스만을 제공한다. 특히 가장 중요한 호스트인 요새 호스트에는 각각의 호스트들이나 사용자들에 대해서 제공할 서비스들에 대해서 정확하게 관리를 해야 한다. 이에 대해서는 5절에서 다시 자세하게 다루기로 한다.

4 사용자 인증 프로토콜의 설계

사용자의 신분 인증을 위해서 공개키 암호 알고리듬을 사용하며, 인증 서버의 도움이 필요하지 않는 상호 인증 프로토콜을 설계한다. 여기서 설계한 인증 프로토콜은 국제 표준 문서인 IS9798-3[8]에서 제안된 프로토콜을 응용한 것이다. 대표적인 서비스로 로그인을 위한 인증 과정으로 인증 프로토콜을 설명한다.

4.1 Local 로그인

등록을 마친 사용자가 시스템을 이용하고자 할때 가장 먼저 local 호스트에 로그인을 해야한다. 이는 그림 2에서 사용자 I와 호스트 A 또는 사용자 J와 호스트 M 사이에서 이루어지는 상호 인증 프로토콜이다. 이때 사용자는 스마트카드를 이용해서 local 호스트와 상호 인증을 수행하게 된다. 먼저 사용자는 자신의 스마트카드를 호스트의 카드 reader에 삽입하고, 카드의 PIN을 입력하여 카드로부터 소유주임을 인증받는다. 스마트카드는 소유주 인증을 끝낸 후 호스트와 상호 인증을 시작한다.

다음의 인증 프로토콜에서 실체 K의 비공개키는 S_K이며, 공개키는 P_K이다. $S_K\{D\}$는 실체 K가 자신의 비공개키로 데이타 D를 서명한 것이고, 다른 임의의 실체는 실체 K의 공개키 P_K로 이를 검증할 수 있다. $h(D)$는 데이타 D를 해쉬 함수에 적용한 것이다.

1. $I \Longrightarrow A : N_I, I$
2. $I \Longleftarrow A : CertA, S_A\{N_A, N_I, I\}$
3. $I \Longrightarrow A : CertI, S_I\{N_I, N_A, A\}$
4. $I \Longleftarrow A : P_I\{K_{IA}\}, S_A\{h(K_{IA}, N_A, N_I)\}$

단계1에서 사용자(스마트카드를 의미함)는 사용자의 이름 I와 재생 공격[9]을 막기 위한 난수 N_I를 호스트에게 보낸다. 단계2에서 호스트 A는 단계1에서 받은 N_I와 자신의 난수 N_A를 자신의 비공개키 S_A로 서명해서 자신의 인증서 $CertA$와 함께 사용자에게 보낸다. 사용자는 $S_A\{N_A, N_I, I\}$의 검증을 통해서 호스트 A가 등록된 호스트임을 인증한다. 단계3에서 사용자는 자신의 비공개키 S_I로 서명한 $S_I\{N_I, N_A, A\}$를 자신의 인증서와 함께 호스트 A에게 보내고, 호스트 A는 이를 검증해서 사용자를 인증한다. 호스트 A는 사용자 I의 신원을 확인한 후 자신의 DB에 등록되어 있는 사용자면 로그인을 허락한다.

상호 신분 인증이 끝난 후 호스트 A와 사용자 I는 세션키 K_{IA}를 공유하게 된다. 세션키 K_{IA}는 추후에 사용자 I와 호스트 A사이에 인증이 필요한 경우(예를 들면, 사용자가 다른 호스트에 remote 로그인을 할 경우)에 처음보다 간단하게 상호 인증을 수행하고자 할 때나, 사용자와 호스트 사이에 비밀 정보를 주고 받기 위한 것이다.

4.2 사설망 내부에서의 Remote 로그인

Local 호스트에 로그인한 사용자가 다른 호스트로부터 서비스를 받기 위해서 인증을 받고자 할 때 다음의 인증 과정을 수행한다. 그림 2에서 사용자 인증 과정을 거쳐서 호스트 A에 로그인한 사용자 I가 호스트 C에 remote 로그인 하고자 할 때 이루어 지는 인증 프로토콜이다. 여기서는 사용자 I와 호스트 A 사이의 상호 인증과 사용자 I와 호스트 C 사이의 상호 인증이 동시에 이루어 진다.

1. $I \Longrightarrow A : N_{I1}, N_{I2}$
2. $A \Longrightarrow C : I, N_{I2}$
3. $A \Longleftarrow C : CertC, S_C\{N_C, N_{I2}, I\}$
4. $I \Longleftarrow A : CertC, S_C\{N_C, N_{I2}, I\}, K_{IA}(N_A \oplus K_{IA}(N_{I1})), N_A$
5. $I \Longrightarrow A : S_I\{N_{I2}, N_C, C\}, K_{IA}(N_{I1} \oplus K_{IA}(N_A))$
6. $A \Longrightarrow C : CertI, S_I\{N_{I2}, N_C, C\}$
7. $I \Longleftarrow A \Longleftarrow C : P_I\{K_{IC}, K_{AC}\}, S_C\{h(K_{IC}, K_{AC}, N_C, N_{I2})\}$
8. $I \Longrightarrow A : K_{IA}\{K_{AC}\}, K_{AC}(N_A \oplus K_{AC}(N_A))$

호스트 A와 사용자 I는 4.1절의 local 로그인 과정에서 생성한 세션키 K_{IA}를 DES와 같은 대칭키 알고리듬의 입력키로 사용해서 간단하게 상호 인증을 수행한다. 사용자 I가 난수 N_{I1}을 호스트 A에게 전송하면, 호스트 A는 K_{IA}를 입력키로 $K_{IA}(N_A \oplus K_{IA}(N_{I1}))$ 를 만들어서 사용자에게 전송한다. 이것은 known-plaintext atack을 막기 위한 것이다. 사용자는 이 값을 확인해서 호스트 A가 세션키 K_{IA}를 공유하고 있음을 확인하고 호스트 A를 인증하게 된다. 마찬가지로 호스트 A는 자신의 난수 N_A를 이용해서 사용자 I를 인증할 수 있다.

사용자 I와 호스트 C는 각각의 난수인 N_{I2}와 N_C를 이용해서 4.1절에서 이루어진 상호 인증과 동일한 과정을 통해서 상호 인증을 수행한다. 단계7에서 호스트 C는 사용자 I를 위한 세션키 K_{IC}와 호

스트 A를 위한 세션키 K_{AC}를 만들어서 사용자에게 전송한다. 사용자 I와 호스트 C는 세션키 K_{IC}를 공유하게 되며, 추후에 상호 인증을 간단하게 수행하기 위해서 사용하거나, 사용자 I와 호스트 C 사이의 비밀 채널 형성에 이용할 수 있다. 마찬가지로 호스트 A는 사용자 I를 통해서 호스트 C와 세션키 K_{AC}를 공유하게 되며, 이 세션키로 호스트 A와 호스트 C 사이에 비밀 채널을 형성할 수 있다.

4.3 방화벽을 통한 Remote 로그인

그림 2에서 사설망 외부의 호스트 M에 사용자 인증 과정을 거쳐서 로그인한 사용자 J가 사설망 내부의 호스트 C에 remote 로그인을 하고자 할 때 이루어지는 인증이다. 사용자 J가 호스트 C로부터 사용자 인증을 받기 위해서는 먼저 호스트 M이 다음의 5절에서 설명할 접근 제어와 네트웍 주소 인증을 통과해야만 한다. 요새 호스트 B는 먼저 호스트 M과 사용자 J가 호스트 C로부터 서비스를 받을 수 있는지를 확인한 후, 서비스가 허가되어 있는 경우에는 접속을 허락한다. 그런 후에는 단순하게 호스트 M과 호스트 C를 연결만 해줄뿐 인증 프로토콜에는 직접 참여하지 않는다.

사용자 I와 호스트 M은 호스트 C와 4.2절에서의 remote 로그인과 동일한 사용자 인증 프로토콜을 수행한다. 기존의 방화벽에서는 사용자 J가 호스트 C로 remote 로그인 하기 위해서는 먼저 요새 호스트 B에게 별도의 사용자 인증을 거친 후, 다시 한번 호스트 C에 사용자 인증을 받아야만 했다. 그러나 여기서는 사용자가 요새 호스트의 접근 제어를 받은 후, 호스트 C에게 직접 단 한번만의 사용자 인증을 받으면 된다.

5 네트웍 주소 인증을 통한 접근 제어

5.1절에서는 먼저 공중망에 속해 있는 각 호스트들의 위치에 따라서 각각 다른 수준의 네트웍 서비스를 제공하기 위한 접근 제어 정책에 대해서 알아본다. 5.2절에서는 보다 확실한 네트웍 주소의 인증이 필요한 호스트들을 위해서 암호학적 기법을 이용한 안전한 네트웍 주소 인증 방법을 제시한다.

5.1 접근제어 정책

여기서 접근 제어 정책이라 함은 요새 호스트 B를 통해서 사설망과 공중망을 연결하는 네트웍 서비스를 받고자 할 때에, 각각의 신뢰도에 따라서 제공 받을 수 있는 서비스의 종류가 제한됨을 말한다. 요새 호스트 B는 그림 2에서 호스트 A, C, M, N, L에 대해서 다음과 같은 신뢰도의 차이를 둔다.

$$host\ A > host\ C > host\ M > Host\ N > host\ L$$

기본적으로 사설망 내부의 호스트 A와 C는 외부의 호스트 M, N, L에 비해서 신뢰도가 높다. 따라서 내부의 호스트들이 공중망으로 나갈 때는 단순한 방법으로 인증을 수행하며 많은 제약을 받지 않는데 비해서, 공중망의 호스트들이 사설망 내부로 향하고자 할 때에는 엄격한 접근 제어와 사용자 인증을 수행하며 제공받는 서비스에 대해서도 제약이 많다. 그러면 각 호스트들이 가지는 신뢰도에 따라서 적용할 수 있는 접근 제어 정책에 대해서 알아보자.

- 호스트 A
  사설망 내부의 호스트 A는 최고의 신뢰도를 가진다. 따라서 호스트 A에 로그인 한 사용자는 아무런 제약없이 모든 네트웍 서비스를 받을 수 있다.

- 호스트 C
  호스트 C는 내부의 호스트이지만 약간의 서비스 제한을 받도록 한다. 즉 호스트 C에서는 내부의 중요한 정보를 외부의 공중망으로 유출할 수 없도록 한다.

- 호스트 M
  호스트 M은 사설망 외부에 있는 호스트들 중에서 가장 신뢰성 있는 호스트이며, 3.1절에서 언급한 인증 프로토콜을 수행할 수 있는 모든 준비가 갖추어진 호스트이다. 또한 호스트 M은 내부 호스트 C 다음으로 높은 신뢰도를 가지며, 5.2절에서 설명할 안전한 네트웍 주소 인증을 수행할 수 있다. 호스트 M에 로그인한 사용자 J가 사설망 내부의 호스트 A나 호스트 C로 사용자 인증을 받기를 원할 때 요새 호스트 B로부터 안전한 네트웍 주소 인증을 거쳐서 사용자 인증을 받을 수 있다. 사용자 인증을 받은 후에는 내부 호스트들과 동일한 서비스를 받을 수 있다.

- 호스트 N
  호스트 N은 사설망과는 특별한 관계가 없는 평범한 공중망의 호스트이다. 호스트 N에서는 인증 시스템을 이용할 수 없기 때문에 만약에 사용자 K가 스마트카드를 가지고 있다고 해도 사용할 수가 없다. 또한 신뢰도가 낮은 호스트이기 때문에 호스트 N에서는 공공 서비스(예를 들면, anonymous ftp 또는 WWW service 등)만 받을 수 있으며, 단순한 네트웍 주소 인증 방식으로 요새 호스트 B에게 인증을 받는다.

- 호스트 L
  호스트 L은 신뢰도가 전혀 없는 호스트로 아무런 서비스를 받지 못한다. 오히려 평소에 아주 위험하다고 판단되는 호스트로 요새 호스트로부터 모든 서비스를 거절당해서 사설망에 접근을 할 수 없다.

5.2 안전한 네트웍 주소 인증

5.1절에서 설명한 것과 같이 사설망 외부의 호스트들은 요새 호스트 B로부터 서로 다른 수준의 신뢰를 받으며, 요새 호스트는 각 호스트로부터의 서비스 요청을 받으면 그들의 네트웍 주소에 근거해서 서비스를 허가할지 거절할지를 결정한다. 따라서 신뢰도가 낮은 호스트인 N이나 호스트 L의 사용자가 네트웍 주소를 조작해서 마치 자신이 신뢰도가 높은 호스트 M에 있는 것처럼 속이고자 할 것이다. 이런 종류의 공격 시도를 막기 위해서 기존의 단순한 방법으로만 네트웍 주소를 인증하는 것은 부족하다. DEC의 Estrin 등은 [10]에서 Visa와 접근 제어 서버를 이용해서 각 게이트웨이를 통과하는 패킷을 통제하는 방법을 설계했다. 본 논문에서는 방화벽을 위한 패킷들의 인증 방법을 설계한다. 여기서는 공개키 암호 알고리듬과 대칭키 암호 알고리듬을 함께 사용한다. 그리고 이 방법은 요새 호스트로부터 신뢰를 받고 3.1절에서 언급한 조건을 갖춘 호스트 M에서만 적용하며, 다른 호스트인 N, L에서는 기존의 단순한 방법으로 네트웍 주소 인증을 받는다.

그림 2에서 호스트 M은 사설망내의 호스트인 호스트 C에 접근하고자 할 때 먼저 요새 호스트 B에게 네트웍 주소 인증을 받아야만 한다. 호스트 M은 처음으로 서비스를 요청하는 패킷에 자신의 비

공개키로 서명해서 보내고 요새 호스트 B에게 인증을 받는다. 요새 호스트 B는 호스트 M의 신원을 인증하고 추후의 패킷들을 효율적으로 인증하기 위한 세션키를 만들어서 호스트 M에게 전송한다. 호스트 M은 추가의 패킷을 보낼 때 TCP/IP[11]에서 사용하는 sequence number와 세션키를 사용해서 자신의 패킷임을 인증받게 된다. 이 sequence number는 매번 패킷을 보낼 때마다 달라지는 값으로 Nonce[9]의 역할을 수행하게 된다.

- 서비스 요청
  호스트 M은 처음에 서비스를 요청할 때 자신의 주소(A_M), 원하는 목적지인 호스트 C의 주소(A_C), 사용자 이름(J), 그리고 호스트 M과 호스트 B의 sequence number인 SYN_M과 SYN_B를 자신의 비공개키(S_M)로 서명을 한다.

$$M \Longrightarrow B : S_M\{A_M, A_C, J, SYN_M, SYN_B\}$$

  호스트 M은 위의 서명을 자신의 인증서($CertM$)과 함께 요새 호스트에게 전달한다.

- 서비스 승인
  요새 호스트 B는 호스트 M의 인증서로부터 호스트 M의 공개키 P_M을 구해서, 서명을 검증한다. 요새 호스트 B는 호스트 M, 호스트 C, 사용자 J, 그리고 서비스의 종류를 가지고 서비스를 허가할 것인지 거절할 것인지를 판단한다. 허가된 서비스이면 추후의 패킷들의 인증을 위한 세션키(K_{MB})와 이에 대한 서명을 생성해서 자신의 인증서($CertB$)와 함께 호스트 M에게 전송한다.

$$M \Longleftarrow B : P_M\{K_{MB}\}, S_B\{h(K_{MB}, SYN_B, SYN_M)\}, CertB$$

- 패킷들의 전송
  호스트 M은 자신의 패킷에 sequence number와 세션키 K_{MB}를 이용한 메세지 인증 코드를 추가해 보냄으로써 요새 호스트 B로부터 인증받을 수 있도록 한다. 여기서 SYN_M와 SYN_B는 매번 달라지는 값이다.

$$M \Longrightarrow B : K_{MB}(SYN_M \oplus K_{MB}(SYN_B))$$

- 패킷들의 인증
  요새 호스트 B는 위의 메세지 인증 코드를 확인해서 호스트 M이 자신이 발급한 K_{MB}를 소유하고 있음을 확인한 후 사설망으로의 진입을 허가한다.

6 결론

본 논문에서는 기존의 방화벽에서 소홀히 다루었던 사용자 인증과 네트웍 주소의 인증 문제를 해결하기 위한 인증 시스템을 설계하였다. 사용자 인증을 강화하기 위해서 설계한 인증 프로토콜은 공개키 암호 알고리듬을 사용하고 서로의 인증서를 교환해서 인증 서버의 도움을 받지 않는 상호 인증 프로토콜이다. 이는 각 사용자가 지닌 스마트카드를 이용하며, local 로그인 프로토콜과 remote 로그인 프로토콜의 두 가지 인증 프로토콜로 나누어 진다. 또한 외부의 공중망에서 요새 호스트를 통해서 사설망으로 진입하고자 하는 패킷들을 통제하기 위해서, 암호학적 기법을 이용한 네트웍 주소 인증 방법을 제안하였으며, 신뢰도에 따른 차별화된 서비스를 제공하기 위한 접근 제어 정책에 대해서 알아 보았다.

참고 문헌

[1] Marcus J. Ranum, "Thinking About Firewalls", *Proceedings of Second International Conference on Systems and Network Security and Management(SANS-II)*, April 1993

[2] William R. Checwick and Steven M. Bellovin, "Firewalls and Internet Security : Repelling the Wily hacker", Addison-Wesley Publishing Company, 1994

[3] Robert Morris and Ken Thompson, "Password Security : A case History", *Communication of the ACM*, Volume 22, Number 11, November 1979

[4] Steven M. Bellovin, "Security Problems in the TCP/IP Protocol suite", *Computer Communications Review*, Volume 9, Number 2, April 1989

[5] D. Brent Chapman, "Network (In) Security Through IP Packet Filtering", *In Proceedings of the Third USENIX UNIX Security Symposium*, September 1992

[6] Jeffrey C. Mogul, "Simple and Flexible Datagram Access Control for UNIX-based Gateways", *In Proceedings of the USENIX Summer 1989 Conference*, 1989

[7] Cisco System, "Gateway System Manual ; Software Release 8.2", 1990

[8] ISO/IEC IS 9798-3, Information technology - Security techniques - Entity authentication mechanisms - Part3 : Entity authentication using a public key algorithm, 1993

[9] Roger M. Needham and Michael D. Schroeder, "Using encryption for authentication in large networks of computers" *Communications of the ACM*, Volume 21, Number 12, December 1978

[10] Deborah Estrin and Jeffrey C. Mogul and Gene Tsudik and Kamaljit Anand, "Visa Protocols for Controlling Inter-Organizational Datagram Flow : Extended Description", WRL Research Report 88/5, December 1988

[11] W. Richard Stevens, "TCP/IP Illustrated Volume 1. The Protocols", Addison-Wesley Publishing Company, 1994