CC/CEM에서 유도한 오픈소스 내포형 정보보호시스템의 평가지침

강연희° 김정대 최성자 이강수* 윤여웅 이병권**
(*) 한남대학교 컴퓨터공학과, (**) 한국정보보호진흥원
{dusi82°, jdcom}@se.hannam.ac.kr, irecomm@dreamwiz.com, gslee@eve.hannam.ac.kr, {ywyun, byungkle}@kisa.or.kr

Evaluation Guide of OSS-embedded Information Security System derived from CC/CEM

Yeonhee Kang° Jungdae Kim Sungja Choi Gangsoo Lee* Yeowung Yun Byungkwon Lee**
(*) Dept. of Computer Engineering. Hannam University, (**) Korea Information Security Agency

요 약

오늘날 조직에서 공개된 소프트웨어를 이용한 오픈소스 내포형 정보보호시스템(OSS-embedded Information Security System) 개발이 증가되고 있으며 소스의 상당부분을 오픈소스 소프트웨어(OSS : Open source Software)를 이용함으로써 복잡한 IT 환경 속에서 효율성 증대와 고가의 라이센스에 대한 비용 절감 효과 등을 통해 높은 시장성이 예상된다. 그러므로, 오픈소스 내포형 정보보호시스템에 대한 평가 제출물 준비 및 평가에 대한 기준을 정의할 필요가 있으며 공통평가기준(CC : Common Criteria)과 공통평가방법론(CEM : Common Evaluation Methodology)에서 유도한 OSS 평가요구사항을 분석하고자 한다.

1. 서 론

최근 들어 리눅스를 비롯해, 다양한 공개된 소프트웨어에 대한 시장에서의 관심이 고조되고 있으며, 제품 자체의 신뢰성이 점차적으로 높아짐에 따라 다양한 프로젝트에 적용되는 사례가 늘어나고 있는 추세이다[1]. 소스코드를 공개하고 무료로 배포된 소프트웨어(예 : OpenSSL, 아파치, Linux 등)를 오픈소스 소프트웨어(OSS : Open Source Software)라 하며 오픈소스 라이센스는 OSI(open source institute)에 의해 인증된다. 특히, OpenSSL은 내부에 각종 표준 암호알고리즘을 C로 구현한 암호모듈을 포함하며 메일의 안전한 전달을 보장한다. 정보보호제품(TOE)을 개발할 때, 보안기능의 대부분을 OSS를 이용해 구현(기존 OSS의 API 이용)하며, 따라서 OSS의 보안성은 정보보호시스템(TOE : Target of Evaluation, 평가대상물)의 보안성을 좌우한다. 현재 대부분의 정보보호시스템 개발에서 GUI부분을 제외한 소스의 상당 부분을 OSS를 이용하여 개발하며 이 경우, OSS에 대한 평가 제출물 준비 및 평가에 대한 기준을 정의할 필요가 있다.

그러므로 본 논문에서는 정보보호제품 및 시스템의 평가기준과 방법론인 CC와 CEM에 나타난 OSS관련 평가기준을 정리 및 분석하였다. 본 논문의 2장에서는 CC/CEM 및 OSS에 대한 간략한 개념을 정의하며 3장과 4장에서는 각각 CC와 CEM에서 나타난 OSS의 평가기준에 대하여 제시하도록 한다. 마지막으로 5장에서는 요약 및 결론을 맺는다.

2. 관련연구

2.1 CC/CEM (Common Criteria/Common Evaluation Methodology)

CC와 CEM은 국제적 정보화를 촉진시키기 위해서 평가기준 및 방법론의 단일화에 관심이 고조된 결과이다. CC는 미국의 TCSEC, 유럽의 ITSEC, 카나다의 CTCPEC기준을 통합한 국제표준으로써 평가의 상호인증을 위한 골격을 제시하며 우리나라의 정보통신부 표준이다. CC는 모든 정보보호제품 및 시스템에서 필요로 하는 보안기능요구사항의 전체집합을 클래스-패밀리-컴포넌트를 통해 계층적으로 분류하고 있으며 보증요구사항(컴포넌트)에 대해서 7단계(EAL1~EAL7)의 보증수준별로 정의하고 있다[2,3,4]. 또한, TOE의 제품유형에 따라 보안기능요구사항의 일부와 7단계의 보안수준 중 하나를 택하여 보호프로파일(PP : protection profile) 또는 보안목표명세서(ST : security target)를 구성한다.

CEM은 IT 보안평가 기준인 CC에 대한 지침 문서로써 적절하고 비용 효과적인 평가를 수행할 수 있는 골격을 제시하여 효율적인 평가결과를 도출할 수 있도록 한다. CEM은 CC에 정의된 기준과 평가증거를 사용하여, 평가지침에 따른 평가자가 수행해야하는 최소한의 행동을 기술하고 있다. CC의 클래스-컴포넌트-평가자요구사항은 각각 CEM의 활동-하부활동-행동과 매핑되며 몇몇 CEM 업무단위는 CC 개발자행동, 내용과 표현요소의 주된 요구사항으로부터 생성될 수 있다[2,5].

※ 본 논문은 2004년 한국정보보호진흥원 OpenSSL 암호컴포넌트 시험방법론 연구 지원에 의하여 수행되었음.

2.2 오픈소스 소프트웨어(OSS : Open Source Software)

(1) 개념

"오픈소스"는 소스코드가 공개되어 있어 누구나 사용할 수 있는 것을 말하며 1998년 OSI가 만들어지면서 "오픈소스 소프트웨어"가 생겨났다. OSS라는 용어를 사용하기 위해서는 자유로운 재배포 및 소스코드 포함, 2차적 저작물 허용, 소스코드의 무결성 보장, 그리고 라이센스 배포 등에 대한 조건을 만족해야 한다[6]. OSS를 사용하면 총소유비용(TCO)이 줄어들 뿐만 아니라 선택의 폭도 넓어지고 코드의 품질과 기능 향상 등 많은 장점이 있다. OSS제품에는 대표적으로 OpenSSL(암호라이브러리), Linux(운영체제), Apache(웹서버), Sendmail(인터넷메일 유틸리티) 등이 존재하며 OpenSSL을 사용하는 유명한 어플리케이션은 SSH와 https를 지원하는 웹브라우저(explore, Mozilla 등)와 웹서버(Apache, IIS 등)가 있다[7]. 이와 같이 OSS를 포함한 정보보호시스템 개발은 증가하고 있으며 자체적인 개발보다 시스템 전체의 상당부분을 OSS를 이용하여 보안기능을 구현하고 있다. 그림 1은 OSS와 TOE개발 및 TOE평가 프로세스간의 관계를 나타낸다.

(2) 국외 오픈소스 암호라이브러리 조사

암호라이브러리는 암호에 대한 특별한 지식이 없는 비전문가와 어플리케이션 개발자에게 짧은 시간에 국제적으로 인증된 암호 알고리즘과 인증보안 서비스를 설치할 수 있도록 하며 라이브러리 소스의 공개는 문제를 찾아 해결하는 것을 더욱 쉽게 한다. 공개된 암호 라이브러리에는 OpenSSL, Botan (formerly OpenCL) 등이 있으며 OpenSSL은 C로 작성되었고, Apache 방식 라이센스를 가지고 있다. OpenSSL은 커맨드 라인 어플리케이션의 패밀리와 같이, SSL 과 TLS 프로토콜에 대한 지원에 의해 구별된다. 또한, Botan은 ECC를 제외한 공통 PK 알고리즘(RSA, DH, DSA NR, RW) 대부분과 AES, SHA-1, DES, PSS, OAEP 등의 문서를 지원한다. 이외에도 BouncyCastle, Flexiprovider, MatrixSSL, Cryptlib, Cryptix, Crypto++, libgcrypt, LibTomCrypt, MIRACL등의 종류가 있다. 표 1은 각 라이브러리에 대해서 비교·조사한 것을 나타낸다.

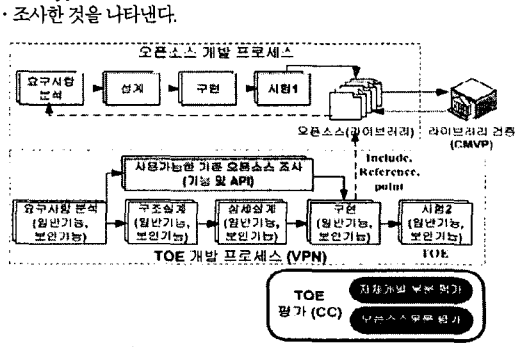

그림 1. OSS 개발, TOE 개발 및 TOE평가 프로세스간의 관계

표 1. 국외 오픈소스 암호라이브러리 비교

공개된 암호라이브러리	개발자/개발판	개발 언어	실행환경 제스트		라이센스
			OS	CPU	
Botan	Jack Lloyd/ Botan Project	C++	Unix/POSIX, Windows, FreeBSD, OpenBSD, QNX, BeOS, Linux, MacOS X, Solaris	x86, x86-64, IA-64, PowerPC, SPARC, Alpha, MIPS, ARM	수정 유무에 관계없이 모든 사용은 무료, 재배포시 라이센스 명시를 포함해야함
BouncyCastle	Bouncy Castle	C++	J2ME를 포함 모든 환경에 적합	-	라이센스 명시 포함으로 모든 목적의 사용이 가능
botZoi	Anthony Mulcahy	C++ Java	-	-	GNU GPC의 조건하에 무료 사용
Cryptrix	Cryptix Foundation Limited	Java	-	-	수정 유무에 관계없이 라이센스 명시의 포함으로 배포 및 사용 가능
Cryptlib	Peter Gutmann	C	Unix, BOS, Windows, Amiga, OS/2, BeOS, Macintosh, Tandem	-	비상업적인 사용은 무료, 상업적인 사용은 라이센스 계약이 필요
Crypto++	Wei Dai	C++	Windows, Unix, MacOS, Solaris, BeOS, DOS	-	사용, 수정, 배포무료 단, 소프트웨어 구입자는 특별에 의해 보호 받을 수 있다.
FlexiProvider	FlexiProvider Group	Java	-	-	공개 소스이며, 프로젝에 따라 LGPL, GPL 라이센스를 따른다.
libgcrypt	Free Software Foundation	C	-	-	라이센스 명시 포함으로 모든 사용 무료
Lib FomCrypt	Tom St Denis	C	-	x86 (Intel and Amd), ARM7 TDMI, PowerPc, MIPS	상업적 및 비상업적인 용도에 관계없이 공개 수정사용이 가능
MatrixSSL	PeerSec Network	C	Windows, Linux, MacOS X, SMX, pSOS, PocketPC	x86-64, I386, PowerPC, MIPS32, ARM7	비상업적인 사용은 소스코드 공개로 무료로 사용가능 상업적인 사용은 라이센스 구입 사용
MIRACL	Shamus Software Ltd	C C++	-	일레아드/DSP 확정	라이센스 명시 포함으로 모든 사용 무료
OpenSSL	OpenSSL Project	C	UNIX, VMS, Win16-32	-	상업적 및 비상업적인 용도에 관계없이 공개 수정사용이 가능

※ — : 지원하지 않거나 공개된 자료 없음

3. CC에 나타난 OSS 평가기준

3.1 가정 사항 및 분석 기준

CC에서 OSS에 대해 직접적인 평가기준으로 제시되지는 않으며 OSS는 정보보호시스템을 개발하기에 앞서 내부에 검증된 암호 알고리즘을 사용한다. 따라서, OSS에 대한 가정 사항 및 분석기준을 다음과 같이 제시한다.

- OSS는 소스코드로만 존재하며 OSS의 보안성은 TOE의 보안성을 좌우한다. 그리고 OSS를 이용한 보안환경에 대해 제시되어야 한다.
- OSS의 사용에 대한 요구사항이 존재하며 TOE에서 OSS를 이용에 대한 요구사항을 평가해야 한다.
- 설계 명세는 존재하지 않으며 개발자가 정보보호시스템(TOE) 구현시 구현 표현 등 설계 명세를 작성한다.

OSS에 대한 분석기준으로써 CC의 평가 보증등급과 상관없이 OSS와 관련된 평가기준에 대해서 개괄적으로 분석하였으며 개발자는 OSS를 포함한 TOE의 기능성 및 보안성을 보증해야 하며 문서 작성시 이를 반영해야 한다. 그리고 OSS와 자체 개발 부분을 구분지어야 한다.

3.2 CC에서의 OSS에 대한 평가요구사항 분석

(1) 보안목표명세서 평가

"TOE의 범위와 경계를 일반적인 용어를 사용하여 물리적인 방식과 논리적인 방식으로 서술"해야 하며 "TOE의 용도 및 사용 환경에 대한 가정 사항들을 식별 및 서술"해야 한다. ASE_SRE(별도로 명시한 IT 보안요구사항) 패밀리의 경우 "공통평가기준을 참조하지 않은 다른 요구사항을 명시할 수 있다고 제시"되었으며, 이는 OSS 자체에 대한 요구사항이 될 수 있다.

요구사항 개발자는 보안목표명세서에 TOE 환경 및 보안환경, 보안목적, IT 보안요구사항에 대해 명시하여야 하며 이때, OSS 부분과 자체 개발 부분을 분류하여 나타내어야 한다.

(2) 형상관리

TOE의 구현의 표현으로써 TOE내에 OSS(예 : 소스코드)가 포함되거나 이용된 사항과 절차를 요구한다. TOE에 대한 구현의표현은 물리적인 TOE를 구성하는 모든 하드웨어, 소프트웨어, 펌웨어로 이루어진다. 소프트웨어로만 구성된 TOE의 경우, 구현의 표현은 소스코드와 목적코드로만 이루어진다.

요구사항 OSS는 구현적합하다면 전자와 후자의 경우 모두 포함될 수 있으며 개발자는 형상항목 목록에 OSS를 포함함에 따른 구현표현, 보안결함, 개발도구 및 관련 정보, 보안목표명세서의 보증 컴포넌트에서 요구하는 평가증거를 포함해야 한다.

(3) 배포 및 운영

배포 및 운영 클래스는 TOE의 정확한 배포, 설치, 생성, 시동에 대한 요구사항을 정의하며 OSS의 평가와 거의 무관하다.

(4) 개발

보안목표평가 클래스는 보안목적 및 기능요구사항과 TOE 요약명세간의 일치성에 대한 요구사항도 정의하며 TOE 보안기능은 TOE 보안정책을 수행하기 위해 의존해야 하는 TOE의 모든 부분을 말한다. 또한 이 클래스에서 요구하는 명세는 비정형화된 명세(자연어), 준정형화된 명세(제한된 구문언어(예 : 제한된 문장구조), 도식(예 : 자료 구조도, 프로세스 구조도 등)), 정형화된 명세(잘 정립된 수학적 개념에 기초한 표기이며 정형화된 표기법의 구문과 의미를 정의하고 논리적인 추론을 뒷받침하는 증명 규칙들을 정의)의 세 가지 방식이 있으며 정형화수준을 높일수록 명세의 모호성은 감소된다.

요구사항 ST에 제시된 보안목적 및 기능요구사항과 개발된 OSS 부분의 보안기능이 일치하는지 입증해야 하며 이러한 보안기능이 보안정책을 수행하는지 여부를 확인해야 한다. 또한, OSS를 이용한 정형화 수준 (예 : OpenSSL의 라이브러리를 구성한 프로그래밍 언어나 표기법, 또는 정형화된 일치성 증명을 위한 검증되니 알고리즘 사용 등)을 높일수록 모호성이 감소된다.

"보안기능성"이란 TOE가 구현한 보안기능을 위하여 서브시스템이 수행하는 오퍼레이션 집합을 표현하는데 사용되며 각 서브시스템을 식별하고 제공하는 보안 기능성을 서술해야 한다. TSF의 구현의 표현은 가장 구체적인 표현을 나타내며, 예를 들어, 컴파일 될 소스코드나 하드웨어도면이 그 일부가 된다.

요구사항 OSS의 역공학하여 TOE에 제공되는 기능을 표현하여야 하며 실제로 사용한 OSS의 소스코드가 문서에 포함되어야 한다.

TSF 내부(ADV_INT) 평가에서 설계 복잡도의 최소화는 "소스코드가 이해"되어야 한다는 보증에 기여한다. TSF의 코드가 덜 복잡할수록 TSF의 설계를 이해할 가능성이 높아지며 설계 복잡도의 최소화는 참조 검증 메커니즘의 주요한 특성이며 EAL5등급부터 적용된다.

요구사항 보다 높은 수준의 평가보증등급(EAL)을 받고자 한다면 OSS의 설계 복잡도를 최소화하여 이용해야 할 것이다.

(5) 설명서

설명서는 관리자용, 일반사용자용, 소프트웨어 또는 하드웨어 인터페이스를 사용하는 응용 프로그래머 및 또는 하드웨어 설계자용 등 분리된 문서로 제공되어야 한다. OSS 평가와는 직접적으로 연관되지 않으며 시험 클래스와 연관해서 OSS 평가에 간접적인 영향을 미친다.

(6) 생명주기 지원

결함교정(ALC_FLR)의 경우 모든 유형의 결함을 다루기 위한 방법을 서술해야 하며 생명주기 정의(ALC_LCD)에서는 "측정 가능한 생명주기 모델", 즉 TOE 개발 특성(예 : 소스코드 복잡도)을 측정하는 산술 매개변수 및 또는 척도를 갖는 모델을 서술해야 한다. 또한 도구와 기법(ALC_TAT)에서는 TOE의 개발, 분석, 구현 등에 사용되는 도구를 선택하는 측면이며 프로그래밍 언어, 문서화, 구현표준 등과 실행시간 라이브러리 지원과 같은 TOE의 다른 부분들을 포함하지만 이에 한정되지는 않는다. "개발자에 의해 적용되는 구현 표준과 추가로 제 3자의 소프트웨어나 하드웨어, 또는 펌웨어를 포함한 "TOE의 모든 부분"에 대한 구현 표준을 구분한다.

요구사항 개발자는 OSS를 포함한 TOE에 대해서 OSS 부분과 자체 개발한 부분 등 구현 표준을 구분하여 서술해야 하며 OSS의 소스코드 복잡도 등을 측정하는 측정가능한 생명주기 모델을 적용해야 한다.

(7) 시험

시험은 발견되지 않은 결함이 발생할 가능성이 비교적 적다는 것을 보증하는데 기여하며 서브시스템과 모듈을 그 명세에 따라 시험하는 것과 같이 TSF 내부구조에 대한 시험이 수행될 수도 있다. 개발자는 개발자 시험의 효율적인 재현을 위하여 필요한 자료를 평가자에게 제공해야 하며 이는 기계 판독형 시험문서나 시험 프로그램 등을 포함한다.

요구사항 개발자는 OSS를 포함한 TOE에 대해서 시험을 수행하여야 하며 이에대한 결과와 시험의 효율적인재현을 위해 필요한자료(예: 기계 판독형 시험문서나 시험프로그램)를 평가자에게 제공해야 한다.

(8) 취약성 평가

TOE 보안기능이 우회 또는 비활성화되거나 손상될 수 없더라도 하부 보안 메

커니즘 개념에 취약성이 존재하기 때문에 여전히 파괴될 수 있으며 하부 보안 메커니즘의 보안 행동에 대한 양적 또는 통계적 분석 결과와 이를 극복하는데 필요한 노력에 의하여 보안 행동의 강도가 결정된다. 이는 TOE 보안기능 강도 선언에 의해 이루어진다. 또한 취약성 분석(AVA_VLA)에 제시된 개발자 요구사항에서 개발자는 식별된 취약성의 특성을 문서화하여야 하며 증거 요구사항으로서 어떠한 식별된 보안 취약성도 TOE의 예상된 환경 내에서 악용될 수 없음을 확인하고 TOE가 명백한 침투공격에 내성이 있음을 확인하여야 한다.

요구사항 OSS에 속한 암호 모듈 및 기능 모듈에 대한 구현적합성 검증이 보안기능 강도를 말하는 것은 아니며 OSS의 보안 기능 강도를 결정하고 평가하는데 위의 요구사항을 만족해야 한다. 또한 개발자는 식별된 취약성의 특성을 문서화해야 하는데, OSS는 대부분 취약성이 공개되어 있다. OpenSSL의 경우 OSS로서 CAN(취약성 후보)17개, CVE(취약성) 2개 등 취약성이 존재한다.

4. CEM에 나타난 OSS 평가요구사항(EAL4를 중심으로)

CEM은 CC 평가 지침에 따른 평가자가 수행해야하는 최소한의 행동을 기술하고 있으며 EAL4는 적당히 높은 보증을 제공한다. 보안 기능은 기능명세, 지침 문서, TOE의 상위 수준 설계, 그리고 보안행동을 이해하기 위해 구현의 부분집합을 이용하여 분석하였으며 EAL4를 중심으로 OSS 에 관한 평가기준을 다음과 같이 분석 및 제시하도록 한다. 제시하지 않은 클래스는 OSS 평가요구사항과 관련이 적으므로 생략하였다.

(1) 형상 관리

■"자동화된 생성 절차" 조사 : 예를 들어, 소프트웨어 TOE에 있어서 자동화된 생성 절차는 TSP의 시행을 위해 믿을 수 있는 모든 소스 파일과 관련 라이브러리가 컴파일 된 목적 코드에 포함되었음을 보증하는데 도움이 되는지 체크하는 것을 포함한다.

■"자동화된 도구의 이용 방법" 관련 정보 조사 : 도구에 의해 제공되는 기능성(OSS의 경우 그 자체의 기능이 될 수 있음), 개발자에 의한 이용 등에 대해 포함하여야 한다.

■"CM 계획" 조사 : 개별적인 형상항목 상에 오퍼레이션을 수행하기 위해 요구되는 개인의 역할과 책임(서로 다른 역할들은 서로 다른 형태의 형상항목(예 : 설계문서 혹은 소스코드)에 대해 다른 역할들이 식별)을 포함하여야 한다.

■"인수절차" 조사 : TOE 구성의 각 단계 (예: 모듈, 통합, 시스템), 소프트웨어 및 펌웨어, 하드웨어 컴포넌트의 인수 또는 성공적인 시험을 포함해야 한다.

■ 형상 목록에는 TOE의 구현표현 및 구현과 관련되어 보고된 보안결함들에 관한 상태보고서 등 CC에서 요구하는 항목의 집합을 포함해야 한다.

(2) 개발

개발 활동의 목적은 TSF가 TOE의 보안기능을 어떻게 제공하는지를 이해하기 위해 설계 문서를 평가하는 것이므로 OSS를 포함한 TOE는 보안기능 제공에 대한 상세한 설계 문서를 제공해야 할 것이다.

■ 모든 "TOE 보안기능 인터페이스 식별" 조사 : TOE에서 외부 인터페이스는 TSF에 대한 직접적인 인터페이스뿐만 아니라 비-TSF 부분에 대한 인터페이스를 말한다. TSF에 직접 또는 간접적으로 접근하는 이 외부 인터페이스는 집합적으로 TOE 보안기능 인터페이스(TSFI)를 만들며 OSS 부분과 자체 개발 부분이 결합되어 TSFI를 만들 수 있으므로 그러한 사항을 문서에 포함해야 한다.

■"TSFI의 표현" 조사 : 어떻게 보안 요구사항들이 인터페이스에 적용되었는지 TSF 표현에 제시되지 않은 보안기능은 기술해서는 안 된다. 예를 들어, OSS를 이용한 보안기능 구현에서 PP나 ST에 표현되지 않은 사항을 기술해서는 안 되며 이럴 경우 "fail" 평결을 유도한다.

■"상위수준 설계" 조사 : 표현은 TOE가 하드웨어, 펌웨어 또는 소프트웨어(TOE가 TOE 보안 목적을 지원하기 위해 나름의 IT 환경에 대한 보안 요구사항을 구현한)내에 제공되는 기능들을 어떻게 사용하는가를 설명해야 하며 TOE에 대한 필요한 보안기능을 제공하는지를 결정해야 한다. 여기에서 소프트웨어는 OSS가 될 수 있다.

■"구현표현의 내부적 일관성" 조사 : 예를 들어, 원시 코드의 경우에 원시 코드의 한 부분이 다른 부분에의 서브프로그램을 호출한다면 평가자는 호출한 프로그램의 인수들이 호출된 프로그램의 인수들의 처리와 일치하는지를 알기 위해 조사해야 한다.

■"모듈간의 상호관계 정의" 결정 : 모듈들이 다른 것에 서비스를 제공하고, 보안 기능들을 지원한 협력으로 상호작용하는 것을 보여야 한다. 또한 각 TSP-수행 기능이 정확하게 지원되고 명시하기 위해 필요한 모듈관의 상호관련성을 보증해야 한다. 모듈은 OSS일 수 있다.

■"개발보안문서" 조사 : 절차에 따라 보증하기 위하여 문서증거 작성을 조사하며, 작성된 증거의 예는 엔트리 로그 및 감사 추적을 포함할 수 있다. 또한 보안수단 적용에 대한 관찰과 절차적용의 문서증거를 조사해야 한다.

(3) 생명주기지원

■"생명주기 모델" 조사 : TOE 개발 및 유지보수에 사용되어질 절차, 도구와 기술을 포함해야 하며 모델의 설명은 개발자에 의해 사용된 절차, 도구와 기술에 대한 정보를 포함해야 한다(예 : 설계, 코딩, 시험, 고장 수리). 또한 프로그래밍언어 명세 및 사용자매뉴얼 등에 명확한 문장을 사용해야 한다.

(4) 시험

■"시험 범위 및 깊이 분석" 조사 : 기능 명세 내에 설명된 모든 보안 기능과 인터페이스는 시험 범위 분석에 표현될 것이며 완전성이 선언되기 위해서 보안기능과 인터페이스에 대응되는 시험들을 가져야 하고 설명을 포함해야 한다. 또한 깊이는 상위수준 설계에 설명된 모든 서브시스템들을 식별하고 이러한 서브시스템들에 시험의 대응을 제공한다. 시험은 시험 전체 목표에 의존하는 하나 혹은 다수의 보안기능을 포함한다.

■"시험"결과 조사 및 수행 : 시험될 TSF의 부분집합을 선택할 때 평가자는 특정한 보안기능들에 대한 개발자 시험증거 및 시험부분집합에서 추출되는 보안기능의 수 등을 고려해야하며 개발자 시험결과 및 수행한 시험의 엄격함, 보안기능의 중요성, ST내의 SOF요구 등이 시험부분집합크기의 선택에 영향을 준다.

■"시험 재현을 위한 자료" 제공 : 평가자는 일련의 시험케이스들을 이용하여 각 보안기능을 시험하는 것이 실질적임을 알 것이다. 평가자는 시험을 재생산할 수 있도록 설명된 시험부분집합에 대한 시험문서를 작성해야 한다. 이때, 사용될 접근법(예외상황에서의 코드점검), 보안기능 인터페이스, 시험초기조건, 보안기능 관찰을 위한 특별한 시험장치 등은 개발자에 의해 제공되어야 한다.

(5) 취약성 평가

■"SOF 선언 분석 제공" 체크 : 공격 가능성에 관하여 정의된 SOF-기본, SOF-중간, SOF-높음 중의 하나로써 표현되며 등급으로써 표현된 최소한의 전체 SOF 요구사항은 모든 확률이나 치환 보안 메커니즘에 대해 적용한다.

■"취약성 분석" 조사 : 적어도 모든 평가 자원과 공개 도메인 정보원에 있는 취약성에 대한 개발자 조사가 반영되어야 한다. OpenSSL의 경우 OSS로서 19개의 공개된 취약성이 존재한다. 평가자는 또한 개발자 취약성 분석을 기본으로 하는 침투 시험을 고안해야 한다.

5. 요약 및 결론

CC의 모든 평가 요구사항들에 OSS에 대한 직접적인 평가기준으로 적용되지는 않으며 OSS는 TOE의 부분집합으로써 TOE를 평가하면서 부수적인 평가 부분집합이라 할 수 있다. 물론, 요즘 OSS가 TOE의 대부분을 차지하므로 CC 평가시 TOE의 구현방법(즉, 자체개발했는지 오픈소스를 이용했는지)에 상관없이, TOE 개발자는 CC 기준에 따라 각종 제출물을 제공해야 하며 평가자는 이를 평가해야 한다. CC와 CEM에서 자체 개발한 TOE와 오픈소스 내포형 TOE의 평가 방법은 상당부분 일치하며 검증된 OSS라고 해서 오픈소스 내포형 TOE가 평가된 것은 아니므로 새로이 평가가 이루어져야 한다. 또한, OSS는 TOE 평가의 일부분으로 수행되며 계속해서 오류 보고 등 문제점이 발생되면 OSS를 포함한 TOE도 업그레이드 및 재평가가 이루어져야 할 것이다.

참고문헌

[1] CRISPIN COWAN., "Software Security for Open-Source Systems," IEEE SECURITY & PRIVACY, 2003.

[2] 한국정보보호진흥원, "정보보호시스템 평가·인증 가이드," KISA, pp.52-68, 2002. 12.

[3] Part 3 : Security Assurance Requirements, Common Criteria for Information Technology Security Evaluation(CC), CCIMB-2004-01-003, Version 2.2 : ISO/ IEC 15408, Jan. 2004.

[4] European Community, "Information Technology Security Evaluation Criteria (ITSEM)," Ver.1.0, CESG, 1993.

[5] Part 2 : Evaluation Methodology, Common Evaluation Methodology for Information Technology Security(CEM), CCIMB-2004- 01-04, Version 2.2, Jan. 2004.

[6] OSI, "Open Source Definition (version1.9)," http://www.opensource.org/docs/definion.php, 2004.

[7] Stefano Comino, Fabio M. Manenti, "Open Source vs Closed Source Software :Public Policies in the Software Market," 2003. 6.