情 報 保 護 學 會 誌
第 13 券 第 6 號, 2003. 12

정보보호시스템 시험 커버리지 보증을 위한 시험 케이스 연구

윤 여 웅*, 안 재 영*, 서 동 수**

요 약

정보보호시스템 평가·인증 제도는 사용자들이 신뢰된 인증제품을 사용할 수 있도록 평가기준의 등급별 요구사항을 근거로 평가하여 정보보호시스템에 구현되어 있는 보안기능의 신뢰성을 보증하는데 있다. 본 연구에서는 정보보호 제품의 시험 커버리지 보증을 위한 시험케이스 생성에 관한 연구로서, 평가기준에서 요구되는 시험 커버리지를 분석하고 이를 만족시키기 위한 시험케이스 생성 방법을 제시함으로써 보증성을 향상시키고자 한다.

Ⅰ. 서 론

최근 정보통신의 비약적인 발전에 따른 정보화의 가속화는 사용자가 필요로 하는 유익한 정보를 실시간으로 제공하지만, 개인 및 국가사회의 중요정보가 정보통신망을 통한 다양한 불법적인 침입 및 공격 등으로 위협받고 있다. 이에 대응하기 위한 대책으로 정보보호시스템의 사용이 점차 증가하고 있으며, 사용자들이 안심하고 신뢰된 정보보호시스템을 사용할 수 있도록 정보보호시스템의 평가·인증 제도를 도입하여 시행하고 있다.

국내에서도 1998년 침입차단시스템에 대한 평가를 시작으로 다양한 정보보호시스템에 대한 평가·인증을 시행하고 있으며, 정보보호시스템 내에 구현된 보안기능의 신뢰성을 보증함으로써 개인, 기업, 공공기관 등에서 안심하고 신뢰된 제품을 사용할 수 있는 환경을 조성하고 있다.

정보보호시스템을 보증하는 방법은 검증(Verification)과 확인(Validation)이 있으며, 검증은 요구사항으로부터 구현된 시스템 코드까지 개발과정에서 산출된 문서를, 확인은 개발된 정보보호시스템을 실행시켜 시스템이 올바르게 작동한다는 것을 시험을 통하여 보증하는 것이다[1]. 국내에서도 정보보호시스템의 검증 및 확인을 위하여 평가등급에 따라 개발과정에서의 산출물과 시험에 관련된 시험서, 시험도구 및 프로그램들을 제출할 것을 요구하고 있다.

확인은 시험을 통하여 결함(Defects)이 없음을 보증하는 것이다. 하나의 결함은 시스템의 여러 다른 부분에 영향을 미칠 수 있다. 결함은 특히 보안시스템, 금융시스템, 국방시스템 등과 같은 중요 시스템(Critical Systems)들을 큰 재앙에 빠뜨릴 수 있고 비중요 시스템에 대해서는 사용자들의 불평을 낳게 할 수 있다. 결함은 명세로부터의 부정확한 구현, 구현된 시스템 내에 명세된 요구사항의 누락, 명세되지 않은 요구사항들의 구현 등으로부터 발생된다. 정보보호시스템이 요구사항에 따라 정확하게 구현되었음을 확인하기 위하여 시험케이스(Test Case)에 대한 설계가 무엇보다 중요하다. 설계된 시험케이스가 정보보호시스템의 보안기능 전체를 포함한다는 것을 보증하고 시험을 통하여 결함이 없다는 것을 보증해야만 신뢰할 수 있다.

본 연구에서는 시험 커버리지(Test Coverage) 분석 및 시험케이스 생성 기법을 알아보고 정보보호시스템 평가기준의 평가등급에서 요구되고 있는 시험 커버리지를 분석한다. 또한, 정보보호시스템 시험 커버리지를 만족시키는 시험케이스 생성 방법과 예를 제시함으로써 정보보호시스템 보증 및 신뢰성을 향상시키고자 한다.

* 한국정보보호진흥원 산업지원단 평가1팀 ({ywyun, jyahn}@kisa.or.kr)
** 성신여자대학교 컴퓨터정보학부 (dseo@sungshin.ac.kr)

Ⅱ. 시험케이스 생성 기법 분석

1. 시험 개요

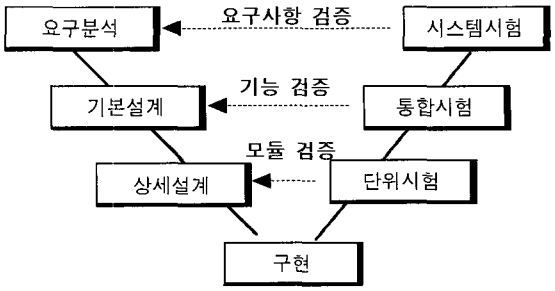
시험은 시스템 내의 결함을 발견하는 모든 행위를 말하며 개발 과정 내에서 결함방지 기법(Defect Preventive Method)과 함께 병행적으로 이루어져야 한다. 결함방지 기법은 프로토타입을 만들거나 코드 검토 등을 수행함으로써 실현될 수 있다. 시험은 버그들을 발견하거나 정확성을 증명하는 것이고 Glen Myers는 "시험은 에러들을 발견하려는 의도를 가지고 프로그램을 실행시키는 절차"이며, Dijkstra는 "시험은 버그가 없다는 것을 증명할 수 없으며, 버그가 존재한다는 것을 증명할 수 있다"라고 정의했다[2].

많은 사람들은 시스템이 정확하게 동작된다는 것을 증명하기 위하여 모든 조합들에 대해 시험해야 한다고 생각하지만 모든 입력, 반복 횟수, 시간과 같은 타이밍 문제 등으로 인하여 모든 조합에 대하여 시험할 수 없다[3]. 따라서, 시험을 통해 발견된 버그의 수는 시스템 내에 남아있는 버그의 수와 비례된다고 볼 수 있으며, 결함을 감소시키는 것은 기하급수적으로 더 많은 노력이 요구되며, 결과적으로 모든 결함을 제거한다는 것은 사실상 불가능하다고 볼 수 있다[4]. 시스템 시험에서 발견될 수 있는 결함은 사용자 인터페이스 에러, 경계값 처리 에러, 산술 및 논리 계산 에러, 초기상태 및 전이 상태 처리 에러, 제어흐름 에러, 데이터 처리 및 해석 에러, 한계값 처리 에러 등이 발생될 수 있으며[5], 이러한 결함을 발견하기 위해 철저한 시험이 요구된다.

2. 시험 프로세스 및 환경

전통적인 시험 프로세스(Test Process)는 코딩이 완료된 후에야 비로소 시험을 시작할 수 있기 때문에 결함을 발견할 수 있는 시간이 충분하지 못했다. 이를 극복하기 위하여, 제안된 병행 시험은 시험 계획 및 시험케이스 설계를 시스템 개발과 병행적으로 수행함으로써 개발과정에서 발생할 수 있는 버그들을 발견할 수 있다. 따라서, 시스템 개발과정에 따른 시험 과정을 표현하면 〔그림 1〕과 같다.

시스템 검증 및 확인은 위 그림처럼 왼쪽의 개발 프로세스를 통하여 요구사항 분석, 기본설계 및 상세설계 검토를 통하여 검증을 수행하고 오른쪽에 시험 프로세스를 통하여 요구사항, 기능 및 모듈 등에 대한



〔그림 1〕 개발 및 시험 프로세스

시험을 통하여 시스템 정상적으로 동작됨을 확인한다. 또한, 개발 프로세스 동안에 시험환경을 지원하기 위한 시험케이스 설계 및 시험에 필요한 프로그램을 개발하게 된다. 시험 환경은 시험 프로세스를 지원하기 위하여 시험 중에 시스템 주변에 설치된 장치들로써 아래의 사항들을 포함한다.

- 시험되어야 할 시스템 또는 컴포넌트
- 수행될 시험케이스들과 결과를 포함하는 시험 문서
- 시험을 위하여 시스템을 제어하는데 사용될 수 없는 시스템의 일부를 대체하는데 사용되는 시험하니스(Test Harnesses)와 시험스텁(Test Stubs)
- 시험케이스의 결과를 예상하거나 점검하는데 사용되는 시험오라클(Test Oracles)
- 시험케이스의 입력으로 사용되는 데이터파일들을 포함하는 시험데이터

3. 시험 기법

시험 기법은 시스템 상에 존재할 수 있는 여러 종류의 에러들을 발견하는데 사용될 수 있는 많은 종류의 시험 방법들이다. 이러한 방법들을 시스템의 특성 및 발견하고자 하는 에러들에 따라 선택하여 사용할 수 있다. 또한, 기술된 시험 기법 하나만을 적용하기보다는 여러 기법들을 병행적으로 사용함으로써 시스템의 신뢰성을 증가시킬 수 있으며 적용할 수 있는 대표적인 시험 기법을 간략히 기술하면 아래와 같다.

3.1 정적/동적 시험

정적 시험은 코드 실행 없이 컴포넌트나 시스템을 검사하는 것으로 단지 코드만을 조사하는 것이 아니라 요구사항 명세, 기본/상세 설계, 사용자 문서 등의 조사를 병행한다. 동적 시험은 자체적으로 코드를 조사하지 않고 시스템이 정확히 동작하는지를 확인하기 위

하여 시스템 코드를 직접 수행시킨다.

3.2 기능(블랙박스) 시험

기능 시험은 시험하는 시스템이나 컴포넌트의 설계 및 구현 구조를 볼 수 없기 때문에 시스템이나 컴포넌트의 기능요구사항에 기반한 시험으로 기능요구사항에 대한 문서들로부터 시험케이스를 생성할 수 있으며, 각각의 기능요구사항을 시험한다. 기능요구사항에서의 시험 커버리지는 주로 기능요구사항과 시험기준간 매트릭스를 작성함으로써 만족될 수 있다. 기능요구사항에 대한문서가 없다면, 시스템이나 컴포넌트의 운영 행위들을 조사함으로써 시스템의 요구사항을 역으로 해석해야만 한다.

3.3 구조(화이트박스) 시험

구조 시험은 시스템이나 컴포넌트의 설계 및 구현 구조에 기반한 시험 방법으로 내부 구조를 기반으로 제어 및 정보의 흐름을 분석하여 시험케이스를 생성한다. 구조 시험의 시험 커버리지는 주로 시스템이 수행될 때 프로그램의 경로를 분석함으로써 확인할 수 있으며, 프로그램 내의 모든 문장이 컴포넌트나 시스템의 시험 동안에 수행되는 것을 점검하면 된다. 즉, 전체가 시험되었다는 것은 시험 동안에 수행되지 않는 코드가 하나도 없다는 것을 의미한다.

3.4 모듈(단위) 시험

모듈(단위) 시험은 시험하는 단위가 시스템의 각 모듈 수준으로 모듈이 다른 모듈들과 통합되기 전에 수행하는 시험이다. 이 수준에서의 시험은 구조 시험과 일치한다. 모듈 시험은 단위 시험이라 불리는데 단위(Unit)는 시험하는데 사용될 수 있는 가장 작은 수준의 컴포넌트를 의미한다.

3.5 통합 및 시스템 시험

통합 시험은 통합된 서브시스템의 상호작용 및 일치성을 시험하기 위하여 모듈 시험과 시스템 시험 사이에서 적용되는 중간수준의 시험이다. 통합 시험은 모듈이 더 큰 서브시스템으로 통합되어 점진적으로 적용될 수 있으며, Bottom-up, Top-down 또는 두 가지 방식을 혼용한 방식을 이용한다. 이 시험에서는 기능시험과 구조 시험을 혼용해서 수행한다.

시스템 시험은 완전히 시스템이 통합된 후에 시험을 수행하며, 주로 모든 모듈 시험과 통합 시험이 성공적으로 끝난 후에 수행한다.

3.6 인터페이스 시험

인터페이스는 주로 다른 사람들에 의해 개발되기 때문에 컴포넌트와 서브시스템을 작동시키는 방법에 대한 오해를 가져올 수 있기 때문에 많은 에러들이 발생할 수 있다. 인터페이스 시험은 컴포넌트들과 서브시스템들이 서로 연결되고 동시에 작동하는 방식에 집중되어 있으며, 시스템의 서브컴포넌트간 내부 인터페이스와 다른 시스템으로의 외부 인터페이스까지도 적용될 수 있다.

3.7 인수/성능/부하 시험

인수 시험은 시스템이 미리 정의된 인수기준을 만족시키는지 사용자에게 증명하는 것이다. 전형적으로 시스템을 인수하기 위한 것으로 사용자는 인수 시험 기준을 작성하고 개발자에게 인수 기준을 만족할 것을 요구한다.

성능 시험은 시스템 운용 면에서 시간 및 메모리 사용 등과 관련된 것으로 한 동작이 정해진 한계값 내에 완료될 때 할당된 메모리 크기를 점검한다. 부하 시험은 시스템에 과부하를 제공하여 시스템이 어느 수준까지 운영에 문제가 발생하지 않음을 시험한다.

4. 시험 기법을 통한 시험케이스 생성

위에서 살펴본 시험 기법을 통하여 시험케이스를 생성하기 위하여 시험 커버리지에 대한 분석이 필요하다. 이 절에서는 구조 시험 및 기능 시험에 대한 시험 커버리지 분석과 이를 통한 시험케이스 생성 방법을 기술한다.

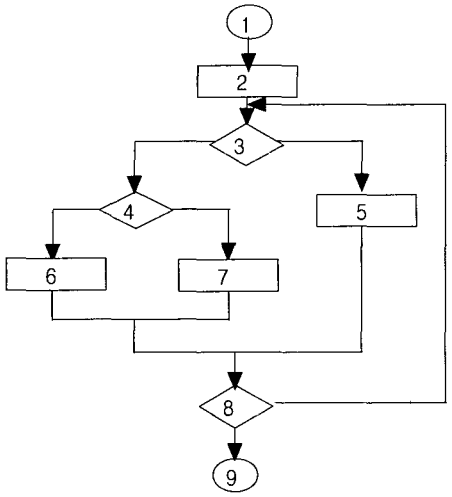
4.1 구조 시험 시험케이스

구조 시험은 시스템의 모든 경로를 실행하는 것이나 모든 경로를 수행하는 것이 불가능하다. 따라서, 시스템의 식별된 모든 경로를 시험하기보다는 시험을 위한 특정 경로들을 선택하는 방법을 제시한다. 구조 시험은 시험 커버리지 매트릭스와 많은 관련이 있으며 이 매트릭스를 통하여 시험케이스에 의해 수행된 경로를 측정하게 된다. 구조 시험은 시스템의 소스코드(프로그램) 수준에서 적용되기 때문에 코드 구조를 알 수 있으며 시험 중에 어떤 코드가 수행되었는지를 알 수 있다. 구조 시험을 지원하는 시험 커버리지에 대한 분

석 방법 중 대표적인 것으로는 문장(Statement) 커버리지, 분기(Branch) 커버리지, 경로(Path) 커버리지가 있다. 이러한 시험 커버리지들은 시스템 코드의 논리적인 흐름에 대한 분석으로부터 시작되며, 프로그램의 논리적인 흐름은 흐름도에 의해 표현될 수 있으며[4][6], 아래의 개괄적인 코드에 대한 흐름도는 [그림 2]와 같다. 또한, 시험케이스는 프로그램의 선택된 문장들을 수행하도록 특정 경로를 선택하고 다른 경로들은 분기 조건을 변화시킴으로써 선택될 수 있다.

◦ 시험케이스의 개괄적 코드 구조

```
1
2
if 3 then
        if 4 then
            6
        else
            7
        end if;
else
        5
end if;
if 8 then
        9
else
        3
end if;
```



[그림 2] 흐름도의 예

4.1.1 문장 커버리지 분석

문장 커버리지는 제안된 시험케이스들에 의해 수행된 문장들의 비율을 평가하는데 소스코드나 프로그램 내의 문장이 적어도 하나의 시험에 의해 수행된다. 문장 커버리지는 [그림 2]의 1~9까지의 블록들을 모두 수행하면 커버리지 100%가 된다. 실제적으로 문장 커버리지를 분석할 때는 흐름도에 정의된 하나의 노드가 여러 문장으로 구성될 수 있기 때문에 1,2,3,5,8,9 경로는 문장 커버리지 66%에 미치지 못할 수도 있다.

4.1.2 분기 커버리지 분석

분기 커버리지는 시험케이스들에 의해 수행된 분기의 비율을 평가함으로써 결정된다. 100% 분기 커버리지는 프로그램 내의 모든 분기가 적어도 하나의 시험에 의해 수행된다는 것이다. 분기 커버리지는 분기된 노드를 실행시키는 것과 같기 때문에 각 분기마다 2가지가 발생한다. [그림 2]의 흐름도에서 3개의 분기가 있고 하나의 시험케이스를 통하여 1,2,3,5,8,9의 경로를 수행했다면, 33%의 분기 커버리지이다.

4.1.3 경로 커버리지 분석

경로 커버리지는 제안된 시험케이스들에 의해 수행된 실행 경로의 비율을 평가함으로써 결정되며, 100% 경로 커버리지는 프로그램 내의 모든 경로가 적어도 하나의 시험에 의해 수행된다는 것이다.

경로 커버리지의 문제는 루프 구문이 포함된 경우이며, 루프는 모든 경로에 대한 시험을 불가능하게 하는 요소이다. 루프 구문을 포함한 프로그램의 경로 커버리지 분석을 위하여 기본 경로(Basis Path)의 개념을 도입하고 실행 경로의 기본 집합을 정의하기 위하여 회귀성 복잡도(Cyclomatic Complexity)를 계산해야 한다. 복잡도는 "그래프에서의 전이수 - 노드수 + 2"로 나타낼 수 있다. 계산된 복잡도는 모든 프로그램 문장 커버리지를 보증하기 위해 수행되어야 할 최대 시험수이다. 경로 커버리지는 적어도 한번은 프로그램의 모든 문장과 모든 분기가 수행할 수 있는 경로들의 집합을 포함하지만 경로 커버리지 집합은 유일하지 않다.

루프가 프로그램 내에 포함된 경우, 시험케이스를 생성할 때 루프 우회, 1번, 2번, 최대, 최대+1번, 최대-1번, 최소, 최소+1번, 최소-1번, 널값 수행, 음수를 수행하는 것에 대하여 고려해야 한다[7].

[그림 2]에서 회귀성 복잡도는 4이며, 4개의 경로들이 있는데 하나의 시험케이스가 1,2,3,5,8,9의 경로를 수행했다면, 25%의 경로 커버리지를 갖는다.

4.1.4 커버리지 비교

구조 시험은 시험케이스들을 통한 특정 경로들을 실행시켜 프로그램 내의 실행 경로를 분석하는데 유용한다. 위에서 분석된 커버리지 기법들은 프로그램 흐름과 분기의 특성을 갖는 흐름도나 상태전이도와 같은

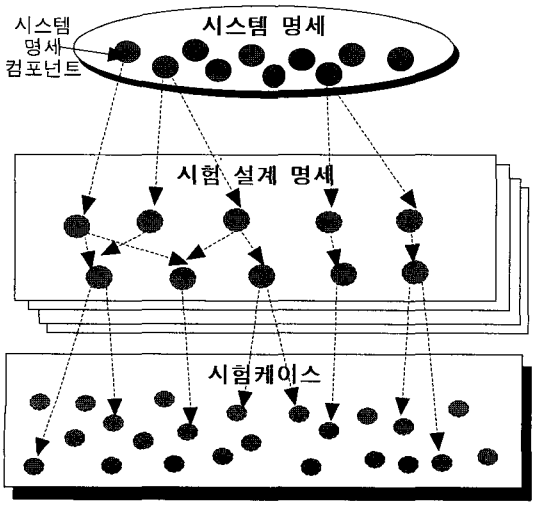
명세에 적용될 수 있으며, 커버리지 기법간에는 관계가 성립된다. 100% 분기 커버리지 없이도 100% 문장 커버리지를 가질 수 있는데 모든 분기 전이 수행없이도 모든 노드들은 실행될 수 있다는 의미와 같다. 또한, 100% 경로 커버리지 없이도 100% 분기 커버리지를 가질 수 있는데 모든 경로 수행없이도 모든 분기 전이들은 수행될 수 있다는 의미와 같다. 따라서, 100% 경로 커버리지는 100% 분기 커버리지도 당연히 포함하고, 100% 분기 커버리지는 100% 문장 커버리지를 포함한다.

4.2 기능 시험 시험케이스

기능 시험은 블랙박스 시험, 명세 기반 시험이라고도 하며, 시스템 또는 컴포넌트의 내부 구조에 대한 지식 없이 시스템 요구사항에 대한 명세를 기반으로 시험을 수행한다. 기능 시험은 명세로부터 시스템이 수행하는 기능을 분석하고, 명세를 참고할 수 없는 경우는 시스템의 메뉴 등을 분석하거나 사용자 매뉴얼을 이용하여 시스템의 기능에 대한 명세를 작성할 수도 있다. 기능 시험을 위한 시험케이스 설계는 〔그림 3〕과 같이 시험하려는 시스템 명세 컴포넌트들을 분해하여 시험하려는 목적에 따라 시험 설계를 위한 명세를 작성하고 작성된 시험 설계 명세로부터 시험케이스를 생성할 수 있으며, 동치분할 커버리지, 경계 커버리지, 상태전이 커버리지가 대표적이다[6].

4.2.1 동치분할 커버리지 분석

동치분할 커버리지는 컴포넌트의 입력과 출력이 비슷한 영역으로 분리될 수 있으며, 컴포넌트들의 명세에 따라 컴포넌트에 의해 비슷하게 처리된다는 전제에 기반한다. 이 가정은 동일하거나 비슷한 입력 영역은 비슷한 루틴에 의해 처리된다는 것이다. 동일한 입력 영역의 시험데이터를 실행시켰을 때 오류가 발생하지 않았다면 그 영역의 다른 시험데이터는 오류를 발생하지 않을 것이다.



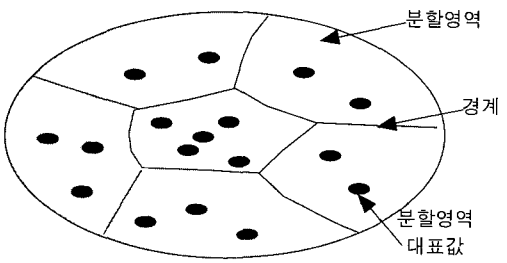
〔그림 3〕 시스템 명세로부터 분할 과정

동치분할 커버리지는 동일한 영역의 모든 입력데이터를 시험하는 것이 불가능하다는 문제를 극복하기 위하여 그 영역의 대표값을 이용한다. 동치분할로 동일하게 처리될 수 있는 영역은 유효하거나 무효한 입력값 및 출력값, 음수, 양수, 0의 값을 갖는 숫자, Empty 및 Non-Empty 문자열, Empty 및 Non-Empty 리스트 등이 있다.

분할을 분석하는 과정은 명세 컴포넌트의 입력 및 출력의 분석으로부터 시작되며 유효한 값뿐만 아니라 무효한 값에 대한 분할도 필요하다. 분할된 동일 영역의 모든 값들은 〔그림 4〕에서와 같이 동일한 컴포넌트에 의해 동일한 방법으로 처리된다. 시험케이스들은 분할을 수행함으로써 생성될 수 있으며 각 시험케이스는 컴포넌트로의 입력, 수행된 분할영역, 예상 결과 등을 포함한다. 분할을 수행하기 위한 시험케이스 생성 방법에는 두 가지 방법을 사용할 있으며 One-to-One 방법과 최소화된 시험케이스 방법이 있다. One-to-One 방식은 각 분할에 대해서 다른 분할의 값을 변경시키면서 시험하는 방식이고, 최소화된 시험케이스 방법은 여러 분할을 포함하도록 시험하는 방식으로 시험케이스들간에 중복이 있을 수 있다. One-to-One 접근법의 단점은 너무 많은 시험케이스들을 생성한다는 점이며 최소화된 시험케이스는 오류 및 실패가 발생했을 때원인을 찾기가 어렵다. 그러나, 시험케이스의 생성 및 시험하는 것은 분할을 정의하는 것보다 훨씬 쉽다.

4.2.2 경계 커버리지 분석

경계 커버리지는 분할의 경계에 대한 분석을 추가



〔그림 4〕 동치분할 커버리지의 분할 영역

적으로 요구함으로써 동치분할을 확장하는 것이다. 동치분할과 같은 가정을 가지며 개발자들이 분할의 경계값을 처리하는데 에러들을 유발하기 쉽다는 점에 중점을 둔다. 어떤 입력값의 조건이 주어졌을 때 그 조건의 경계값에 대한 처리가 누락될 수 있다. 예를 들어, 투표권을 가진 사람인지 확인하기 위해 나이를 입력받을 때, 20세 이상인지를 확인하게 된다. 나이의 경계 커버리지 분석은 19,20,21에 대한 분석이 필요하다. 또한, 나이가 음수일 수 없기 때문에 음수에 대한 무효한 값도 처리해야 한다. 경계값의 예는 일주일의 월요일과 일요일, 1월과 12월, 16비트 정수의 32767과 -32768, 스크린의 왼쪽 위와 오른쪽 아래커서 포인터, 프린트된 첫 번째/마지막 라인, 한문자 스트링과 최대길이 스트링 등과 같다.

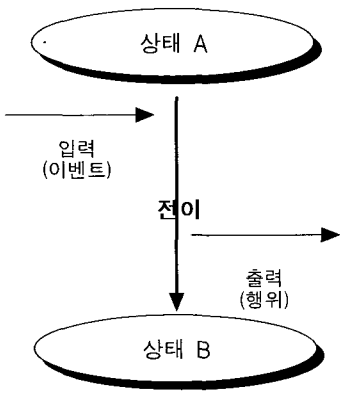
4.2.3 상태전이 커버리지 분석

상태전이 커버리지는 상태전이에 의한 시스템의 행위를 모델화하기 위해 컴포넌트에 대한 명세를 분석한다. 분석을 통하여 상태 모델을 작성하는데 작성된 상태 모델은 상태전이 다이어그램으로 표현되며, [그림 5]와 같이 시스템의 상태들, 상태들간 전이들, 전이를 발생시키는 이벤트들, 결과 행동들로 구성된다.

상태전이 커버리지의 시험케이스들은 시스템의 상태들간 유효한 전이를 실행하도록 설계된다. 또한, 명세되지 않은 전이가 유발되지 않도록 추가적인 시험케이스도 작성되어야 한다.

Ⅲ. 평가등급별 시험케이스 생성 방법

정보보호시스템 평가기준은 각 등급별로 보안기능 요구사항과 보증요구사항을 명시하고 있으며 개발과정



[그림 5] 상태전이 모델

[표 1] 등급별 개발과정 및 시험과정 요구사항

구분	K2	K3	K4
개발과정 제출물	- 기능명세 - 기본설계	- 기능명세 - 기본설계 - 상세설계 (필요시, 일부 원시프로그램 및 검증명세)	- 기능명세 - 기본설계 - 상세설계 - 검증명세 - 원시프로그램
시험과정 제출물	- 시험서 ·각 시험항목별 시험목적 및 시험절차 ·예상결과와 실제 결과 - 시험프로그램 및 시험도구		

과 시험과정에서의 요구사항도 포함되어 있다. [표 1]에서는 평가등급 중 K2에서 K4의 요구사항들을 정리하였다[8].

[표 1]에서 보는 것처럼, 평가기준의 시험과정의 제출물은 K2에서 K4까지 시험서, 시험프로그램 및 시험도구의 동일한 항목을 요구하고 있다. 그러나, 시험수준은 엄격히 구별되며, 이는 [그림 1]에서 확인했듯이 개발과정의 제출물에 의해서 결정된다고 볼 수 있다. 본 절에서는 개발과정의 제출물별로 요구사항을 분석하고 등급별로 요구되는 시험 커버리지를 분석한다.

1. 개발과정 제출물별 요구사항 분석

1.1 기능명세

기능명세는 정보보호시스템이 제공하는 보안기능의 동작과 외부 인터페이스에 대하여 기술하여야 한다. 또한, 기능명세에서는 외부인터페이스의 사용 목적 및 방법 등을 서술하고 효과, 예외사항, 오류 메시지 등에 대한 세부사항을 제공해야 한다[9][10]. 서술된 보안기능의 동작과 외부 인터페이스는 시스템을 이해하는 중요 요소로서 기능 시험을 위한 가이드라인을 제시해 준다. 이는 외부 인터페이스를 통한 입력 이벤트가 보안기능을 동작시키고 오류 등을 유발할 수도 있기 때문에 외부 인터페이스에 대한 시험이 매우 중요하다.

1.2 기본설계

기본설계에서는 시스템의 기본 구조와 인터페이스, 보안기능의 제공방법, 보안 및 비보안 서브시스템 분리방법 등을 기술하여야 한다. 또한, 하부 하드웨어, 펌웨어, 소프트웨어에 구현된 지원 보안메커니즘에 의해 제공되는 기능을 표현하여 시스템에서 요구되는 하부 하드웨어, 펌웨어, 소프트웨어를 식별해야 한다.

기본설계에서의 인터페이스는 기능명세에서 식별된 외부 인터페이스를 포함하고 서브시스템간 인터페이스까지도 기술되어야 한다. 각 인터페이스 설명에는 효과, 예외사항, 오류 메시지 등에 대한 세부 사항을 적절하게 제공하여 서브시스템에 대한 모든 인터페이스의 사용목적 및 방법을 서술해야 한다[9][10]. 즉, 전체 시스템을 주요 구성요소인 서브시스템으로 분해하여 서술하고 각 서브시스템은 기능명세에서 기술된 보안기능 동작이 포함되도록 서술하며, 서브시스템간의 상호관계를 동작중심으로 명확히 서술하여야 한다.

1.3 상세설계

상세설계에서는 기본설계에 서술된 모든 보안기능의 상세설계 내용, 보안 메커니즘 및 보안기능 제공방법들을 기술하여야 한다. 즉, 각 서브시스템을 여러 개의 모듈로 분해하여 서술하고 각 모듈은 목적, 기능, 인터페이스, 종속관계 등을 포함한다[9][10]. 각 모듈이 제공하는 기능은 슈도코드(Pseudo Code) 형태로 서술하고 모듈간 상호관계를 명확히 기술하여야 한다. 모듈간 상호관계에는 효과, 예외사항, 오류 메시지 등에 대한 세부사항을 제공하여 모든 인터페이스의 사용목적 및 방법을 포함하여야 한다.

개발과정에서의 시스템 관련 인터페이스는 [그림 6]과 같으며, 상세설계 수준의 모듈간 인터페이스뿐만 아니라 시스템 외부 인터페이스와 서브시스템간 인터페이스까지도 포함하고 있다. 시스템은 이러한 인터페이스를 통하여 상호유기적으로 동작한다.

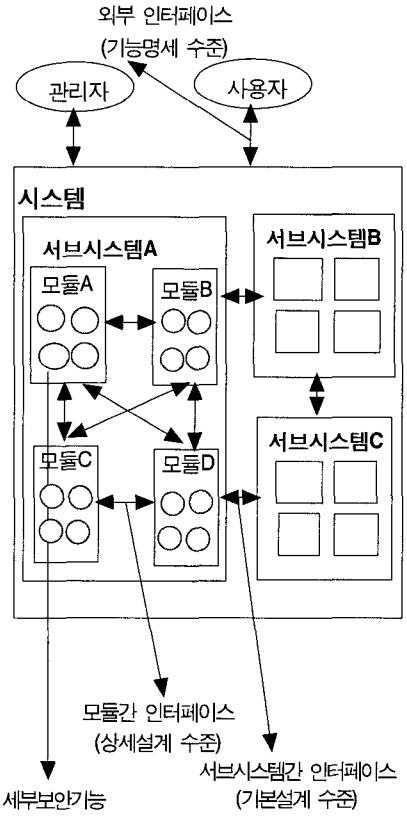
1.4 검증명세 및 원시프로그램

검증명세는 슈도코드 형태의 상세설계 내용에 대한 검증자료로써 소스코드 형태를 포함하고 있으며, 검증명세를 통하여 원시프로그램를 확인할 수 있다. 또한, 상세설계 및 검증명세를 통하여 시스템 동작의 흐름을 파악할 수 있다.

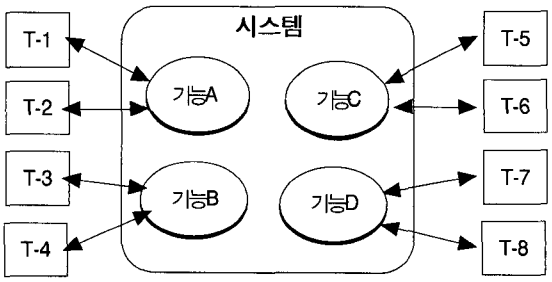
2. 평가등급별 시험 커버리지

2.1 K2 시험 커버리지

정보보호시스템 평가기준의 K2 등급에서는 개발과정의 기능명세와 기본설계를 요구하고 있다. 요구된 기능명세와 기본설계를 통하여 수행할 수 있는 시험은 블랙박스 시험으로 불리는 기능 시험을 할 수 있다. 기능명세로부터 시험되어야 하는 기능들을 분할하고



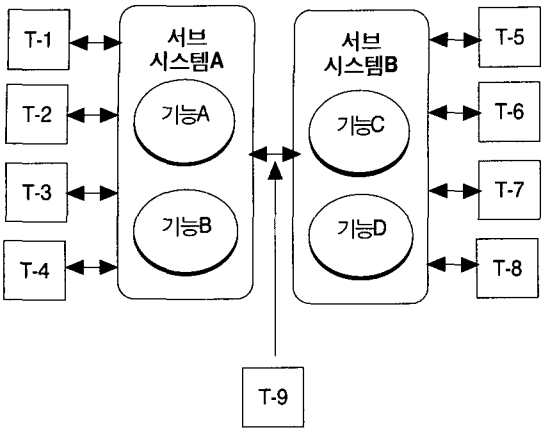
[그림 6] 시스템 관련 인터페이스



[그림 7] 기능명세 수준의 시험 커버리지

적절한 시험 기준을 설정하여 이미 살펴본 기능 시험 기법을 사용하여 시험하게 된다. 생성된 시험케이스들은 기능명세의 모든 외부인터페이스와 보안기능들을 포함하고, 기본설계의 서브시스템간 모든 내부 인터페이스를 포함해야 한다.

[그림 7]과 [그림 8]에서 K2 등급의 시험 커버리지를 만족시키는 시험케이스를 생성하는 방법을 표현하고 있다. [그림 7]에서 볼 수 있듯이, 기능명세에서는 시스템의 기능을 세분화하고 외부인터페이스를 통



〔그림 8〕 기본설계 수준의 시험 커버리지

한 시험케이스를 생성한다. 이를 통하여 시스템의 보안기능 및 외부인터페이스에 대한 시험 커버리지를 만족하게 된다.

〔그림 8〕에서는 기능명세만으로 확인이 불가능한 서브시스템간 인터페이스를 기본설계를 통하여 시험케이스를 생성하였다. 이를 통하여 보안기능 및 외부인터페이스뿐만 아니라 서브시스템간 인터페이스에 대한 시험 커버리지를 만족하게 된다.

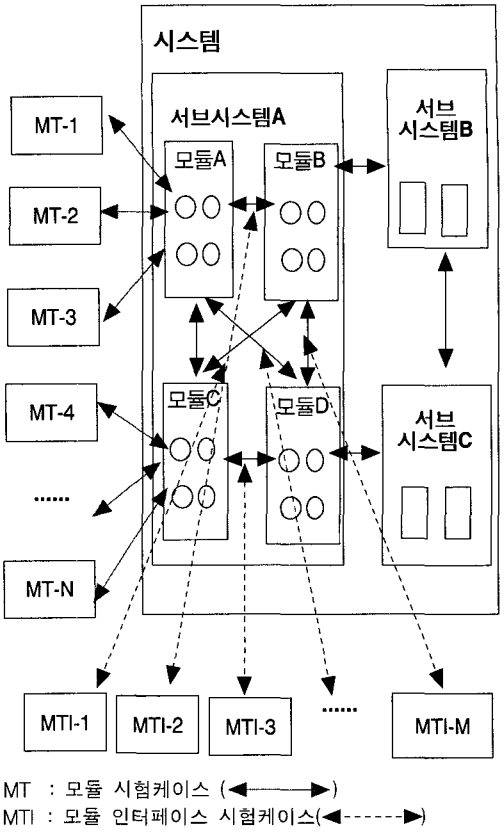
2.2 K3 시험 커버리지

정보보호시스템 평가기준의 K3 등급에서는 개발과정의 기능명세, 기본설계 및 상세설계와 일부 보안기능 및 보안 메커니즘에 대한 검증명세와 원시프로그램을 요구하고 있다. K2 등급의 시험 커버리지와 비교해 볼 때, K3 등급에서는 상세설계서에 기초한 모듈 및 모듈 인터페이스를 추가적으로 요구하고 있다. 또한 일부 보안기능 및 보안메커니즘에 대한 추가 자료를 요청할 수 있기 때문에 시스템 실행 시 실행 흐름을 파악할 수 있다. 따라서, K3 등급에서는 K2 등급에서의 기능 시험뿐만 아니라 모듈을 통한 구조 시험을 추가적으로 요구한다.

모듈에 대한 구조 시험은 이미 살펴본 구조 시험 기법들을 이용하여 시스템 흐름에 기반한 시험케이스를 생성한다. 생성된 시험케이스는 각 모듈의 모든 기능을 확인할 수 있어야 하며 모듈간 인터페이스에 대한 시험도 포함되어야 한다. 따라서, 각 모듈은 여러 개의 시험케이스를 필요로 할 수 있으며 〔그림 9〕에서 볼 수 있다.

2.3 K4 시험 커버리지

정보보호시스템 평가기준의 K4 등급에서는 개발과정의 기능명세, 기본설계, 상세설계, 검증명세 및 원시프로그램을 요구하고 있다. K3 등급의 시험 커버리지와 비교해 볼 때, K3 등급에서 제한적이었던 검증명세와 원시프로그램에 대한 분석을 가능케 함으로써 보다 철저한 구조 시험이 가능하다. 시험케이스 생성 및 시험 방법은 K3 등급에서 알아본 내용과 같다.



〔그림 9〕 상세설계 수준의 시험 커버리지

2.4 K2/K3/K4 시험 커버리지 비교

이미 살펴본 바와 같이 평가등급 K2에서 K4에서는 요구되는 시험 커버리지가 구분되었으며, 이는 개발과정의 제출물 요구사항이 다르기 때문이다.

K2 등급에서는 기능명세와 기본설계를 통한 기능시험이 가능하고, K3 등급에서는 K2 기능 시험에 부가적으로 상세설계에 기술된 슈도코드 형태의 기능을 중심으로 모듈에 대한 구조 시험을 수행한다. K4 등급에서는 K3 등급의 기능 및 구조 시험에 부가적으로 검증명세와 소스코드를 기반으로 철저한 구조 시험을 한다. 각 평가등급별 가능한 시험 및 시험 커버리지는 〔표 2〕와 같다.

〔표 2〕 등급별 시험 커버리지

구분	K2	K3	K4
시험 기법	- 기능 시험	- 기능 시험 - 구조 시험	- 기능 시험 - 구조 시험
시험 커버리지	- 외부인터페이스 - 보안기능	- 외부인터페이스 - 서브시스템간 인터페이스 - 모듈간 인터페이스 - 보안기능	- 외부인터페이스 - 서브시스템간 인터페이스 - 모듈간 인터페이스 - 보안기능

Ⅳ. K4 시험 커버리지 수준의 시험케이스 생성 예

지금까지 일반적인 시험 기법과 정보보호시스템 평가기준에서 요구하는 시험 커버리지를 분석하였다. 본 절에서는 침입차단시스템 K4 등급에서 요구되는 수준의 시험 커버리지를 만족하는 시험케이스 생성 예를 보이고자 한다.

사용된 시험케이스 생성 기법은 침입차단시스템의 일부 기능에 대하여 시스템 명세로부터의 기능 시험은 경계 및 동치분할 커버리지를, 코드로부터의 모듈에 대한 구조 시험은 경로 커버리지를 사용하였다.

1. 기능명세 및 기본설계로부터의 시험케이스 생성

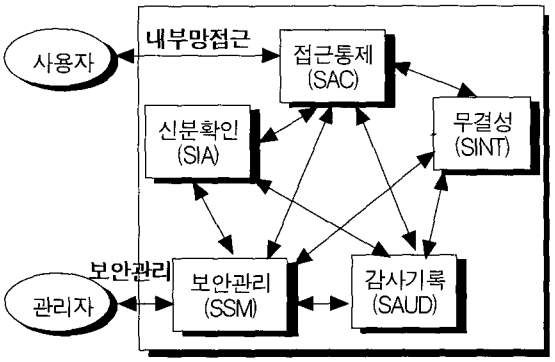
1.1 침입차단시스템 기능명세 및 기본설계

침입차단시스템은 내부망과 외부망 사이에서 정보의 흐름을 안전하게 통제하는 시스템으로 신분확인, 접근통제, 무결성, 감사기록 및 추적, 보안관리 등의 기능을 수행한다. 본 절에서 사용자 추가 기능에 대하여 명세 및 시험케이스를 생성하고자 한다.

◦ 사용자 추가 기능
- 관리자는 사용자 추가를 위하여 아이디와 패스워드를 입력한다
- 아이디는 6~10 자리 숫자 및 문자의 조합
- 패스워드는 6~10 자리이고 숫자, 문자, 특수문자의 조합

◦ 사용자 추가 관련 에러 메시지
- 아이디 길이는 6자에서 10자입니다
- 아이디가 숫자와 문자의 조합이 아닙니다
- 기존 아이디가 존재합니다
- 패스워드 길이는 6자에서 10자입니다
- 패스워드가 숫자, 문자, 특수문자의 조합이 아닙니다.

침입차단시스템의 서브시스템은 〔그림 10〕과 같으며, 서브시스템간 인터페이스와 사용자 및 관리자의 외부 인터페이스를 포함하고 있다.



〔그림 10〕 침입차단시스템 서브시스템

1.2 기능 시험 시험케이스 생성

1.1의 내용을 바탕으로 사용자 추가 기능에 대한 시험케이스를 경계 커버리지 기법으로 생성하면 다음과 같다.

◦ 기능 시험 시험케이스
- 사용자 아이디 및 패스워드를 입력하여 사용자를 추가한다
- 아이디 및 패스워드 길이에 대한 경계 분석

5 6 7 9 10 11

시험 케이스	입력		출력
	아이디	패스워드	
1	a1234	rtfs123$	에러출력
2	vdf345	rtfs123$	사용자추가
3	ddf1234	rtfs123$	사용자추가
4	dfss34569	rtfs123$	사용자추가
5	df345dfds4	rtfs123$	사용자추가
6	lkjklei4953	rtfs123$	에러출력
7	od2f4sfq	ilkk4	에러출력
8	od2f4sfq	ilsfe#	사용자추가
9	od2f4sfq	dlf34&w	사용자추가
10	od2f4sfq	lslkf8@kl	사용자추가
11	od2f4sfq	cn*ie9flks	사용자추가
12	od2f4sfq	djlfs94!34q	에러출력

- 문자, 숫자, 특수문자 조합에 대한 동치분할 분석
  · 아이디 : 문자만, 숫자만, 문자·숫자로 구성
  · 패스워드 : 문자만, 숫자만, 특수문자만, 문자·숫자·특수문자로 구성

시험 케이스	입력		출력
	아이디	패스워드	
1	acddfi	rtfs123$	에러출력
2	345256	rtfs123$	에러출력
3	ddf123	rtfs123$	사용자추가
4	od2f4sfq	bbdfsf	에러출력
5	od2f4sfq	356225	에러출력
6	od2f4sfq	@$^&#$	에러출력
7	od2f4sfq	dfs45%	사용자추가

◦ 외부 인터페이스 및 내부 인터페이스 시험
- 사용자 및 관리자 외부 인터페이스를 통한 기능 시험
- 서브시스템간 인터페이스 시험
  · 신분확인과 보안관리/접근통제/감사기록
  · 접근통제와 보안관리/감사기록/무결성
  · 무결성과 보안관리/감사기록
  · 감사기록과 보안관리

내부 인터페이스 시험에서는 외부 인터페이스를 통하여 내부 인터페이스가 시험될 수 있도록 시험케이스를 작성하여 시험하는 것이 중요하다.

2. 상세설계로부터의 시험케이스 생성

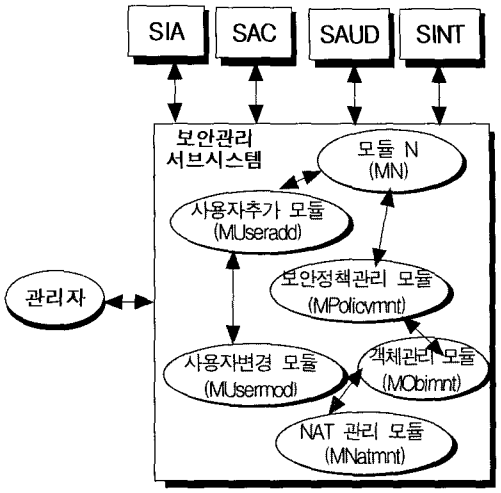
2.1 침입차단시스템 상세설계

상세설계에서는 서브시스템을 여러 모듈로 구성하고 각 모듈 및 모듈간 인터페이스에 대하여 상세하게 기술하여야 한다. 이를 토대로 모듈에 대한 실행 흐름을 분석하여 흐름도를 작성함으로써 구조 시험을 위하여 시험케이스를 작성할 수 있다.

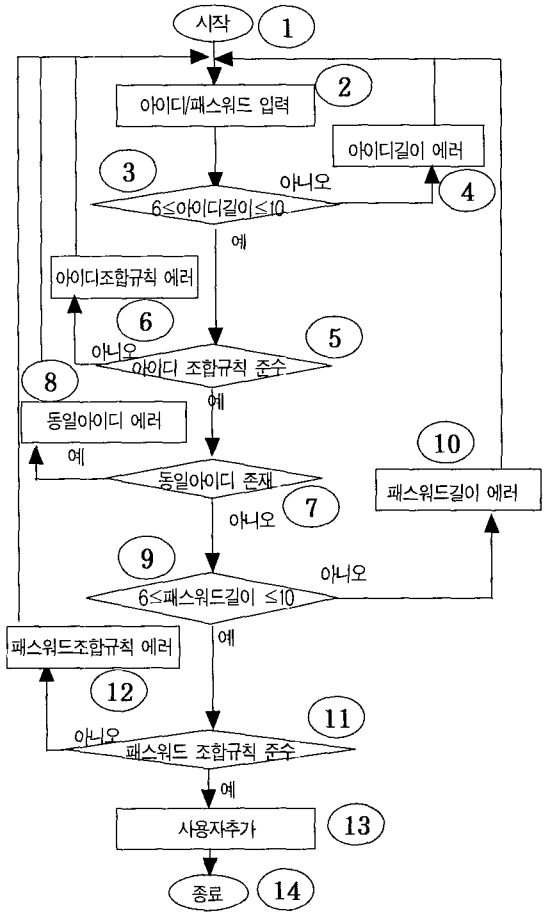
사용자 추가 기능은 보안관리 서브시스템에 속하며, 보안관리 서브시스템을 여러 모듈로 구성하면 〔그림 11〕과 같고, 이 기능의 작성된 흐름도는 〔그림 12〕와 같다.

2.2 구조 시험 시험케이스 생성

2.1의 내용을 바탕으로 사용자 추가 모듈에 대한



〔그림 11〕 보안관리 서브시스템의 모듈 구성도



〔그림 12〕 사용자 추가 기능 흐름도

구조 시험의 시험케이스를 경로 커버리지 기법으로 생성하면 아래와 같다.

◦구조 시험 시험케이스

구조 시험을 위하여 [그림 12]의 흐름도로부터 경로 커버리지에 대한 회귀성 복잡도를 계산하면 6을 얻을 수 있으며, 모든 경로를 포함하는 시험케이스를 생성하면 아래와 같다.

- 경로1 : 1,2,3,5,7,9,11,13,14
- 경로2 : 1,2,3,4,2,3,5,7,9,11,13,14
- 경로3 : 1,2,3,5,6,2,3,5,7,9,11,13,14
- 경로4 : 1,2,3,5,7,8,2,3,5,7,9,11,13,14
- 경로5 : 1,2,3,5,7,9,10,2,3,5,7,9,11,13,14
- 경로6 : 1,2,3,5,7,9,11,12,2,3,5,7,9,11,13,14

시험 케이스	입력		출력
	아이디	패스워드	
경로1	ddf123	rtfs123$	사용자추가
경로2	34525	rtfs123$	아이디길이 에러출력
경로3	34525	rtfs123$	아이디조합규칙 에러출력
경로4	ddf123	rtfs123$	동일아이디 에러출력
경로5	od2f4sfq	35622	패스워드길이 에러출력
경로6	od2f4sfq	dfs45dfs	패스워드조합규칙 에러출력

위에서 보는 바와 같이, 한 모듈에서도 구조 시험의 경우, 6개의 시험케이스가 생성되었다.

◦모듈간 인터페이스 시험케이스

상세설계의 모듈간 인터페이스 시험은 [그림 11]로부터 얻을 수 있다.

- 사용자추가 모듈과 사용자 변경모듈/모듈 N
- 보안정책관리 모듈과 모듈N/객체관리 모듈
- NAT 관리 모듈과 객체관리 모듈

V. 결 론

시험은 시스템 내의 결함을 발견하는 모든 행위를 말하며, 시험케이스 생성을 위해서는 정보보호시스템을 이해하기 위해 개발과정의 어떤 산출물을 제출되었느냐에 따라 시험 커버리지 및 시험케이스 생성 방법이 결정되었다. 국내의 정보보호시스템 평가·인증 제도에서는 평가등급별로 제출물이 다르기 때문에 적용될 수 있는 시험 기법 등이 선택적으로 적용될 수 있다.

본 연구에서는 침입차단시스템의 K4 등급의 시험을 위하여 동치분할 커버리지, 경로 커버리지, 경계 커버리지 기법을 이용하여 기능 시험 및 구조 시험을 위한 시험케이스 생성 예를 제시하였다. 이를 통하여 시스템의 보증성 및 신뢰성을 향상시킬 수 있을 것이며, 사용자들이 신뢰된 인증제품을 안전하게 사용할 수 있을 것이다.

참고문헌

[1] W. E. Perry, "Effective Methods for Software Testing, 2nd Ed.", John Wiley & Sons, Inc., 1999

[2] G. Myers, "The Art of Software Testing", John Wiley & Sons, Inc., 1979

[3] C. Kaner, "Quality Cost Analysis: Benefits and Risks", http://www.kaner.com/qualcost.htm

[4] K. J. Ross, "Practical Guide to Software System Testing", K. J. Ross & Associates Pty. Ltd., 1998

[5] C. Kaner, J. Falk, Q. Nguyen, "Testing Computer Software, 2nd Ed.", Thomson Computer Press, 1993

[6] British Computer Society Specialist Interest Group in Software Testing(BCS SIGIST), "Standard for Software Component Testing, Working Draft 3.4", April 2001

[7] B. Beizer, "Black-Box Testing", John Wiley & Sons, 1995

[8] 정보통신부, "정보통신망 침입차단시스템 평가기준", 정보통신부, 2000.2

[9] 정보통신부, "정보보호시스템 공통평가기준", 정보통신부, 2002.8

[10] 한국정보보호진흥원, "개발자를 위한 침입차단·탐지시스템 평가제출물 작성가이드", 한국정보보호진흥원, 2003.10

〈著 者 紹 介〉

윤 여 웅 (Yeo-Wung Yun)

1996년 2월 : 한남대학교 컴퓨터공학과 공학사
1998년 2월 : 한남대학교 컴퓨터공학과 공학석사
2000년 3월~현재 : 충북대학교 전자계산학과 박사과정
2000년 10월~현재 : 한국정보보호진흥원 연구원
〈관심분야〉 정보보호시스템 평가, 네트워크 및 시스템 보안

안 재 영 (Jae-Young Ahn)

1998년 2월 : 중앙대학교 전자공학과 공학사
2002년 2월 : 중앙대학교 전자공학과 공학석사
2001년 7월~현재 : 한국정보보호진흥원 연구원
〈관심분야〉 정보보호시스템 평가, 네트워크 및 시스템 보안

서 동 수 (Dong-Su Seo)

1986년 2월 : 중앙대학교 컴퓨터공학과 이학사
1990년 9월 : 영국 맨체스터 이공대학 이학석사
1994년 6월 : 영국 맨체스터 이공대학 공학박사
1994년 6월~1998년 2월 : 한국전자통신연구원 선임연구원
1998년 3월~현재 : 성신여자대학교 컴퓨터정보학부 조교수
〈관심분야〉 소프트웨어 재사용, 객체지향 프로그래밍 기법, 정형 기법, 정보보호 평가인증