정보보호제품 평가 위한 취약점 분류체계 및 적용방안

방기석*, 김일곤**, 이지연***, 이준석***, 최진영****

요 약

취약점 분류체계는 공격자의 위협 행동 패턴을 이해하고 보호 대상 자산에 대한 피해를 야기시키는 취약점 유형 및 경향을 분석하여, 보안사고를 미연에 방지 하는데 중요한 역할을 담당한다. 이런 이유로, 정보보호제품을 평가하기 위한 국제표준인 공통평가기준에서는 취약점 정보를 수집하고 침투시험을 수행하는 과정을 요구하고 있다. 하지만, 방대한 취약점 정보로부터 해당 정보보호제품의 잠재적인 취약점을 수집 및 필터링하기 위해서는 개발자 또는 평가자가 직관적으로 이해하고 적용할 수 있는 취약점 분류 체계 및 적용방안이 마련되어야 한다. 본 논문에서는 정보보호제품 구성요소, 보안구조, 보안기능 및 소스코드 관점에서 보안취약점을 분류하고 적용하는 방안을 제시한다. 이를 위해, 공통평가기준에 적용할 수 있는 보안 취약점 분류체계 동향을 조사하고 차이점을 분석한다. 또한, 개발자 또는 평가자가 정보보호제품 개발 또는 평가에 실용적으로 적용할 수 있는 새로운 취약점 분류 체계 및 적용방안을 제시하고자 한다.

Classification Criteria and Application Methodology for Evaluating IT Security Products

Ki-Seok Bang*, Il-Gon Kim**, Ji-Yeon Lee***, Jun-Seok Lee***, Jin-Young Choi****

ABSTRACT

CC(Common Criteria) requires to collect vulnerability information and vulnerability analysis by using penetration testing for evaluating IT security products. However, CC has been criticized from developers or QA managers due to its complexity of terms, abstract description of evaluation methods and non-existence of guidelines. In this paper, we propose a guideline of vulnerability assessment for developers and evaluators by analyzing and summarizing of its requirements and processes defined in CC. To do this, we classify the evaluation process of AVA assurance family into 4 parts and describe each evaluation working systematically unit under every steps.

Key Words : Common Criteria, Vulnerability Classification, AVA

* 한림대학교 기초교육대학(✉mysaver@hallym.ac.kr)
** 한국인터넷진흥원
*** 동남보건대학 경영학과
**** 고려대학교 컴퓨터전파통신공학부
· 제1저자(First Author) : 방기석 · 교신저자(Correspondent Author) : 이지연
· 접수일(2011년 9월 1일), 수정일(1차 : 2011년 10월 4일), 게재확정일(2011년 10월 7일)

Ⅰ. 서 론

최근 무선 네트워크 개방화 및 클라우드 컴퓨팅의 활성화와 더불어 공격자에 의한 침투경로는 다양해지고 또한 스턱스넷[1] 등 새로운 공격 및 보안 취약점이 점차 발견됨에 따라 보안사고의 위험성은 점차 증가되고 있다. 또한 IT 제품에서 소프트웨어가 차지하는 중요성 및 비중이 증가하여, SW 보안오류에 의한 취약점 발견 및 신속한 보안 대응방안이 더욱 더 요구되고 있다.

SW 보안 취약점에 대응 방안을 마련하기 위해서는 보안 취약점을 정확히 인식하고 분류하는 기준이 마련되어야 한다. 취약점 분류체계는 공격자의 위협 행동 패턴을 이해하고 보호 대상 자산에 대한 피해를 야기시키는 취약점 유형 및 경향을 분석하여, 보안 사고를 미연에 방지 하는데 중요한 역할을 담당한다. 하지만, 일반적인 보안 취약점 기준은 SW 취약점을 기준으로 작성되었기 때문에 매우 추상적이고 비정형화된 특성을 갖고 있어 그 유형과 항목을 일관되게 분류하기 어려운 한계점을 갖고 있다.

최근 클라우드 컴퓨팅, 가상화 등 IT 기술의 발전과 더불어 개인정보보호법 등 제도적 뒷받침에 따라 다양한 유형의 정보보호제품군이 등장하고 있다. 이에 따라, DDoS 대응장비, 웹방화벽, IPS 등 각종 정보보호제품이 기업의 자산을 보호하기 위해 큰 역할을 수행하고 있으며. 국내에서는 국가기관에 조달되는 정보보호제품에 대한 일정 수준의 보안성을 보증하기 위해 공통 평가기준(Common Criteria)[2][3] 인증을 의무적으로 취득 하도록 하는 평가제도를 운영하고 있다.

CC는 ISO 15408 표준[4]으로, 1단계부터 7단계 까지의 보증등급을 제공하고 있으며, '등급'은 EAL(Evaluation Assurance Level)로 불리어진다. 공통평가기준에서는 정보보호제품의 SW 개발 생명주기에 따른 각종 산출물을 평가하고 보증하기 위한 활동으로 구성되어 있다. 그 중에서도 AVA 보증 클래스에서는 취약성 평가활동을 통해 잠재적인 보안 취약점을 식별하고 시험하는 과정을 요구하고 있다. 특히, 잠재적인 취약점 정보를 수집하고 침투시험을 수행하는 과정을 통해 취약점 평가를 수행하도록 명시하고 있다. 하지만, 방대한 취약점 정보로부터 해당 정보보호제품의 잠재적인 취약점을 수집 및 필터링하기 위해서는 개발자 또는 평가자가 직관적으로 이해하고 적용할 수 있는 취약점 분류 체계 및 적용방안이 마련되어야 한다. 또한, 국내 보안제품 시장의 특성상 빠르게 변화하는 개발 생명주기 및 제한된 평가기간을 고려한 실용적인 취약점 분류체계 및 적용 방안이 필요하다. 이에 따라, 본 논문에서는 공통평가기준에 적용할 수 있는 보안 취약점 분류체계 동향을 조사하고 차이점을 분석한다. 또한, 개발자 또는 평가자가 정보보호제품 개발 또는 평가에 실용적으로 적용할 수 있는 새로운 취약점 분류 체계 및 적용방안을 제시하고자 한다.

본 논문의 구성은 다음과 같다. 제Ⅱ장에서는 관련 연구를 소개한다. 제Ⅲ장에서는 CWE 및 CAPEC 취약점 분류체계를 비교한다. Ⅳ장에서는 본 논문에서 제안한 취약점 분류 체계 및 적용사례를 서술한다. 마지막으로 제Ⅴ장에서는 결론 및 향후 연구방향을 제시한다.

Ⅱ. 관련연구

소프트웨어 보안에 대한 중요성이 점차 증가함에 따라, 취약점 정보를 효율적으로 공유하고 전파하여, 보안 위협에 능동적으로 대응하기 위한 노력으로 취약점 정보를 분류하기 위한 필요성이 대두되었다. 이러한 필요성에 따라, CVE[5], CWE[6], CAPEC[7] 등 취약점 정보를 유일하게 식별하고 체계적으로 분류하고 지속적으로 관리하기 위한 다양한 연구가 진행되었다[8][9][10].

CVE는 서로 유사한 취약점 정보가 개별 기관에 따라 상이하게 분류되는 혼선을 방지하기 위해 공통된 유일한 명칭으로 취약점을 식별하기 위해 제안되었다. CWE는 보호대상 자산 관점에서 설계, 구조 및 코드

등에서 다양한 취약점을 분류하고 있다. 이에 반해, CAPEC은 공격행위 패턴 관점에서 취약점 정보를 분류한다. 공통평가기준은 취약점 평가활동을 수행하기 위한 선행과정으로 취약점 정보를 식별하는 과정을 요구하고 있으며, 이를 위해 CWE와 CAPEC을 적용할 수 있다. 이에 따라, 본 논문에서는 공통평가기준 기반 정보보호제품 취약점 평가에 적용할 수 있는 CWE와 CAPEC을 위주로 취약점 분류체계에 대한 동향 및 차이점을 우선적으로 분석하였다.

III. 취약성 평가

3.1 CWE

CWE(Common Weakness Enumeration)는 미국 MITRE 기관에서 일반적으로 소프트웨어에서 공통적으로 발생하는 잠재적인 취약점(weakness)을 체계적으로 분류한 목록을 제공하기 위해 개발되었다[6]. CWE에서 소프트웨어 취약점을 분류하는 핵심 사상은 안전하게 보호하고자 하는 공격 대상인 자산(Asset)의 보안 취약점에 초점을 두어 작성되었다. 또한, 소프트웨어 취약점은 SDLC 과정에서 발생할 수 있는 있기 때문에, 설계, 아키텍처, 코드 단계 등에 대한 취약점 목록도 포함하고 있다. 2011년 3월 22일 기준으로 버전 1.11이 사용되고 있으며, [표 1]과 같이 '뷰(View), 카테고리(Category), 취약점(Vulnerability)' 등으로 구성되어 있다.

'뷰(View)' 에 해당하는 대표적인 예를 살펴보면, 국제 웹 애플리케이션 보안 연구단체에서 개발한 2007년 OWASP Top 10[11] 취약점(CWE-629), NVD[12]에서 사용된 취약점(CWE-635), CERT C 시큐어 코딩 표준에서 정의한 취약점(CWE-734) 등이 속하다. 이처럼 CWE는 CVE 보다 추상화되고 광범위한 수준의 취약점 분류체계를 제공하고 있다.

(표 1) 에서 보는 바와 같이, CWE 분류체계는 취약점 명, 취약점 설명과 더불어 취약점이 어떤 SDLC 단계에서 발생할 수 있는지 설명하기 위한 '도입 시점', 취약점 정보의 상관관계를 표현하는 '연관관계' 및 '관련 공격 패턴' 정보 등으로 구성되어 있다.

표 1. CWE 분류체계 예제
Table 1. Example of CWE Classification

<table>
<tr><th>구분</th><th>내용</th></tr>
<tr><td>취약점 명</td><td>CWE-312: 민감한 정보 평문저장 (Cleartext Storage of Sensitive Informatin)</td></tr>
<tr><td>상태</td><td>초안</td></tr>
<tr><td>취약점 ID</td><td>312</td></tr>
<tr><td>설명</td><td>[개요]
정보가 암호화되어야 하거나 보호되어야 할 때, 애플리케이션이 다른 제어 영역에서 접근 가능한 자원 내에 평문으로 민감한 정보를 저장한다
[부가 설명]
정보가 평문으로 저장되기 때문에, 공격자가 잠재적으로 정보를 읽을 수 있다</td></tr>
<tr><td>도입 시점</td><td>아키텍처 및 디자인</td></tr>
<tr><td>연관관계</td><td><table><tr><th>특징</th><th>ID</th><th>이름</th><th>뷰</th></tr><tr><td>Child Of</td><td>311</td><td>중요 데이터 암호화 실패</td><td>699
1000</td></tr></table></td></tr>
<tr><td>관련 공격패턴</td><td>CAPEC-ID : 37</td></tr>
</table>

예를 들어, 'CWE-312 : 민간함 정보 평문저장' 취약점은 패스워드와 같은 중요 정보가 평문으로 저장되는 취약점을 의미하며, 이는 설계 단계인 '아키텍처 및 디자인' 단계에서 사전 고려되고 제거되어야할 취약점으로 명시되어 있다. 또한, 취약점 ID 311의 하위 분류체계(ChildOf)에 속하며, CAPEC 에서의 유사한 취약점은 ID 317 임을 보여주고 있다.

3.2 CAPEC

CAPEC(Common Attack Pattern Enumeration and Classification)은 공격자 관점에서 공격패턴유형

에 대한 목록을 제시하여, 보안 취약점을 분류하는 방법을 제공한다[7]. CWE가 공격의 대상이 되는 자산 관점에서의 취약점을 유형별로 구분한 것과 달리, CAPEC은 보호대상이 되는 자산을 취약하게 야기하는 취약점을 분류하는데 초점을 맞추고 있는 것이다.

CAPEC이 공격자의 공격 행위를 반영한 취약점의 능동적 측면을 강조했다면, CWE는 공격 당하는 보호자산 취약점의 수동적 측면을 부각시켰다고 볼 수 있다. CAPEC-2000 목록은 버전 1.0에서 시작해서, 2011년 3월 현재 버전 1.6까지 목록이 있으며, 버전 1.6 목록에는 6개의 뷰, 68개의 카테고리, 386개의 공격 패턴을 보유하고 있다.

표 2. CAPEC 분류체계 예제
Table 2. Example of CAPEC Classification

구분	내용
취약점 명	인증 우회(Authentication Bypass)
공격 패턴 ID	210
심각도	중간(Medium)
상태	초안
설명	공격자가 인증 메커니즘을 우회하여 인가된 관리자의 권한으로 응용 프로그램, 서비스 또는 디바이스에 접근한다
공격 전제조건	패스워드, 인증서 등의 방법으로 인증 매커니즘을 구현한다
요구 자원	웹 브라우저 등과 같은 응용프로그램 또는 스크립트 언어
관련 취약점	CWE-ID : 592
관계	특징 / ID / 이름 / 뷰 : ChildOf / 225 / 인증 익스플로잇테이션 / 1000

(표 2)에서 보는 바와 같이, CAPEC 분류체계는 CWE와 유사한 필드 정보와 더불어 '심각도', '공격 전제조건', '요구 자원' 등 보다 많은 부가 정보를 포함하고 있다. 예를 들어, 'CAPEC 210 : 인증 우회'' 취약점은 공격자가 패스워드 또는 인증서 기반 등의 기술을 이용한 인증 메커니즘을 우회할 수 있는 취약점을 명시하고 있다. 인증 우회 취약점을 이용하기 위해서는 웹 브라우저 또는 스크립트 언어 등의 자원이 요구되며, 관련 취약점은 CWE-592와 유사하며, 'Authentication Exploitation' 취약점 명 하위분류(ChildOf)에 속함을 알 수 있다.

IV. 제안된 취약점 분류체계 및 적용방안

4.1 분류체계

공통평가기준에서는 AVA 보증 패밀리에서 보안 등급에 따라, 보안 취약점 수집 및 침투시험 항목 도출에 대한 프로세스 및 요구사항을 정의하고 있다. 정보보호제품에 대한 잠재적인 취약성 항목을 도출하고 침투시험 과정을 통해, 실제로 보안 위협에 따른 공격 가능성을 확인하기 위해서는 우선 보안 취약점 항목을 수집하고 선정하는 과정이 우선시 되어야 한다. 하지만, 앞에서 언급하였듯이 한정된 방대한 취약점 정보수집 채널 및 제한된 평가기간 안에, 정보보호제품에 적합한 취약점을 선정하기 위한 논리적인 취약점 분류방법이 요구된다. 이에 본 논문에서는 정보보호제품의 공통된 구성요소, 보안기능, 보안구조 및 소스코드의 4가지 관점에서 보안 취약점을 직관적이고 체계적으로 분류하기 위한 방안을 제시한다..

1) 구성요소

보안목표명세서(ST)를 살펴보면, 정보보호제품의 운영환경과 더불어 논리적 구성요소를 파악할 수 있다. (그림 1)에서 살펴보듯이 정보보호제품을 일반적으로 공통구성요소 관점에서 분석하면, 시스템(하드웨어, 운영체제, 애플리케이션), 네트워크, 웹, DB 등의 세부 구성항목으로 분류할 수 있다. 일반적인 취약점 분류체계에서는 소프트웨어 보안 취약점을 대상으로 제안되었기 때문에, 매우 추상적이고 광범위한 범위에서 취약점 유형을 분류하였다. 하지만, 정보보호제품의 경우에는 평가범위 및 대상 서브시스템 혹은 모듈이

존재하기 때문에, 공통적인 구성요소를 기반으로 취약점 정보를 수집하는 보다 실용적인 과정이 요구되어진다고 판단하였다.

2) 보안구조

공통평가기준 버전 2.3과 버전 3.1의 큰 차이점 중 한 가지는 보안구조(Security Architecture) 평가 활동이 새롭게 추가되었다는 점이다[13]. 보안 취약점은 크게 있도록 구현 되어진다. 특히 외부보안기능 인터페이스(TSFI)는 공격자가 내부 보안모듈에 접근하여 보안기능을 우회할 수 있는 논리적 채널 역할을 제공할 수 있기 때문에, 안전한 보안 설계 및 구현과정이 요구되어진다. 예를 들어, 사용자가 취약한 웹 프로그램을 이용하여 구현된 '사용자 식별 및 인증' 보안기능에 SQL Injection 공격 등을 이용하여 보안기능을 우회하여 관리자 권한을 획득할 수 있다.

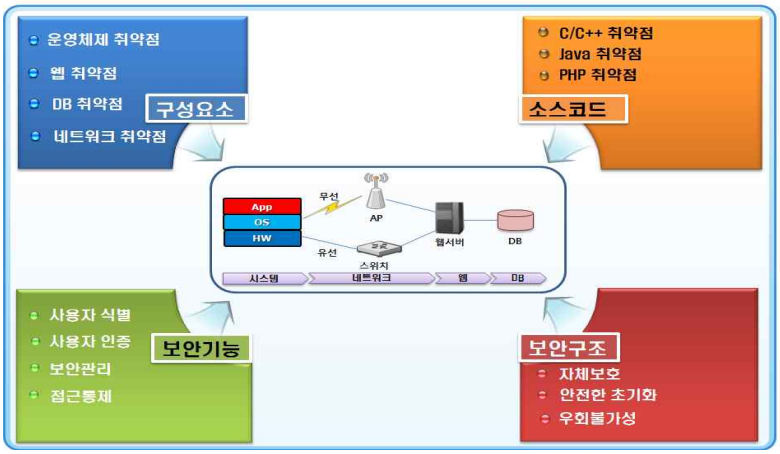

그림 1. 정보보호제품 취약점 분류기준

Fig. 1. Vulnerability Classification Criteria for Information Security Products

설계 및 구현 관점에서 발견될 수 있으며, 보안구조는 설계단계에서 발생할 수 있는 보안 취약점을 미연에 방지하는데 목적이 있다. 4가지 보안특성(영역분리(Domain Separation), 자체보호(Self Protection), 안전한 초기화(Secure Initialization), 우회공격(Bypassing))을 중심으로 보안구조의 보안성 평가할 수 있다. 예를 들어, 리눅스 및 MS 윈도우 등의 운영체제에서는 공격자에 의한 최상위 관리자 메모리 영역 침해를 방지하기 위해 보안구조 설계가 중요한 핵심 기술로 요구된다.

3) 보안기능

정보보호제품은 보안기능의 구현을 통해 기밀성, 무결성, 가용성 등과 같은 보안요구사항을 만족할 수

4) 소스코드

공통된 구성요소, 보안구조, 보안기능 등은 결국 프로그래밍 언어를 이용하여 구현되며, 안전하게 설계되었다고 하더라도 결국 개발자의 부주의한 코딩은 보안취약점을 야기시키게 된다. 이에 따라, 최근에는 C, Java 언어 등에 대한 시큐어 프로그램에 대한 요구가 보다 강화되어 지고 있다.

4.2. 적용사례

본 절에서는 4.1 절에서 제안한 보안 취약점 분류체계를 통해 정보보호제품에 대한 취약점 정보를 수집하고 분류하는 사례를 설명하고자 한다. 정보보호제

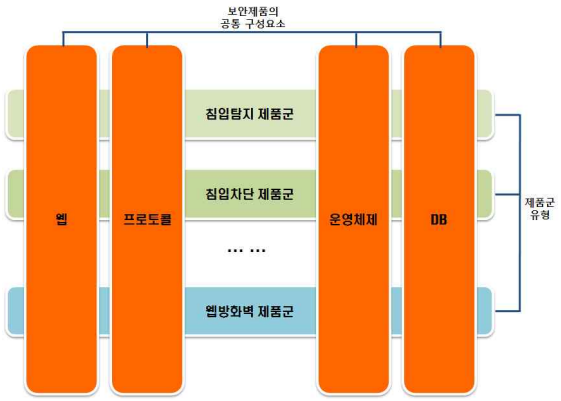

그림 2. 정보보호제품군 및 공통요소간의 상호관계

Fig. 2. Mutual Relationship between Information Security Products' types and Common Parts

(그림 2)에서 보는 바와 같이, 정보보호제품군의 유형은 새로운 기술이 등장할 때마다 시장의 요구사항에 따라 변하게 된다. 초기 보안시장에는 침입탐지, 침입차단 제품군이 대세를 이루다가 최근에는 스마트 전자정부, 클라우드 컴퓨팅, DDoS 공격 등의 이슈와 더불어 자료유출방지, 좀비 PC 차단제품 등이 새롭게 등장되고 있다. 하지만, 근본적으로 이러한 제품군들을 논리적 구성요소 관점에서 바라보면 웹, 프로토콜, 운영체제, DB 등의 공통적인 집합을 발견할 수 있다. 따라서, 빠르게 변화하는 정보보호제품 유형에 효율적으로 대처하기 위해서라도 공통적인 구성요소 관점에서 취약점 정보를 수집하고 보안 대응방안을 마련할 필요가 있다.

(표 3)에서는 가상화 제품으로 널리 알려진 개발 클래스의 패밀리 간 관계 및 다른 클래스와의 관계 Citrix XenServer[16]와 Citrix MetaFrame Presenation Server[17] 가상화 제품을 대상으로 보안 취약점 정보를 분류하고 수집한 사례를 보여주고 있다. 사례에서 보는 바와 같이, 공통된 보안구성 요소, 보안기능 및 보안구조에 대한 취약점 정보는 공개된 취약점 데이터베이스를 통해 수집이 가능하다. 하지만, 소스코드 취약점의 경우는 개발업체의 중요 자산이기 때문에 공개된 취약점 정보를 수집하기 어렵다. 이런 이유로, (표 3) 예제에서도 분류 항목에서 제외되어 있다.

품의 취약점을 수집하기 위해서는 우선 평가하고자 하는 대상 제품에 관련된 산출물(보안목표명세서, 상세 설계서 등)들을 조사하여, 물리적/논리적 구성요소, 보안구조, 보안기능, 소스코드 등을 분석해야 한다. 즉, 정보보호제품의 취약점 정보를 수집 및 필터링 하기 위해서, 공통 구성요소, 보안구조, 보안기능 및 소스코드 관점에서 잠재적인 취약점을 조사하고 분류하도록 한다. 일반적으로 공개된 취약점 데이터베이스 [12][14][15]를 이용하여, 취약점 정보를 조회할 수 있다.

표 3. 취약점 정보 조회 및 분류 사례

Table 3. Example of Search and Classification of Vulnerability Information

분류	구분	취약점	취약점 설명
구성요소	웹	삽입 (Injection)	XenServer Resource Kit 의 login.php 코드에 대한 SQL 삽입공격 취약점(CVE-2009-3758)
보안기능	인증	인증우회	Citrix XenServer Authentication 인증 우회 취약점
보안구조	우회	정책우회	Citrix MetaFrame Presentation Server 보안정책 위회 취약점

이런 경우는 개발자 측면에서는 코딩규칙(coding convention)에 따른 소스코드 개발 절차가 요구되어지며, 평가자 측면에서는 소스코드정적분석 도구 등을 통한 프로그래밍 언어상의 취약점을 검사해야 한다. 본 논문에서 제안한 취약점 분류체계를 적용하여 선별된 취약점 정보를 분석하면 결국은 CWE 및 CAPEC과 연관성을 갖게 됨을 확인할 수 있다. 예를 들어, [표 3]의 웹 취약점은 CWE-713 항목과 매핑되며, 인증 취약점은 CAPEC-210 항목과 연관되어 진다. 즉, 본 논문에서 제안한 취약점 분류체계는 SW 보안 취약점 관점이 아니라 정보보호제품 평가를 위한 관점에서 CWE 및 CAPEC 분류체계를 평가에 실용적으로 활용하기 위한 적용방안인 것이다.

Ⅴ. 결론

정보보호제품의 보안 신뢰성을 증대시키기 위해서는 개발 및 운영단계에서의 취약성 평가 과정이 절대적으로 요구된다. 하지만, 정보보호제품의 취약점 정보는 매우 광범위할 뿐만 아니라 한정된 기간에 잠재적인 취약점을 조사하고 적용하는데 많은 시간과 노력이 요구된다. 또한, 기존에 제안된 취약점 분류기준은 무형의 특징을 갖는 SW 취약점을 대상으로 작성되었기 때문에 매우 추상적이고 영역별 경계구분이 모호한 제한점을 갖고 있다.

이에 본 논문에서는 수많은 취약점 분류기준 중에서도 공통평가기준을 이용하여 정보보호제품 평가에 적용할 수 있는 CWE 및 CAPEC의 동향을 조사하고 차이점을 분석하였다. 또한, CWE 및 CAPEC 분류기준을 활용하여, 정보보호제품 평가에 실용적으로 적용할 수 있는 취약점 분류 체계 및 적용방안을 제시하였다. 마지막으로 Citrix 가상화 제품에 대한 예제를 통해 본 논문에서 제안한 취약점 분류체계 및 적용방안의 타당성을 간략히 증명하였다.

향후 연구방향으로는 각종 취약점 분류체계에 사용되는 정보구성 항목[18]을 분석하여, 정보보호제품 평가에 필수적으로 필요한 취약점 항목용 정보구성 체계를 작성하는 연구를 진행하고자 한다.

참고문헌

[1] [알아봅시다] 스턱스넷 : 국가기간망 공격, "http://www.dt.co.kr/contents.html?article_no=2010100702011860739002"

[2] "Common Criteria for Information Technology Security Evaluation", Ver 3.1, CCMB-2006-09-03, September 2006

[3] Committee on Government Reform House of Representatives, "Exploring Common Criteria: Can it assure that the federal government gets needed security in software?", Serial No. 108-126, September 2003

[4] ISO/IEC International Standard(IS) 15408, Parts 1,2,3, Aug. 1999

[5] CVE, "http://cve.mitre.org"

[6] CWE, "http://cwe.mitre.org"

[7] CAPEC, "http://capec.mitre.org"

[8] K. Tspipenyuk, B. Chess, G. McGraw, "Seven Pernicious Kingdom:s: A Taxonomy of Software Security Errors", Proc. of the NIST Workshop of Software Security Assurance Tools, Techniques, and Metrics(SSATTM). Nov. 2005.

[9] 이진영 외 6명, "CWE와 7PK 취약점 분류 비교", 정보과학회 학술발표 논문집, Vol 36, No, 2(D), 2009

[10] 김동진, 조성제, "국가 DB 기반의 국내외 보안 취약점 관리체계 분석", Internet and Information Security, 제1권 제2호, pp.130-147, 2010. 11

[11] OWASP TOP 10,- 2010 : The ten most critical web application seurity risks, http://oval.mitre.org/oval/documents/docs-06/an-introductoin_to_the_oval_language.pdf, August 30, 2010

[12] NVD, "http://nvd.nist.gov"

[13] 조혜숙 외 6명, 공통평가기준 v2.3과 v3.1 비교 분석, 정보과학회지, 제17권, 제6호, 2007. 12.

[14] SecurityFocus, "http://www.securityfocus.com"

[15] Exploit-db, "www.exploit-db.com"

[16] Citrix XenServer, "http://www.citrix.com/xenserver"

[17] Citrix MetaFrame Presentation Server 3.0, "http://support.citrix.com/article/CTX106319"

[18] 김동진 외 3명, "정보신기술 보안취약점 활용을 위한 효율적인 취약점 관리체계", 정보과학회 학술발표논문집 Vol.37, No.2(B). 2010.

감사의 글

이 논문은 2011년도 동남보건대학 연구비 지원에 의하여 수행된 것임

저자소개

방기석 (Bang-Ki Seok)

2000년 고려대학교 컴퓨터학과 석사

2005년 고려대학교 컴퓨터학과 박사

2004 ~ 현재: 한림대학교 기초교육대학 조교수

※ 관심분야: 정형기법, 모델체킹, 시큐어 코딩, 소프트웨어 공학

김일곤 (IL-Gon Kim)

2002년 고려대학교 컴퓨터학과 석사
2005년 고려대학교 컴퓨터학과 박사
2006년 프랑스INRIA 연구소 포닥연구원
2007년~ 현재: 한국인터넷진흥원 책임연구원
※ 관심분야: 소프트웨어공학, 정형기법, 정보보호제품 평가

이지연 (Ji-Yeon Lee)

1999년 동덕여자대학교 전자계산학과
2001년 고려대학교 컴퓨터학과 석사
현재 고려대학교 컴퓨터학과 박사수료
2002년~ 현재: 동남보건대학 경영학과 조교수
※ 관심분야 : 소프트웨어공학, 정형기법, 네트워크 보안

이준석 (Jun-Seok Lee)

1991년 청주대학교 전자계산학과 석사
1995년 한남대학교 전자계산공학과 석사
2001년 성균관대학교 통계학과 박사
1994년~ 현재: 동남보건대학 경영학과 부교수
※ 관심분야 : 소프트웨어공학, 네트워크 보안

최진영 (Jin-Young Choi)

1982: 서울대학교 컴퓨터 공학과 학사
1986: Dept. of Mathematics and Computer Science, Drexel Univ. 석사
1993: Dept. of Computer and Information Science, University of Pennsylvania 박사
1996년~ 현재: 고려대학교 컴퓨터·전파통신공학부 교수
※ 관심분야 : 정형기법, 임베디드 실시간 시스템, 프로그래밍 언어, 프로세스 대수, 소프트웨어 공학