정보학연구 제3권 제3호 2000년 9월(pp.27-37)
The Journal of Information Technology, Vol.3, No.3

침입차단시스템 제품 인증을 위한 취약성 평가 방법*

김정구**

요 약

정보보호 제품의 신뢰성 보증에 대한 평가 기준은 통일을 가져왔지만 국가별 평가 방법 등의 객관성 확보는 모호한 부분이 있다. 즉, 정보보호 제품의 취약성 평가에는 평가의 범위, 깊이, 시험 등에 적용하는 도구와 시나리오, 그리고 평가자 등이 있으나 이러한 요소들에 따라 평가 결과가 달라질 수 있으므로 평가의 공정성, 객관성 확보에 어려움이 있다.

따라서 본 논문은 정보보호 제품 보안 취약성 평가의 공정성, 객관성, 그리고 효율성을 보증할 수 있는 정보보호 제품 보안 취약성 평가 방법을 제안하고, 평가 수행을 자동화 할 수 있는 시스템(TSVES)을 설계 구현하였다.

Ⅰ. 서론

인터넷을 통한 네트워크 침해사고가 급증함에 따라 이에 대응할 수 있는 다양한 형태의 정보보호 제품이 개발되어 설치 운용되고 있다. 그러나 정보보호 제품을 적절히 설치하지 않았거나 잘 설치되었다 할지라도 정보보호 제품 자체가 취약점을 가지고 있다면 오히려 더 큰 위험과 손실이 발생할 수 있으므로, 정보보호 제품의 신뢰성 보증이 무엇보다도 중요하다고 할 수 있다[1]. 이에 따라 일부 국가들은 자체 개발된 제품들과 수입된 제품들에 대한 신뢰성 보증을 위해 평가기준을 제정하여 정보보호 제품 인증 제도를 시행하고 있고, 미국, 영국, 캐나다, 프랑스, 독일 등은 상호 평가 인증을 위한 국제공통 평가기준(CC:Common Criteria)을 제정하여, 99년 ISO 표준을 획득하였다. [15]

이러한 CC의 평가 기준에서 보증요구사항을 충족시키기 위한 정보보호 제품(평가진행중인 제품은 TOE(Target of Evaluation)라 함) 보안 취약성 평가는 인적, 시간적 낭비 요소가 많으며, 평가자의 경험과 전문성에 따라 각각 다른 결과를 가져올 수 있는 문제점이 있다.[3]

따라서 이러한 TOE 취약성 평가의 효율성을 높이고, 평가 결과의 공정성과 객관성을 확보하여 CC의 상호인증 목적을 충족하는 결과를 얻을 수 있도록 평가방법의 개발과 이를 자동적으로 수행할 수 있는 도구의 개발이 절실히 요구되고 있다.

그러므로 본 논문은 CC 요건과 평가 자동화 요건을 만족할 수 있는 TOE 취약성 평가 방법을 제안하고, 취약성 평가 도구를 설계 구현함으로써, CC의 평가 기준을 모든 정보보호 제품 평가에 효율적으로 적용할 수 있게 하여, 인증에 대한 신뢰도를 더욱 높일 수 있도록 하였다.

* 본 논문은 99년도 남서울대학교 교내 연구비 지원에 의해서 수행되었습니다.
** 남서울대학교 컴퓨터학과 전임강사

II. 기존 보안 취약성 평가 방법

CC의 평가 기준이 일반화되지 않은 시점에서, 국가별 TOE에 대한 평가는 자체 평가 기준에 따라 평가 절차를 작성하여 평가를 수행하고 있다. 따라서 평가 방법은 TOE 취약성 평가 방법으로는 개발자에 의해서 제공된 도구를 이용하여 보안 기능에 대한 평가를 수행하고, 평가자의 독립적인 보안 취약성 평가는 다음과 같은 매뉴얼 방식, 보안 스캐너 이용 방식, 그리고 해킹 도구 이용 방식 등이 있다[3].

2.1 매뉴얼(수작업) 방식

매뉴얼 방식은 평가자가 TOE 개발자로부터 TOE 보안 기능에 관련된 정보를 받아, 평가자 회의를 거쳐 TOE 취약성 평가를 위한 시나리오를 만들고, 평가 시나리오에 따라 TOE와 대화형태(interactive)로 평가하는 방법이다. 즉, 침입차단시스템의 경우 동시 세션의 연결 개수에 대한 평가는 평가자가 하나씩 연결을 설정하여, TOE가 허가하는 개수 이상에서 연결 거부가 일어나는가를 점검하는 형태로 최대 허용 개수만큼의 실질적인 평가는 사실상 불가능하다. 이러한 매뉴얼 방식은 TOE가 어떻게 반응하고 있는가를 즉시 알 수 있으나, 정보보호 제품의 기능 허용 한계 값에 대한 평가와 그 허용 값의 증가에 따른 제품의 성능에 대한 평가, 그리고 한계 값 이상에서 나타날 수 있는 오용에 대한 평가가 불가능하고, 평가자는 고도의 기술이 요구되며, 평가에 많은 시간이 필요한 문제점이 있다.

2.2 보안 스캐너 이용 방식

SATAN, ISS 등 보안 스캐너를 이용하는 방법은 매뉴얼 방식에 비해서 매우 빠른 결과를 얻을 수 있다. 그러나 일반적인 보안 스캐너는 네트워크 보안 스캐너라 할지라도 호스트 보안을 목표로 하는 운영체제 취약점과 위험성 있는 서비스를 찾는 기능 수행을 목표로 하고 있어, TOE에 대한 시스템 구성상 취약성과 구조상 취약성을 평가하는데는 미약한 점이 있다.

2.3 해킹 도구 이용 방식

해킹 도구는 인터넷을 통해서 쉽게 얻을 수 있다. 그러나 이러한 해킹 프로그램을 사용하는 데는 문제가 있다. 해킹 프로그램을 사용하기에 앞서 소스 프로그램 코드를 분석하고, 정확하게 동작하는지를 먼저 파악한 다음, 실제 자신의 시스템에서 실행 검토 후 적용하여야 한다. 그래도 해킹 프로그램을 TOE 취약성 평가에 이용하는 것은 예측할 수 없는 부작용이 있을 수 있음을 고려치 않을 수 없다.

III. 제안 방법

평가 결과의 공정성과 객관성을 만족시키기 위해서는 평가자나 평가 기관의 인위적 평가 작업을 최대한 배제 할 수 있도록 평가 자동화 방법이 개발되어야 한다. CC의 TOE 취약성 평가는 매우 중요한 보증요구사항으로 정보보호 제품의 잘못된 구성, 운영, 그리고 오용으로 인하여 취약점이 발견되지 않음을 보증하는 것으로 독립적 취약성 평가, 보안 기능 강도 평가, 오용 평가, 그

리고 가상 침투 시험으로 평가 단계를 나누었다.

3.1 독립적 취약성 평가

독립적 취약성 평가는 우선적으로 TOE의 보안기능이 명시된 대로 수행됨을 보여야 한다. 그리고 TOE가 안전하지 못하게 구성, 설치, 및 운용되며 관리자에게 이러한 사항이 탐지되지 않을 가능성을 평가하여야 한다. 평가자가 수행하는 시험은 개발자가 수행했던 시험과 같은 방법과 다른 방법을 병행하여 반복적으로 수행하여야 하며, 이를 자동화할 수 있도록 개발자는 TOE의 정보를 기계가 읽을 수 있는 문서로 평가자에게 제공하여야 한다. 또한, 평가자는 시험 결과 신뢰성 확보를 위해서 개발자의 자체 시험을 표본으로 추출하여 반복 시험하고, 이를 근거로 TOE가 광범위한 조건에 정확하게 작용하는지를 평가한다.

3.2 보안 기능 강도 및 오용 평가

TOE에 대한 보안 기능 강도 및 오용 평가는 (그림 1)과 같이 평가 환경을 구축하고 개발자가 제시한 기능의 강도(strength of function)가 정확한지를 점검한다. TOE 보안 기능이 우회, 무력화, 또는 변조되지 않는다 하더라도, 아직 그 하부 메커니즘의 개념에 취약성이 있기 때문에 그것을 파괴하는 것이 가능하다. 그러므로 평가자가 수행하는 기능 및 오용 시험은 TSF (TOE Security Function)가 기능 요구사항을 만족하는데 필요한 특성을 나타내는지를 평가한다.

단순한 TOE의 기능만을 시험하는 기능 시험은 TSF가 명시된 이상을 하지는 않지만 기능 강도 시험은 TSF의 명시된 범위에서 적응 능력을 평가하는 것으로, 과한 시험 평가를 적용할 경우에 대한 TOE의 반응을 평가한다. 그리고 바람직하지 않은 특정한 활동이 없으며, 부정적인 시험을 통해서 오용에 대한 TOE의 반응을 평가한다.

3.3 가상 침투 시험

취약성 분석에서 찾아낸 취약점들을 이용한 위험 발생여부를 반복적으로 평가하기 위한 시험으로 가상 침투를 통해서 TOE가 공격 능력을 갖는 공격자가 수행하는 가상 침투 공격에 내성

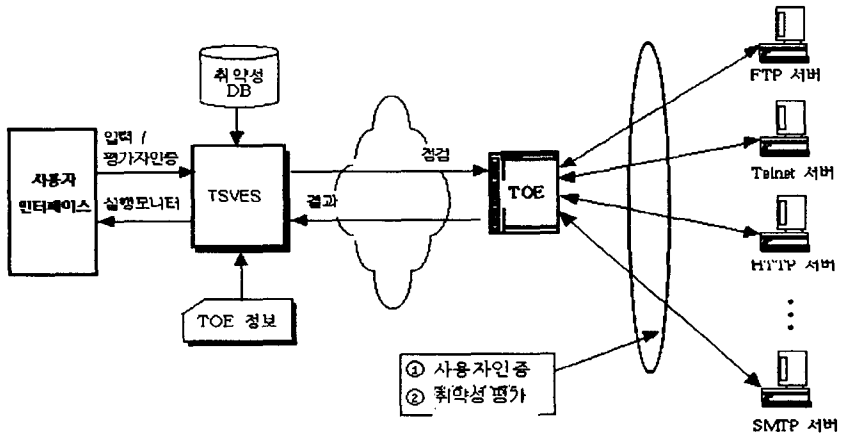

그림 1. 보안 기능 강도 및 침투 시험 모델

이 있는지를 시험하여 확인한다.

가상 침투 시험은 보안 기능 TCB(Trusted Computing Base) 혹은 TSF의 강도를 평가하는 하나의 방법으로, 정보보호 시스템의 보안 통제를 침해하는 방법을 찾기 위하여 평가자가 시행하는 것이다. 즉, 가상 침투 시험은 시스템의 보안을 침해하는 시스템의 스펙, 설계, 구현, 그리고 운용 상 존재하는 취약성을 찾아, 중간 정도의 침투 공격을 가하여, TOE의 내성을 평가하는 것이다.

(1) 외부로부터 가상 침투 공격 : 침입차단시스템을 평가하려고 할지라도 웹(WWW) 서버나 FTP 서버, 외부의 메일 서버를 공격할 필요가 있다. 침입차단시스템이 이런 호스트들을 사용하지 못하게 하거나 침입차단시스템의 파일 시스템이 웹 서버에 마운트된 예가 있다면 가상 침투 시험을 통해서 추가적인 취약성 정보를 발견할 수 있다.

(2) 내부로부터 가상 침투 공격 : 침입차단시스템의 경우 추가적인 보안을 사용해서 내부 사용자의 접근 제어를 실현하고 있다. 이러한 침입차단시스템이 설치된 내부망에서 외부망에 접속하기 위한 가상 침투 시험을 함으로써 침입차단시스템의 새로운 취약성 정보를 발견할 수 있다.

Ⅳ. TSVES 설계와 구현

TSVES(TOE Security Vulnerability Evaluation System)는 CC의 평가 원칙을 기반으로 본 논문에서 제안한 TOE의 보안 취약성 평가 자동화 모델에 따라 취약성 평가의 효율성, 신뢰성, 그리고 표준화를 가져올 수 있도록 설계 구현된 TOE 보안 취약성 평가시스템이다.

TSVES 구현 프로토타입에 이용된 TOE는 침입차단시스템으로 국내·외적으로 현재까지 가장 많이 개발 보급되었고, 새로운 모델의 개발과 평가 인증 지연이 많음을 고려하였다.

또한, 본 논문의 평가 자동화 기술은 다른 TOE의 평가 자동화에 응용되어질 수 있으므로, 국내·외 정보보호 시스템의 확고한 신뢰성 확보와 정보보호 시스템 활용 범위를 넓히고, 개발 촉진에 기여할 수 있다.

4.1 TSVES 주요 기능

TSVES의 주요기능은 다음과 같으며, 특히 TOE의 보안 취약성, 보안 기능 요구사항, 운영체제, 응용 프로그래밍 특성, 그리고 새로운 취약성 추가 등에 따른 시스템 운영 환경과 확장성 등을 고려하여, 다음과 같은 요구 사항을 충족시킬 수 있도록 설계 구현하였다.

(1) 개발자로부터 평가 의뢰된 TOE의 보안 기능과 오용에 대한 평가를 자동적으로 평가할 수 있으며, 평가 후 결과 보고서를 출력한다.

(2) TOE에 대한 평가는 CC의 평가 원칙에 따라 반복 평가가 필요하므로, 이 같은 반복적인 평가를 위해서 입력된 환경 설정 값 등을 재입력이 필요 없도록 한다.

(3) TOE가 제공하는 보안 기능에 대해서 그 기능의 정확한 동작 여부를 평가한다.

(4) 새롭게 발견되는 보안 취약성에 대한 평가 모듈을 쉽게 추가 할 수 있도록 하며, 평가 모듈 및 관련 정보 추가시 기존 프로그램에 쉽게 결합할 수 있도록 한다.

(5) 평가자의 편의를 위해 그래픽 사용자 인

터페이스(GUI:Graphic User Interface)를 제공한다.

(6) TSVES의 안정적인 구현을 위해 TOE를 침입차단시스템 보안 기능 취약성 평가에 제한하고 인쇄, 열람 등의 부가적인 기능은 기존의 편집기 등 다른 시스템 요소를 활용함으로써, 전체 시스템의 규모를 줄여 PC에서도 동작되어질 수 있도록 한다.

4.2 TSVES 설계

TSVES 구조는 모듈 단위로 구성되어 있다. 기본 구성 요소들은 TSVES의 실행 환경과 동작 제어, 모니터, TOE의 보안 기능별 평가 모듈, 그리고 취약성 정보 데이터베이스부분으로 구성되어 있다.

TSVES의 구성 모듈은 그림 2 와 같이 보안 기능별 평가 모듈을 실행시키기 위한 평가 기능 관리 모듈(EFMM:Evaluation Function Management Module), EFMM으로부터 전달된 평가 결과를 평가자가 편리하게 활용할 수 있는 형태로 제공하기 위한 평가 결과 관리 모듈(ERMM: Evaluation Report Management Module), TOE의 보안 기능과 보안 취약성 등을 관리하는 TOE 정보 관리 모듈(TIMM:TOE Information Management Module), 보안 기능 평가 모듈(SFEM:Security Function Evaluation Module)들, 그리고 보안 기능들과 기능별 취약성 정보를 저장하고 있는 취약성 정보 데이터베이스(VID : Vulnerability Information Database)로 구성된다.

(1) 그래픽 사용자 인터페이스 : 그래픽 사용자 인터페이스는 TOE의 보안 기능 및 보안 취약성 평가 실행 상황을 모니터 할 수 있는 표시 창, 그리고 평가를 위한 평가 값 설정 등의 기능을 제공한다.

(2) EFMM(평가 기능 관리 모듈) : EFMM은 TIMM과 VID로부터 받은 정보의 내용에 따라 TOE의 보안 기능 별 평가 모듈을 동작시킨다. 또한 EFMM은 현재 평가 진행 상황을 사용자에게 알리고, 평가자는

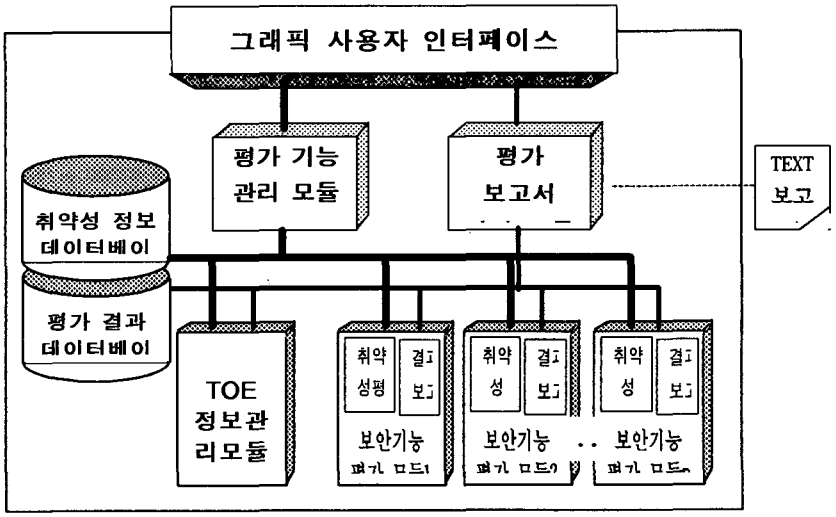

그림 2. TSVES의 기본 구조

원하는 임의의 시점에 동작을 중지시킬 수 있도록 한다.

(3) TIMM(TOE 정보 관리 모듈) : TIMM은 TOE의 정보를 관리하는 모듈이다. 이 모듈은 TOE의 평가 정보 목록을 각 SFEM에게 전달하며, SFEM은 전달받은 TOE의 보안 기능과 취약성을 평가한다. 평가 정보 목록은 평가 의뢰자로부터 기계가 읽을 수 있는 파일 형태로 받는다.

(4) VID(취약성 정보 데이터베이스) : VID는 TOE에 대한 세부적인 평가 취약성, 그리고 각 보안 취약성과 관련된 세부적인 평가 항목 즉, FTP, telnet, HTTP 등 각 프락시의 기능 정보와 취약성 정보들을 저장 관리한다.

(5) ERMM(평가 보고서 관리 모듈) : ERMM은 SFEM이 실행되는 동안 TOE와 TSVES, 그리고 대상(target) 서버로부터 받은 평가 결과 값을 평가 모듈별 보고서 파일로 출력하는 기능을 수행한다. 또한, 평가 결과를 비교할 수 있도록 보고서 파일에 추가할 수 있는 기능을 선택할 수 있도록 하였다.

(6) 보안 기능별 평가 모듈(SFEM)

1) FTP 프락시 평가 모듈

① 사용자 인증 : 특정 정보 시스템에 대한 서비스 허용 여부를 검사한다.

② 접근제어 : 출발지와 목적지에 따라 접근 허용 여부를 평가한다.

③ GET 명령어 허용 : 시작 호스트에서 목적 호스트에 접속 후 파일전송을 제한하는지 여부를 점검하고, FTP 접속 후 파일 전송 시 GET, MGET에 의한 파일 다운로드 가능 여부 및 PUT, MPUT 명령어에 의한 파일 업 로드 차단 여부 등을 점검한다.

④ PUT 명령어 허용 : 시작 호스트에서 목적 호스트에 접속 후 파일전송 제한 여부를 점검한다.

⑤ 세션 타임아웃 : 시작 호스트에서 목적 호스트로 FTP를 이용하여 접속 후 아무 입력 없이 관리자에 의해서 설정된 시간이 경과되면 접속이 종료되는지 여부를 점검한다.

⑥ FTP 명령어 제한 : 시작 호스트에서 목적 호스트에 접속 후 디렉토리 생성 및 삭제, 파일 삭제 등의 명령어를 제한하는지 여부(cd, mkdir, rename, delete.... 등)를 점검한다.

2) HTTP 프락시 평가

① 사용자인증 : 특정 정보 시스템에 대한 서비스 여부를 점검한다.

② 접근제어 : 출발지와 목적지에 따라 접근 허용 여부를 점검한다.

③ method별 제어 : GET, PUT, POST, CONNECT 별로 접근 제어가 발생하는지를 점검한다.

④ 특정 URL 필터링 시험 : 관리자가 지정한 URL, 특정 단어가 포함된 URL 차단 여부 등을 점검한다.

⑤ 내용 필터링 : JAVA, JAVA Script, Active-X, VBScript 코드의 웹 문서내 존재할 때 차단 여부 등을 점검한다.

4.3 TSVES 기능 흐름

TSVES의 기능 흐름은 그림 3 과 같이 평가자가 GUI를 통해서 TOE에 연결하고, EFMM을 통해 SFEM을 지정한 뒤, SFEM에 평가를 시작하도록 지시하면 SFEM은 해당 모듈의 보안 기능과 취약성 평가를 실행한다.

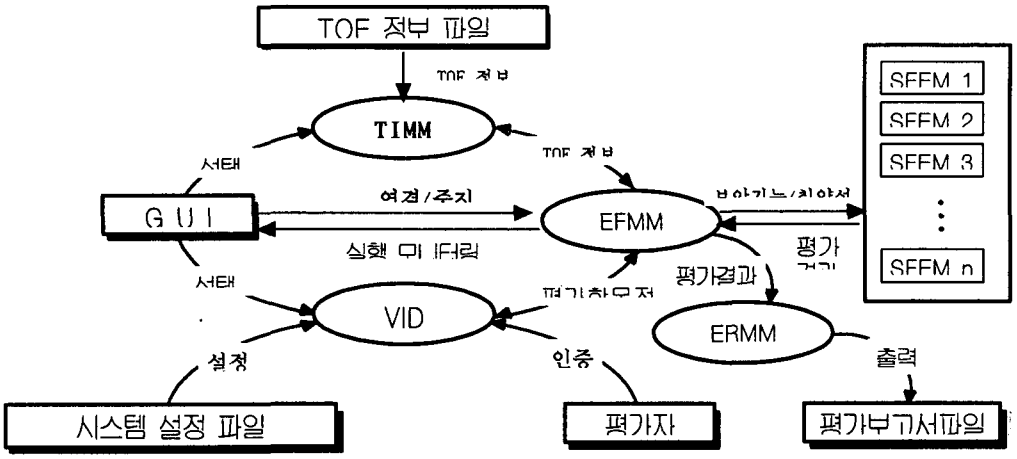

그림 3. 기능 흐름도

평가 결과는 각각 SFEM 실행 후 보고서 파일로 출력되며, 평가가 끝난 후 평가자가 결과를 열람하려면, 시스템 내의 편집기를 통해 결과 TEXT 파일을 열람할 수 있다.

Ⅳ. TSVES 구현 프로토타입

그림 4는 HTTP 프락시 평가 시 프락시의 IP, TOE를 경유하는 서버의 URL, 포트 번호, 전송 받을 파일, JAVA, Active-X 등에 대한 선택사항을 입력으로 받는다. HTTP 프락시 평가 모듈이 실행된 다음 그림 5 와 같이 평가 항목에 대한 결과를 얻게 된다.

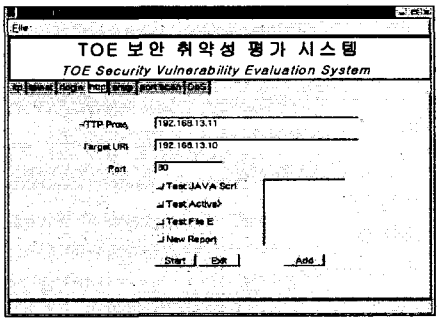

그림 4 HTTP 프락시 평가를 위한 GUI

여기서는 IP 주소가 192.168.13.11인 호스트를 대상으로 평가가 이루어졌으며, 그림 5 는 평가 후 생성된 192.168.13.10_http_report 라는 보고서 파일 내용이다.

보고서 파일의 내용처럼 IP주소가 192.168.13.11인 HTTP Proxy에서 URL 192.168.13.10의 필터링이 제대로 이루어지지 않았음을 보여준다.

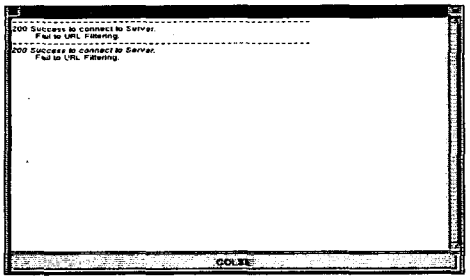

그림 5 HTTP 프락시 평가 결과

TOE 취약성 평가 방법과 기존의 평가 방법을 CEM 평가 원칙 만족도와 실용성 그리고 경제성 요소로 비교하면 <표 1>에서 알 수 있듯이, 기존 평가 방법들은 단편적인 보안 기능과 취약성을 분석하고 평가할 수 있지만, 각각이 가지고 있는 문제점 이외에도 취약성 평가 모델의 전 단계를 동시에 평가하여 결과를 평가자에게 제공할 수 없으므로 평가자는 평가 진행에 따른 결과를 모두 기록하여야 한다.

그러나 본 논문에서 제안한 모델을 적용한 평가 절차는 자동화 도구 활용을 기본으로 한 방법으로 보안 기능 강도 시험, 오용, 그리고 침투 시험 등에 효과적이며 보고서 생성 기능을 제공하고 있다. 특히, 본 논문은 CC의 평가 원칙과, 실용성을 충족시키고 있으며, 평가 자 수와 시간 면에서 탁월한 효율성을 보임으로써 경제성을 입증하였다.

한편 TSVES의 실용성을 더욱 높이기 위해서는 취약성에 대한 충분한 시나리오와 TOE의 기능에 따라 평가를 달리할 수 있는 스크립트 구성 기능이 요구된다 할 수 있다.

V. 결론

정보통신망 환경이 전 세계적인 단일 정보통신망 환경으로 진화되어 가는 기술추세에 각종 컴퓨터 범죄, 해킹 또는 불건전 정보 유통 현상은 더욱 심각해 질 것이고, 이에 따른 정보보호 제품의 수요도 폭발적으로 증가할 것으로 전망된다.

〈표 1〉 취약성 평가 방법 비교

구 분		매뉴얼	SATAN	ISS	해킹 도구	TSVES
평가 시간		매우 길다	짧다	짧다	짧다	짧다
평가자 수		다수	소수	소수	소수	소수
보안 기능 시험		보통	양호	양호	양호	양호
보안 기능/강도		보통	양호	양호	양호	양호
오용 시험		보통	불량	불량	양호	양호
침투 시험	내부	보통	불량	불량	양호	양호
	외부	보통	불량	불량	양호	양호
시험 동시성		불가능	불가능	불가능	불가능	가능
보고서 출력		불가능	가능	가능	불가능	가능
상한 값 검사		불가능	·	·	·	가능
전문지식		고급	중급	중급	고급	하급
개발자 이용성		가능	가능	가능	가능	가능
사용자 이용성		불가능	가능	가능	불가능	가능

따라서 정보보호 제품에 대한 보안기능과 신뢰성이 검증되어져야 할 필요성이 사용자로부터 요구되어지고 있으며, 사용자의 보호수준에 적합한 제품의 목록이 필요한 실정이다. CC가 ISO로부터 표준 평가 기준으로 인정됨으로써 국가별 평가 기준 및 평가 체계는 CC을 따르게 될 것이며, 사이버 범죄 증가와 각종 정보보호 제품에 대한 개발이 활발하게 진행되면서 더욱 빠르게 확산되어질 것이다.

이러한 정보보호 제품의 보안 인증을 위한 평가는 공정성과 객관성 그리고 신속한 평가 진행이 가장 중요한 평가 원칙이 될 수 있다. 이를 위해서는 기존 수작업 중심의 평가 방법을 최대한 자동화 할 수 있는 방법이 필요하고, 본 논문은 이를 실현할 수 있도록 TOE 보안 취약성 평가 방법을 제안하고, 평가 과정을 자동화한 TSVES를 설계 구현하였다.

TSVES는 정보보호 제품의 보안관리 등에 대한 평가 과정과 오용에 대한 시험을 자동화함으로써 제품에 대한 평가 소요 시간을 크게 단축하고, 보다 신뢰할 수 있는 정보보호 제품 보증을 유지할 수 있도록 하였고, 평가 가상 시나리오를 작성하고, 실 데이터에 준해서 시험과정을 진행함으로써 안정적이고 실용성을 높였으며, TOE의 보안 기능에 따른 모듈화 실현으로 향후 확장이 용이하도록 하였다. 또한, 평가의 신뢰성을 높이기 위하여 반복적인 평가가 용이하도록 하였고, 출력 결과를 평가자가 평가 보고서에 활용할 수 있도록 텍스트화 하여 응용 서비스별 단위 파일로 구성하였다.

참고문헌

[1] Common Criteria Editorial Board, Common Criteria for Information Technology Security Evaluation, Part 1 : Introduction and General Model , Version 2.0, Oct., 1997.

[2] Common Criteria Editorial Board, Common Criteria for Information Technology Security Evaluation, Part 2 : Introduction and General Model , Version 2.0, Oct., 1997.

[3] Reto E. Haeni, "Firewall Penetration Testing", The George Washington Univ. Cyberspace Policy Institute, 1997

[4] David A Curry, "UNIX System Security : A Guide for Users and System Administration", Addision-Wesley, 1992

[5] Karanjit S. Siyan & Chris Hare, "Internet Firewalls and Network Security", NRP, 1995

[6] Rovert B. Reinhardt, "An Architectural Overview of UNIX Network Security", 1994

[7] Michel E. Kabay, "The NCSA Guide to Enterprise Security Protecting Information Assets", 1996

[8] William R. Cheswick Steven M.Bellovin, "Firewalls and Internet Security", Addison-Wesley Publishing Co., 1994

[12] http://www.certcc.or.kr/paper/tr1999/1999002/Docs/tr1999002.html

[13] http://jazz.snu.ac.kr/~junker/sysadmin/security.html

[14] http://www.cs.jmu.edu/users/AbzugCX/Trusted-Systems/Common

[15] http://csrc.ncsl.nist.gov/cc

Vulnerability Evaluation Methodology for firewall Certification

Jeom-Goo Kim*

Abstract

Although an insurance criterion for reliance is unified, the difference in evaluation mechanism in every country is already ambiguity. In other words, the aspect of objectivity would be a little because it is true that the vulnerability evaluation include evaluator, scenario and tools (TSVES) applying in test, depth and scope of evaluation. But evaluation results can be difference in accordance with each evaluation elements.

By using TSVES to evaluating network security vulnerability, first, we expected the evaluation results is impartiality, objectivity, repeatability, reproducibility, appropriateness and soundness of results. Second, it could be transferred manual ways into automation ways, and then expected easiness and safety of extension and modification in a quality of products as well as a dramatical reduction of waste of time and energy.

* Dept. of Computer Science, Nam Seoul University