
개방형 네트워크 보안 시스템 구조

김창수* · 정희경**

Open Based Network Security System Architecture

Chang-su Kim* · Hoe-kyung Jung**

요 약

기존의 시스템들이 보안적인 요소를 포함하려 한다면 많은 비용 문제와 함께 복잡한 설치 과정을 거쳐야 했으며 폐쇄적인 구조로 많이 구성되어 사용자 측면에서는 많은 불편을 감수해야만 보안이 확립되는 단점을 갖는 구조를 취하고 있었다.

본 논문에서는 이런 단점들을 공개 보안 도구 사용과 보안 서버를 프락시 서버, 인증 서버 등을 모두 포함하는 베이스천 호스트(bastion host) 형태의 방화벽 시스템으로 구성함으로써 집중적인 구성이 가능하여 비용 및 관리 면에서도 편리함과 독립적인 보안을 동시에 제공된다. 또한, 각각의 보안 대상 호스트에도 별도의 보안 소프트웨어를 설치하여 다양한 보안 레벨을 적용할 수 있도록 구성하였다. 더욱이 암호화 정책에 따라 암호화 알고리즘을 적용하여 보안의 수준을 한 차원 높였으며 자주 사용하는 서비스에 대해서는 프락시 서버를 제공하여 사용자들에게는 투명성을 제공했다.

본 논문에서는 시스템 보안과 네트워크 보안의 유기적인 연동을 통해 개방적이면서도 유연하고 독립적인 보안을 확립할 수 있는 형태의 보안 시스템을 제안한다.

ABSTRACT

If existing system need to expand security part, the security was established after paying much cost, processing of complicated installation and being patient with inconvenience at user's view because of closed structure.

In this thesis, those defects could be overcome by using open security tools and constructing security server, which is firewall of 'bastion' form including proxy server, certification server and so on. Also each security object host comes to decide acceptance or denial where each packet comes from, then determines security level each hosts. Precisely it is possible choosing the packets from bastion host or following at the other policies. Although an intruder enter into inside directly, it is constructed safely because encryption algorithm is applied at communication with security object host.

This thesis suggests more flexible, independent and open security system, which improves existing security through systematic linkage between system security and network security.

키워드

Network Security, Proxy Server, Bastion Host, Encryption Algorithm

* 청운대학교 인터넷학과

** 배재대학교 컴퓨터공학과(교신저자)

I. 서론

정보화 시대가 되어감에 따라 많은 곳에서 정보 시스템을 구축, 운영하고 있고 대부분의 작업들이 정보 시스템을 통해 이루어지고 있으며, 데이터의 이동 또한 컴퓨터와 네트워크를 통해 이루어진다. 하지만 정보 시스템 구축의 보안관리 및 네트워크 시스템 보안 운영을 소홀히 하여 네트워크 보안에 많은 허점이 노출되고 있다.

기존의 네트워크 시스템은 보안을 강화하려 많은 비용과 복잡한 설치 과정을 거쳐야 했으며 대부분 폐쇄적인 구조로 구성되어 사용자 측면에서 본다면 상당한 불편을 감수해야만 보안이 확립되는 구조를 갖는다[1-3].

이에 비용 및 관리측면에서 효율성 및 편리함과 독립적인 보안을 동시에 제공할 수 있는 개방형 네트워크 보안 시스템의 필요성이 대두되었다. 개방형 네트워크 보안 시스템은 네트워크 보안과 시스템 보안의 유기적이고 효과적인 결합이 이루어지게 되어 보다 향상된 보안 시스템으로의 발전 가능성에 대해서 제시한다.

이에 본 논문에서는 네트워크 보안과 시스템 보안의 유기적이고 효과적인 결합이 이루어지게 하여 보다 향상된 보안 시스템 구성이 가능한 개방형 네트워크 보안 시스템 아키텍처를 설계하였다.

본 논문의 구성은 다음과 같다. 제2장에서는 네트워크 보안 침입 유형을 기술하고, 제3장에서는 네트워크 시스템 보안 기반 기술을 설명한다. 제4장에서는 네트워크 보안 시스템 아키텍처를 기반으로 실제 개방형 네트워크 시스템을 설명하고 5장에서 결론과 함께 향후 연구방향을 대해 기술한다.

II. 관련연구

2.1 부당한 액세스 유형

2.1.1 시스템의 방해

어느 시스템의 중심적 역할을 가지고 움직이는 호스트 등에 대해서 대량의 패킷이나 메시지를 보내어 그 호스트의 CPU 사용률을 비정상적으로 높여 운용을 방해하는 것이다[1-4]. 또 다른 방법으로는 부정확한 경로 정보를 고의로 흘려 통신 불능, 또는 패킷루프와 같은 상태가 되는 경우도 있다. 이들 방법은 시스템에 영향을 주며 시스템 전체를 파괴시킬 수도 있다. 이를 그림 1에 보인다.

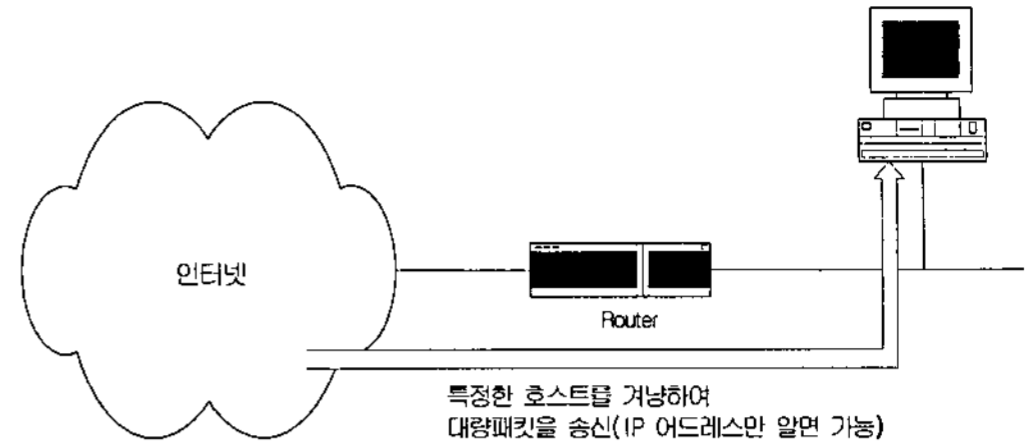


그림 1. 시스템 방해
Fig. 1. Disturbance of System

2.1.2 네트워크상의 데이터 복제

네트워크상에 흐르고 있는 데이터를 훔치는 방법도 있다. 이것은 회선 상에서 직접 데이터를 훔치는 방법이다[3-5]. LAN이나 WAN에서 실제로 사용되고 있는 네트워크에는 프로토콜 해석기 등을 장치함으로써 패킷을 수집할 수가 있으며 그 패킷에서 필요한 정보를 분석하는 것도 가능하다. 네트워크상의 데이터 복제는 그림 2에 보인다.

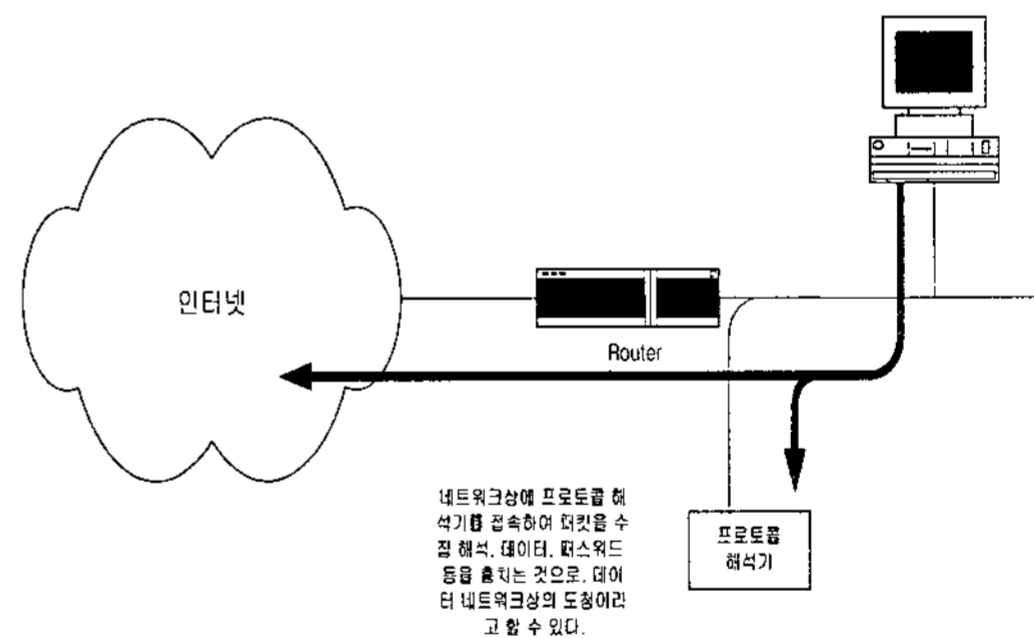


그림 2. 네트워크상의 데이터 복제
Fig. 2. Data Copy on Network

2.2 NIS(Network Information Service) 공격

패스워드, 호스트 등의 시스템 파일을 공유함으로써 문제 되고 있지만 일관성 있는 시스템 환경을 제공한다는 이점 때문에 사용하고 있는 곳도 적지 않다. 그러나 보안상의 약점이 노출되어 있으므로 상업적인 시스템에서는 사용을 자제하고 있는 실정이다. NIS에 의한 공격에는 3가지 유형으로 이루어진다[4-8].

- NIS 서버 스푸핑(spoofting)에 의한 공격방법으로 자신의 IP를 신뢰된 호스트인 것처럼 속여 공격

하는 IP 스푸핑 방법과 비슷한 절차로 이루어진다.

- NIS 클라이언트 스푸핑에 의한 공격방법으로 대상을 클라이언트 쪽으로 한다.
- 데몬 프로그램 버그를 이용하는 방법 - NIS 데몬 프로그램 자체의 버그를 이용하여 공격하는 방법이다.

NIS 서버 스푸핑은 NIS 클라이언트의 RPC 요청에 대해 서버보다 먼저 가짜 정보를 보내주어 클라이언트를 공격하는 방법이다. 그림 3에 NIS 서버 스푸핑에 대해 보인다.

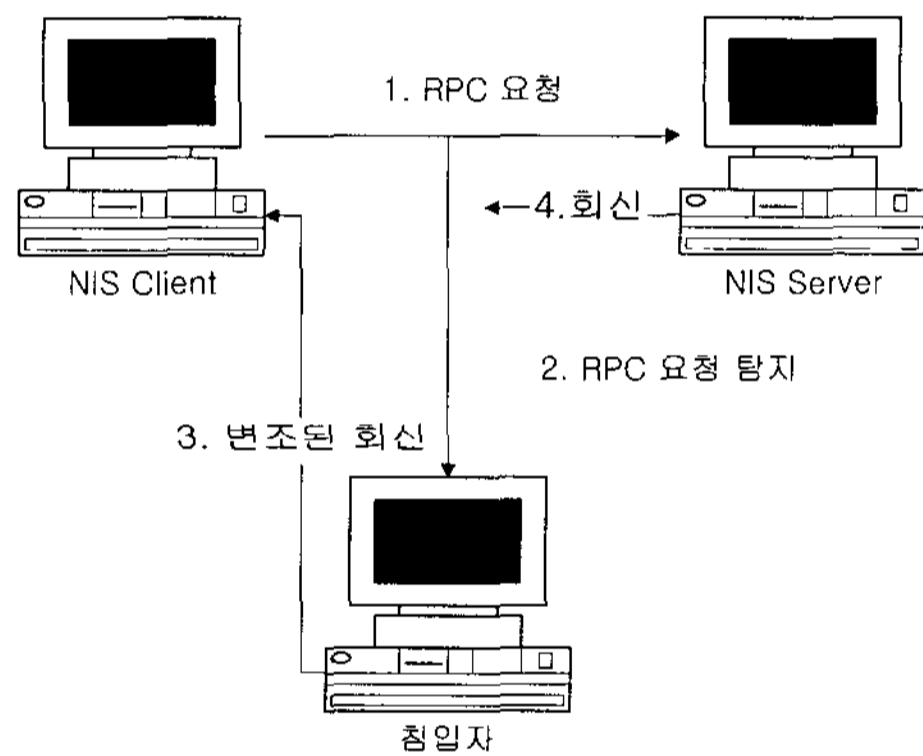


그림 3. NIS 서버 스푸핑 공격
Fig. 3. NIS Server Spoofing Attack

Ⅲ. 시스템 보안

본 장에서는 네트워크 보안과 시스템 보안의 효과적인 결합이 이루어져 향상된 보안 시스템을 구성하여 개방형 네트워크 보안 시스템 아키텍처 설계를 위한 시스템 보안 기반 기술에 대해 설명한다.

3.1 베이스천 호스트

베이스천 호스트는 보호하고자 하는 네트워크의 외부에 공개되어 있는 호스트를 지칭하는 말이다. 베이스천 호스트의 역할은 시스템 관리자가 망 보안의 가장 강력한 장소로 인식하며, 내부 망으로 침입할 수 있는 영역으로 베이스천 호스트를 통하게 함으로써 위험 지역의 범위를 안정시키는 역할을 해 준다[9-12]. 일반적으로

베이스천 호스트는 높은 보안 상태를 유지하고 있다 가정하고, 감사(Audit) 기능 또는 추적(Trail) 기능을 갖고 있으며, 보안 유지를 위한 응용 소프트웨어들이 존재한다. 대부분 2개의 패킷 필터링 라우터 사이의 유닉스 시스템을 베이스천 호스트라고 한다. 외부 라우터는 인터넷과 베이스천 호스트 사이의 트래픽만 허락된다. 내부 라우터는 내부 네트워크(안전한 네트워크)과 베이스천 호스트의 트래픽만이 허락된다.

베이스천 호스트 방식 방화벽의 구성은 유닉스 시스템과 2개의 라우터로 대부분 구성되어 비용 문제 및 지속적으로 발생하는 유닉스 보안 문제에 대처하기가 어려워 대부분의 베이스천 호스트 방화벽은 관리자의 기술에 전적으로 의존해야하는 문제점이 있다.

3.2 S/KEY

S/KEY는 벨코아(Bellcore)사에 의해 개발된 일회용 패스워드 소프트웨어 패키지로써 시스템에서 사용되는 패스워드는 오직 한 번의 사용자 인증에만 유효하며, 동일한 패스워드의 재사용은 불가능하다. 더욱이 이미 사용된 패스워드는 다음번에 사용할 패스워드와 관련한 어떠한 정보도 제공해 주지 않는다. 따라서 네트워크 스니핑(Network Sniffing)에 대처할 수 있다.

S/KEY의 동작은 먼저 사용자는 시크리트(secret) 패스워드와 생성할 패스워드의 갯수인 n 을 선택하여 S/KEY를 초기화하면, 시큐어 해시 함수가 시크리트 패스워드에 n 번 적용되고, 그 결과 값이 서버에 저장된다. 초기화가 완료된 후 처음으로 사용자가 로그인을 시도하게 되면 서버에서 $n-1$ 의 도전이 발생하게 되는데, 이때 사용자가 $n-1$ 의 일회용 패스워드를 기억할 경우 그것을 그대로 입력하여 서버에게 검증을 받으면 된다. 그러나 일반적으로 사용자가 n 개의 일회용 패스워드를 모두 기억하기란 매우 어려운 일이다. 따라서 사용자의 클라이언트에 일회용 패스워드를 생성할 수 있는 프로그램이 있다면 매우 편리하게 사용할 수 있을 것이다. 이런 필요에 의해서 다행스럽게도 다양한 클라이언트 플랫폼에서 동작하는 패스워드 생성 프로그램이 현재 존재하고 있다. 즉 시크리트 패스워드를 물어 보고 여기에 해시 함수의 $n-1$ 순차를 적용하여 얻어진 결과인 일회용 패스워드를 서버로 보내게 되는 것이다. 서버에서 여기에 해시 함수를 적용한 후 미리 저장되어 있는 값과 비교함으로써 사용자 인증을 수행하게 되는 것이다. 사용자

의 인증이 성공하면 서버에서는 패스워드 카운트를 n-2로 줄여 다음 로그인에 대비하게 된다.

그러나 위와 같이 일회용 패스워드의 생성에 사용되는 시크리트 패스워드가 네트워크를 통해 전송된다면 네트워크 스니핑에 의해 이것이 노출될 가능성이 있다. 이처럼 시크리트 패스워드가 네트워크상에 노출되면, 일회용 패스워드 인증 시스템을 사용함으로써 얻어지는 보안상의 장점이 상실되게 된다. 따라서 시크리트 패스워드를 통한 일회용 패스워드 획득의 과정은 반드시 클라이언트 상에서 이루어지도록 해야 한다.

3.3 보안 셸(Secure Shell)

네트워크의 발전으로 원격지에서 서버에 접속해 작업을 하는 경우가 빈번해짐에 따라 원격지에서 작업을 하는데 있어서 보안 유지를 위해 보안 셸을 사용하게 된다.

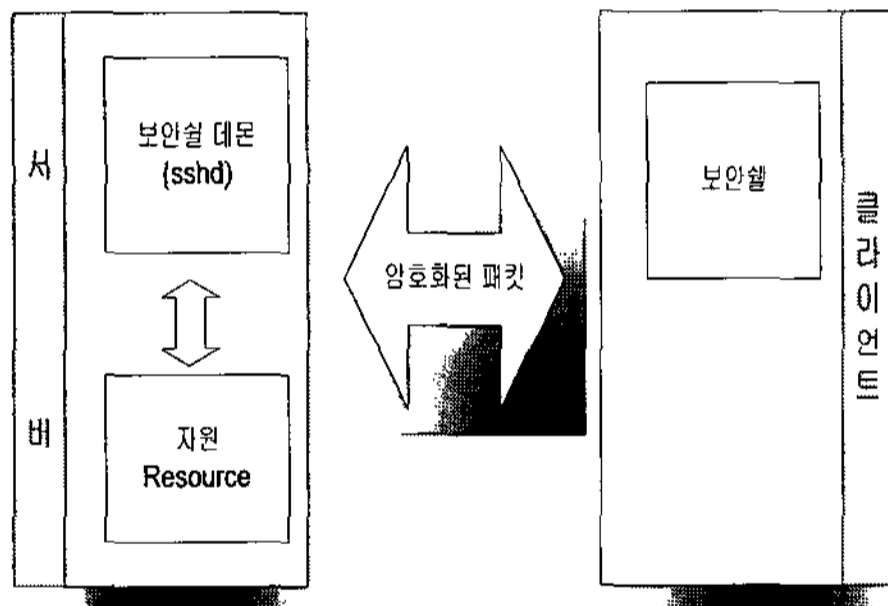


그림 4. 보안 셸의 구조
Fig. 4. Structure of Secure Shell

보안 셸의 구조는 그림 4와 같다. 보안 셸은 네트워크를 흐르는 데이터를 암호화해서 데이터의 유출이나 변조를 막아서 데이터 송수신시에 보안을 강화하기 위한 것이다. 보안 셸에서는 네트워크 데이터 통신을 암호화하고 FTP에서 데이터 통신을 암호화 하며 데이터 통신을 암호화 할 수 있게 해준다.

보안 셸은 보안에 관한 문제를 해결할 수 있지만 이로 인한 성능 저하의 문제점이 있다. 보안 셸을 사용하는 시스템과 사용하지 않는 시스템간의 사용에는 암호화가 되지 않기 때문에 보안 셸을 사용하는 시스템이 적을 경우 효과는 아주 미미하게 되는 문제점이 있다.

IV. 개방형 네트워크 보안

본 장에서는 시스템 보안 기반 기술을 활용한 네트워크에 보안 시스템 아키텍처를 기반으로 설계한 개방형 네트워크 시스템에 대해 설명한다.

4.1 라우터 설정 보안

4.1.1 어드레스 제어

인터넷 접속은 일반적으로 라우터를 사용하며 조직 내부에서 인터넷에 접속하는 경우 라우팅이 수행된다. 라우터는 송신원 IP 어드레스와 수신처 IP 어드레스를 보고 라우팅여부를 판단하여 송신원 IP 어드레스만 한정하거나 송신원 IP 어드레스와 수신처의 IP 어드레스 쌍(pair)으로 한정하여 허가된 어드레스만 인터넷으로의 통신을 허가하는 방법이 있다.

라우터 상에 호스트 A의 IP 어드레스와 호스트 B의 IP 어드레스만 패킷 통신만을 허용한다고 정의하면 호스트 A와 호스트 B만이 통신할 수가 있고 그 이외의 호스트들은 통신을 직접적으로는 할 수 없게 된다. 그림 5에 조직 내외의 통신과 패킷 내의 IP 어드레스 제어를 보인다.

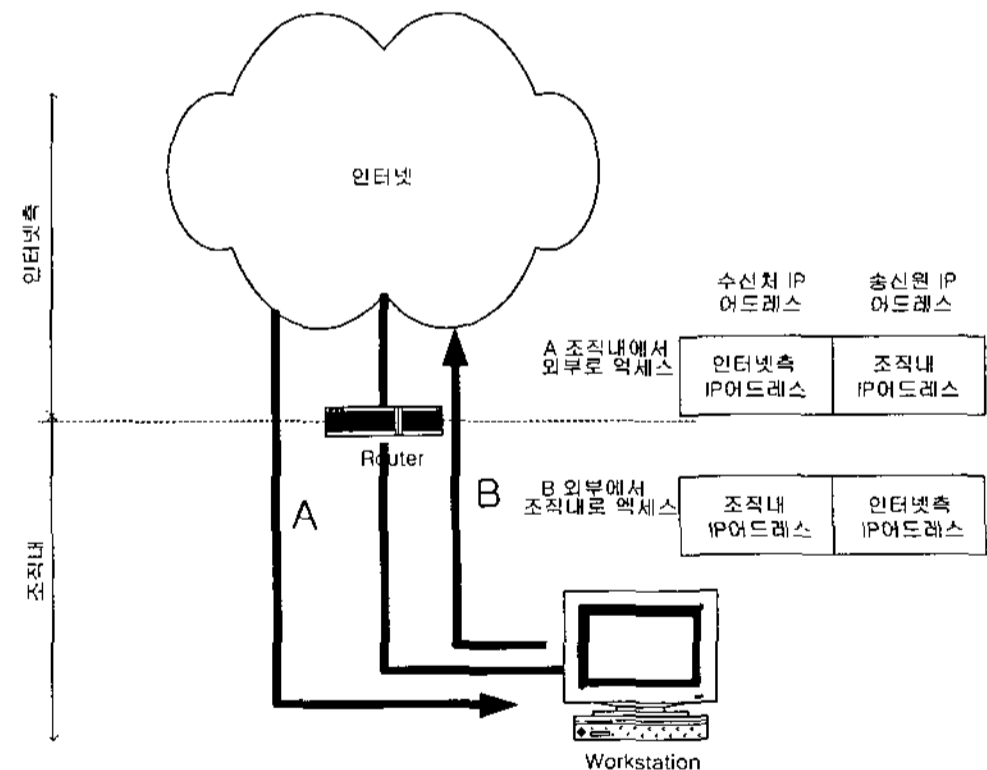


그림 5. IP 어드레스 제어
Fig. 5 IP Address Control

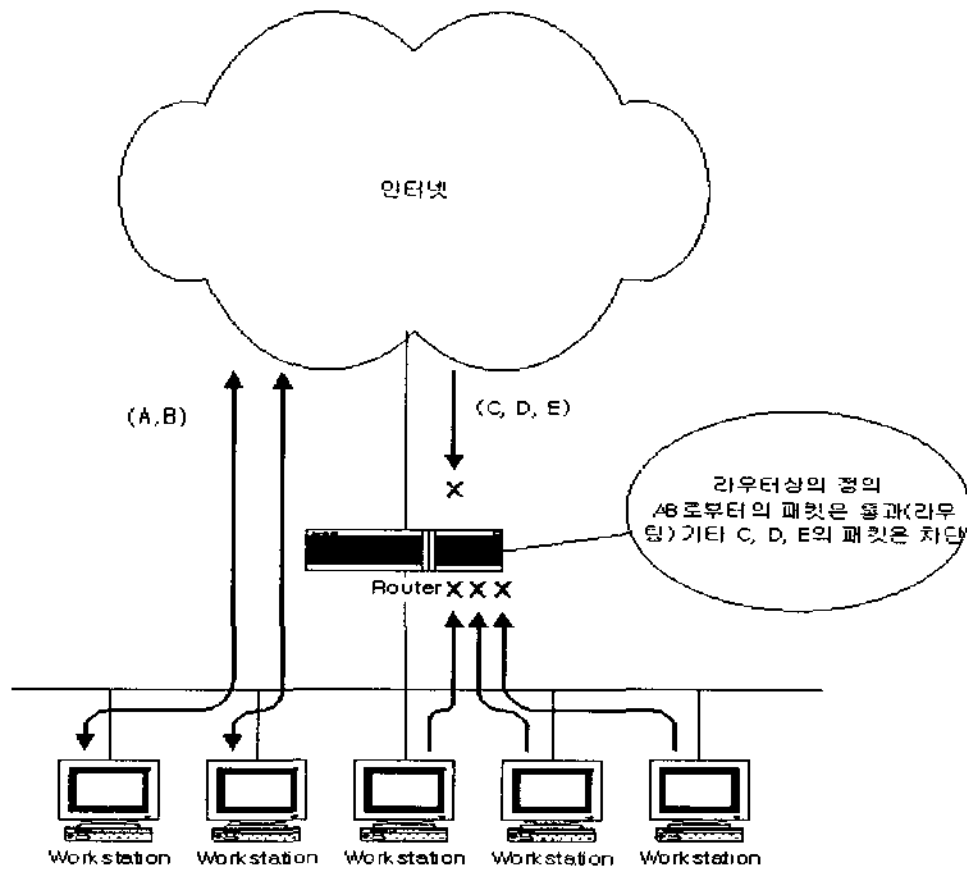


그림 6. 라우터에 IP 어드레스 정의
Fig. 6. Define IP Address Into Router

그림 6은 라우터에 IP 어드레스 정의를 보인다. IP 헤더의 참조 및 분석이라는 라우터의 기능을 이용해서 통신의 보안을 설정한다. IP 어드레스를 체크할 때 보안 여부를 판단하여 보안 설정에 위배되는 패킷은 라우터를 통과시키지 않고 파기한다. 이와 같은 설정을 한 경우는 라우터가 패킷을 처리할 때 일반적인 라우팅 처리에 이 보안용 처리가 더해진다.

일반적인 라우터의 라우팅 처리의 경우 통신 패킷을 수신하며, 우선 IP 헤더를 보고 헤더 필드 중 수신처 IP 어드레스에서 수신처의 IP 네트워크를 판단한다. 수신처의 IP 네트워크까지 패킷을 라우팅 시키면서 라우팅 테이블을 검색하여 다음 라우터에 패킷을 전송한다.

4.2 라우팅 제어

라우팅 제어는 라우팅에 관계되는 경로 정보 패킷에 대한 보안 설정에 따르게 된다. 라우팅 제어를 위한 보안 방법으로 효율적인 것은 정적 라우팅 정의를 통한 경로 설정이다. 정적 라우팅 정의는 각 호스트 및 각각의 라우터가 라우팅 정보를 고정으로 갖고 교환을 하지 않기 때문에 시스템 방해에 대응할 수 있다. 그림 7에 정적 및 동적 라우팅을 보인다.

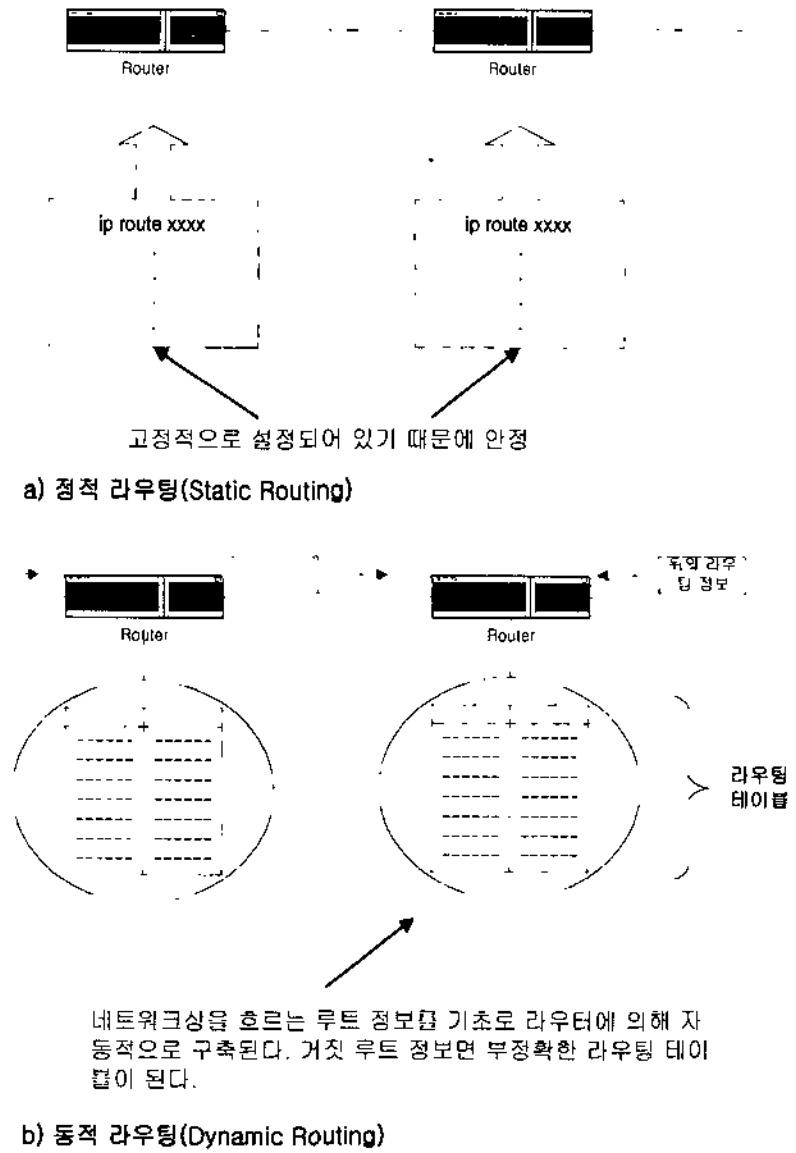
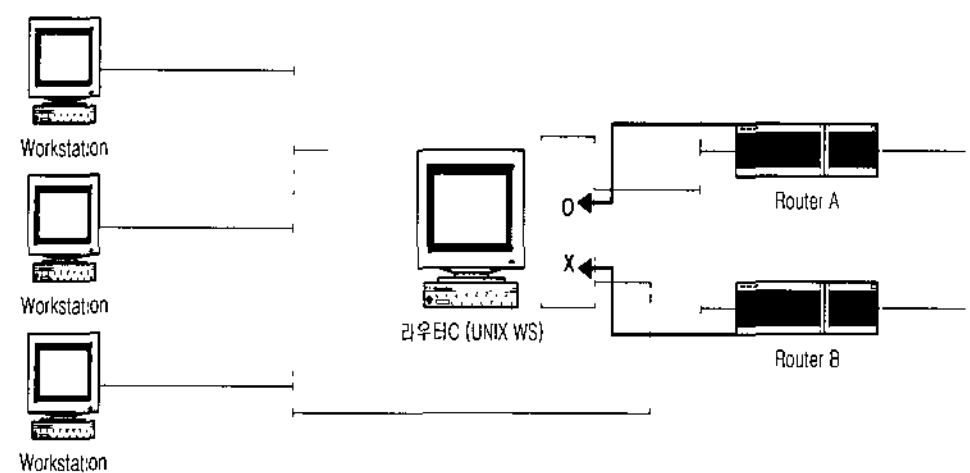


그림 7. 정적 라우팅과 동적 라우팅
Fig. 7. Static Routing and Dynamic Routing

하지만 정적 라우팅 정의는 관리가 복잡하고 동적인 경로 변경을 할 수 없다는 단점을 지니고 있다. 따라서 RIP(Routing Information Protocol)나 OSPF(Open Shortest Path First)와 같은 동적 라우팅 프로토콜을 사용하여 특정 라우터 정보만을 허용하는 gated 액세스 제어의 기능을 사용하여 라우팅 제어를 함으로써 보안을 향상시킬 수가 있다. 이는 그림 8과 같다.



라우터 C는 gated의 설정에 의해 라우터 A로부터의 라우팅 정보는 허용하지만 라우터 B로부터의 정보는 허용하지 않는다.

그림 8. gated에 의한 라우팅 정보의 인증
Fig. 8. Authentication of routing information by gated

4.3 개방형 네트워크 보안 시나리오

네트워크 보안 툴 및 유틸리티들을 이용하여 보다 유연하면서 향상된 기능을 제공 할 수 있는 보안 시스템과 다양한 형태의 네트워크 보안 시나리오 중 중소기업의 회사 및 연구소에 알맞은 보안 시나리오이다. 중소기업이라고 해도 대규모 네트워크에서도 자신만의 보안 서버 네트워크를 구축할 수 있도록 유연성을 확보하여 구성하였다. 그림 9는 개방형 네트워크 보안 시나리오를 위한 구성도이다.

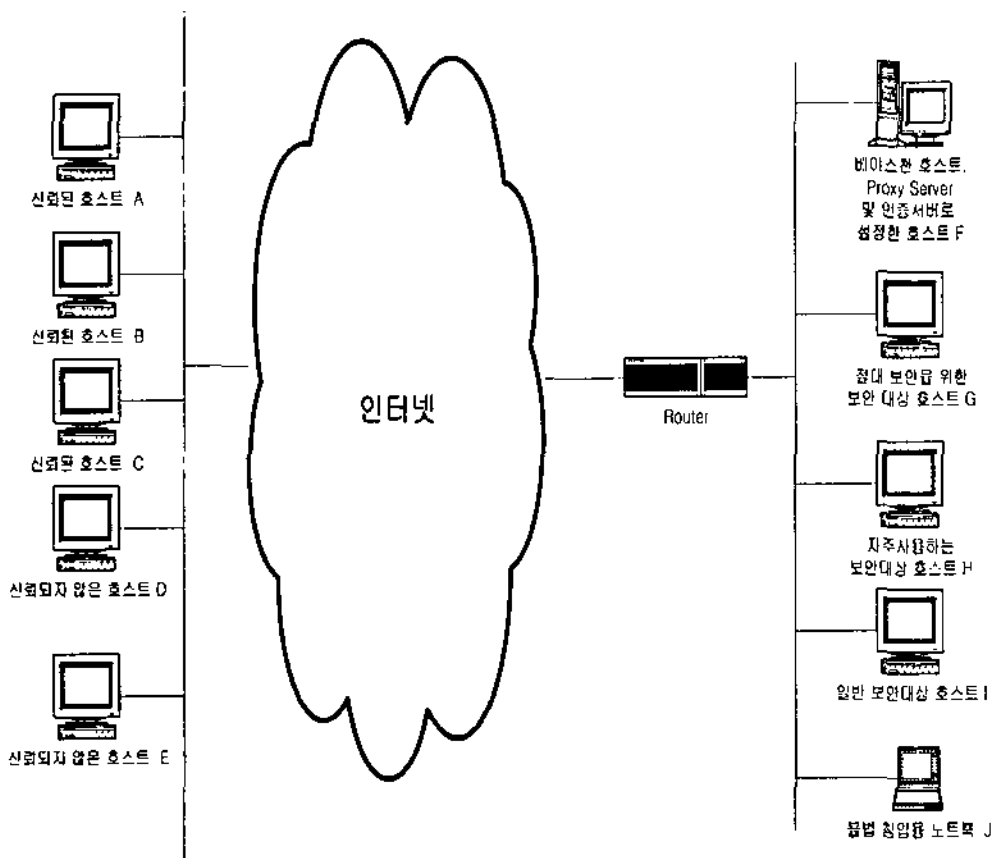


그림 9. 개방형 네트워크 보안 시나리오를 위한 구성도.

Fig. 9 Diagram for Open Type Network Security Scenario

4.3.1 호스트 F 구성

베이스천 호스트의 구성은 베이스천 호스트 구성 규칙에 따라 최대한 철저히 안정된 서비스만 제공되며 사용자 계정은 관리자 외에는 모두 제거했다. 또한 관리자의 경우에도 외부에서의 접속에 의한 관리는 허용하지 않았다. TIS 방화벽 툴킷(TIS Firewall Toolkit)을 이용하여 인증서버와 프락시 서버를 구성했으며 도메인 네임 서비스를 제공하고 있다. 암호화 기법 적용단계에 따라 암호 셸과 S/KEY, logdaemon등을 사용할 수도 있다.

4.3.2 보안대상 호스트 G, H, I 구성

보안을 위한 보안 대상 호스트에는 일단 TCP Wrapper를 설치한 후 각각의 특성에 따른 접근 규칙을 따로 두었는데 그 접근 규칙은 절대 보안을 요하는 호스트 G의 경우 내부네트워크에 대해서도 오직 호스트 F로

부터의 정상적인 접속만을 허용하고 외부 네트워크에 대해서는 접근조차 허용되지 않게 구성한다. 자주 사용하는 호스트 H의 경우는 외부 네트워크에 대해서만 접속을 허용하지 않으며 내부 네트워크에 다른 호스트들로부터의 접속을 모두 허용한다. 일반 보안 대상 호스트 I의 경우는 호스트 F로부터의 접속과 정상적 외부접속을 허용한다. 그리고 내부 네트워크에서도 접속 대상 호스트를 규정하여 그 호스트만이 접근가능 하게 구성하였다.

4.3.3 그 밖의 구성

각각의 보안 대상 호스트에는 보안 셸을 적용하였고 보안 대상 호스트가 적고 일반 사용자가 많다면 보안 셸 적용을 하지 않아도 된다. 프락싱 동작 시 S/KEY로 암호화 기법을 적용하였고 각각의 호스트 접근 규칙을 portmap, netacl, TCP wrapper등으로 네트워크 접근 규칙을 설정 한다.

4.4 개방형 네트워크 보안 시스템 동작 방식

1) 신뢰된 호스트로부터 보안 대상 호스트로의 접속
 신뢰된 호스트로부터 보안대상 호스트에 접속은 라우터에서 라우팅 테이블의 규칙에 따라 접속 가능한 호스트인지를 판단하고 각 서비스에 따라 베이스천 호스트를 경유하여 또 한 번의 접속 규정을 따르게 된다. 각각의 해당 서비스의 프락시 서버를 사용하여 보안대상 호스트로의 접속이 이루어진 후 인증이 필요한 서비스에 한해서 인증 과정을 거친 호스트에 한해서 접속이 허용 된다.

2) 신뢰되지 않은 호스트로부터 보안 대상 호스트로의 접속

신뢰되지 않은 호스트로부터 보안 대상 호스트로 접속을 시도하면 라우터에서 라우팅 테이블 규칙에 따라 접속 허용 여부를 판단한다. 이때 대부분의 신뢰 되지 않은 컴퓨터는 제외된다.

일부 해킹기술로 라우터를 통과 하여 베이스천 호스트로 접속이 이루어지면 베이스천 호스트에서 해당 호스트의 접속 규정을 다시 한 번 적용하게 된다.

3) 보안대상 호스트에서 다른 보안 대상 호스트로의 접속

보안 대상 호스트에서 다른 보안 대상 호스트로의 접속은 베이스천 호스트와 프락시 서버 및 인증 서버를 통

과하게 된다. 이는 각 보안 대상 호스트의 보안 수준을 각각 다르게 줄 수 있어 높은 수준의 보안을 요하는 호스트와 자주 사용하면서 일반적인 보안 수준을 따르는 호스트들의 유연성을 확보할 수 있다.

4.5 개방형 시스템의 구성 이점

1) 호스트 보안과 네트워크 보안의 유기적인 연동

최신의 해킹 기술들이 발전하여 이전의 보안 기술에 많은 문제점이 나타나고 있어 최신 해킹 기술들로부터 안전한 베이스천 호스트를 구성하고 암호화 어플리케이션을 적용하여 보안 수준을 높일 수 있다.

2) 보안대상의 범위의 확장

방화벽 혹은 보안 자제가 내부 네트워크의 보안만을 위한 폐쇄적인 구성이 대부분이지만 개방형 보안 시스템의 경우는 외부에 있는 시스템도 보안 대상에 포함할 수 있다.

3) 유연하고 다양한 보안수준

폐쇄적인 보안 환경에서도 외부 네트워크와 내부 네트워크간의 연결을 프락시 서버를 사용하여 유연하게 제공하므로 사용자들에게는 투명성을 보장할 수가 있다. 즉 사용자 입장에서의 편리함을 최대한 고려하여 설계되었다. 내부 네트워크에 한 가지 보안 수준이 아닌 다양한 보안 수준을 적용할 수가 있다.

각각의 보안 대상 호스트 자체에서 보안을 적용하며 베이스천 호스트와 프락시 서버 인증 서버를 전문 서버 하나가 담당하게 되고 대부분이 공개 소프트웨어로 보안 시스템 구축하므로 상당한 비용 절감 효과 및 설치가 용이하며 이식성이 뛰어나다.

보안 적용 자체는 집중 식으로 구성되어 있어 관리가 간결하며 또한 접근 제한 자체는 따로 관리 되므로 유연한 구성이 가능하다.

V. 결 론

정보통신의 급격한 발전과 더불어 컴퓨터가 보편화되면서 인터넷 사용이 생활화되고 그에 따른 다양한 형태의 네트워크 시설과 기술들이 선보이고 있다. 네트워크의 보급은 생활의 편리성을 제공하였지만 정보에 대한 전송방해, 도청, 불법변조 등의 위험뿐만 아니라, 네트워크 불법접근으로 보관 중인 정보가 유출되는 실정

이다.

이에 대처하기 위해 많은 비용과 시간이 투자될 뿐만 아니라 다양한 기술과 고도의 시설 및 장비가 투입되고 있다. 하지만 소규모 단일 네트워크에서는 보안장비를 관리하고 운용하는데 소요되는 비용과 기술 부족에 따른 어려움이 있다.

이에 본 논문에서는 시스템 보안과 네트워크 보안의 유기적인 연동을 통해 개방적이면서도 유연하고 독립적인 보안을 확립할 수 있는 형태의 보안 시스템을 제안하였다.

본 논문에서는 공개 보안 도구 사용과 보안 서버를 프락시 서버, 인증 서버 등을 모두 포함하는 베이스천 호스트 형태의 방화벽 시스템으로 구성함으로써 집중적인 구성이 가능하여 비용 및 관리 면에서도 편리함과 독립적인 보안을 동시에 제공한다. 또한, 각각의 보안 대상 호스트에도 별도의 보안 소프트웨어를 설치하여 다양한 보안 레벨을 적용할 수 있도록 구성하였다. 더욱이 암호화 정책에 따라 암호화 알고리즘을 적용하여 보안의 수준을 한 차원 높였으며 자주 사용하는 서비스에 대해서는 프락시 서버를 제공하여 사용자들에게는 투명성을 제공했다.

향후 연구로서 다양한 네트워크 환경의 동적인 변화에 따라 동적으로 적용할 수 있는 네트워크 보안 관리 모델에 관해 연구하고자 한다.

참고문헌

- [1] 최용락 외 3인, 통신망 정보 보호, 도서출판 그린, 2002
- [2] 채규혁, 인터넷 방화벽 구축하기, 한빛 미디어, 2001
- [3] Simson Garfinkel, Gene Spafford, Practical UNIX & INTERNET Security 3rd edition, O'Reilly & Associates, 2003
- [4] Dieter Gollmann, Computer Security, John Wiley & Sons, 2006
- [5] Raymond R.Panko, Corporate Computer Network Security, Prentice Hall, 2003
- [6] Elizabeth D. Zwicky, Simon Cooper, D. Brent Chapman, Building Internet Firewalls, O'Reilly & Associates, 2000

- [7] Norbert Pohlmann, Tim Crothers, Firewall Architecture for the Enterprise, John Wiley & Sons, 2002
- [8] Loza, Boris, Unix, Solaris And Linux : A Practical Security Cookbook: Securing Unix Operating System Without Third-party Applications, Lightning Source Inc, 2005
- [9] 김재선, about Firewall & Network Security, 영진.com, 2002
- [10] 한국전산원, 방화벽 프로토타입 개발 보고서, 1996
- [11] 한국 전산원 표준본부, 방화벽 시스템의 구축과 운용, 1996
- [12] 한국 정보보호센터, Firewall 시스템 총서, 1996

저자소개

김 창 수(Chang-Su Kim)



1996년 배재대학교 전자계산학과
(이학사)
1998년 배재대학교 전자계산학과
(이학석사)

2002년 배재대학교 컴퓨터공학과(공학박사)
2001년~2004년 배재대학교 IT 교육센터 책임강사
2005년~현재 청운대학교 인터넷학과
※관심분야 : XML, Semantic web, 멀티미디어문서정
보처리, Ubiquitous Computing, USN

정 회 경(Hoe-Kyung Jung)



1985년 광운대학교 컴퓨터공학과
(공학사)
1987년 광운대학교 컴퓨터공학과
(공학석사)

1993년 광운대학교 컴퓨터공학과(공학박사)
1994년~현재 배재대학교 컴퓨터공학과 교수
※관심분야 : 멀티미디어 문서정보처리, XML, SVG,
Web Services, Semantic Web, MPEG-21, Ubiquitous
Computing, USN