**정보보안기사(최종수정일 8/19)**

**4장 시스템 보안**

**12절 보안 운영체제**

보안 운영체제란 기존 운영체제 내에 시스템 보안 기능을 통합한 보안 커널을 추가로 이식한 운영체제이다. 기술적 핵심은 정보영역의 분리, 역할기반 영역분리, 최소권한 유지, 커널 레벨의 강제적 접근통제 등을 들 수 있다.

**보안 운영체제의 주요 제공 기능**

**보호대상:** 메모리, 보조기억장치 상의 파일 또는 데이터 집합, 메모리상에서 실행 중인 프로그램, 파일들의 디렉터리, 하드웨어 장치

**보호방법:** 물리적 분리(사용자별 하드웨어 분리), 시간적 분리(프로세스가 시간 당 하나만), 논리적 분리(프로세스에 논리적 구역 제한), 암호적 분리(내부 사용 정부를 외부에서 알 수 없도록).

나열된 방법들은 복잡도가 증가하는 순이다. 분리방법을 조합해 이용할 수도 있다.

**파일시스템 보호**

파일의 지나친 보호는 자원낭비, 지나친 공용은 보안성 상실을 야기한다. 다양한 파일 보호기법으로 균형을 찾아야 한다.

1. **파일의 이름 명명(Naming):** 다른 사용자의 파일 이름을 알 수 있는 기법이 없고 추측도 어렵다는 가정에 기초한다.
2. **패스워드:** 사용자마다 패스워드를 부여하는 방법.
3. **암호화:** 파일 내용을 암호화 해 누구나 공유할 수 있지만 인가된 사용자만 내용을 알 수 있게 한다.

**보안 운영체제와 보안커널**

**보안 운영체제의 보안 기능:**

1. **사용자 식별 및 인증**
2. **임의적(DAC, Discretionary Access Control-신분 기반 정책)/강제적(MAC, Mandatory Access Control-규칙기반정책) 접근 통제**
3. **객체 재사용(Object Reuse) 보호:** 재사용 가능한 저장매체에서 한 번 저장된 데이터가 삭제되지 않고 존재하는 경우가 많다. 이를 통한 비밀 데이터 유출(객체 재사용)을 막기 위해 재할당되는 모든 기억장치 공간을 깨끗이 지우는 것이 객체 재사용 보호이다.
4. **완전한 조정:** 임의적/강제적 접근통제가 효과적이기 위해서는 모든 접근을 통제해야 한다. 따라서 보안 운영체제는 모든 접근을 통제하는 완전한 통제를 수행해야 한다.
5. **신뢰 경로(Trusted Path):** 패스워드 설정, 접근 허용 변경 등과 같은 보안 작업 수행 시에 사용자의 안전한 통신 경로를 제공할 수 있어야 한다.
6. **감사 기록(Audit Logging) 및 기록 축소:** 모든 보안 관련 사안은 감사 기록부(Audit Log)에 반드시 기록되어야 하고 감사 기록부는 명백히 보호되어야 한다.

**보안 커널(Security Kernel):** 신뢰 컴퓨팅 기반TCB(Trusted Computing Base) 내에 있는 하드웨어, 소프트웨어, 펌웨어로 구성된다. 참조 모니터 개념을 구현하고 집행. 보안 커널은 주체와 객체 사이의 모든 접근과 기능을 중재한다. 보안 커널 자체가 TCB의 핵심이다.

참조 모니터 개념을 수행하는 프로세스를 위해 분리가 제공되고 프로세스는 변조 방지 기능이 있어야 한다. 모든 접근에 대해 실행되어야 하고 우회가 불가능해야 한다. 따라서 보안 커널은 완전하고 오류가 없어야 한다. 또한 완전하고 포괄적인 방식으로 충분히 시험되고 확인되어야 한다. 보안 커널의 도식은 반원형과 원형(허니웰의 멀틱스)이 있다.

**신뢰 컴퓨팅 기반(TCB, Trusted Computing Base):** 하나의 컴퓨터 시스템(H/W, F/W, S/W) 내의 모든 보호 메커니즘의 총체이다. 시스템과의 조화를 통해 보안 정책을 적용할 책임을 갖는다. 신뢰된 경로는 사용자, 프로그램, 커널과의 통신채널을 말한다. TCB는 어떤 경우에도 이 채널이 훼손되지 않도록 보호 기능을 제공한다. **보안경계(Security Perimeter)**는 시스템의 구성요소 중 신뢰/비신뢰 구성요소의 경계이다. 물리적 실체가 아니라 신뢰/비신뢰 구성요소 사이에 선을 긋기 위해 운영체제에서 사용하는 개념적 선이다. TCB는 안전을 위한 건축설계 기준과 같다고 생각하자.

**참조모니터(Reference Monitor):** 주체의 객체에 대한 접근통제를 담당하는 추상 머신이다. 다음 세 가지 규칙을 가진다.

* 반드시 부정조작이 없어야 한다 – Must be tamperproof
* 항상 무시되지 않고 호출되어야 한다 – Always invoked
* 모든 동작을 항상 분석과 테스트를 통해 확인할 수 있어야 한다 – Must be Verifiable

주체가 참조 모니터를 거치지 않고 객체에 직접 접근하는 경우가 참조 모니터를 위반하는 경우이다. 참조 모니터는 보안커널 데이터베이스(SKDB)를 참조해 객체에 대한 접근허가 여부를 결정한다.

**신뢰 플랫폼 모듈(TPM)**

Trusted Platform Module이란 신뢰 컴퓨팅 그룹 TCG라는 산업체 컨소시엄에서 표준화된 개념이다. TC에서 핵심이 되는 하드웨어 모듈을 말한다.

**TPM의 특징:** 신뢰 컴퓨팅의 가장 하위에 위치한다. 훼손 방지(Tamper Protected)가 필수이기에 하드웨어칩으로 구현하는 게 일반적이지만 소프트웨어로 구현하기도 한다. 칩으로 구현하면 소프트웨어 공격과 물리적 도난에도 정보의 탈취가 어렵다는 장점이 있다. TPM에는 신뢰 관련 연산이 있다. 1. 암호화 키의 생성/저장 2. 패스워드 저장 3. 무결성(플랫폼/프로그램)의 검증을 위한 측정값의 저장 4. 디지털 인증서 관련 신뢰 연산 등이 있다.

**TPM의 기본 서비스:** 인증된 부트 서비스(단계별 부트, 적재될 때마다 승인된 버전임을 보장), 인증 서비스(사설키를 사용해 설정 정보에 서명함으로써 디지털 인증), 암호화 서비스

**TPM 함수:** 다음과 같은 구성요소로 이루어진다. I/O, 암호화 보조프로세서, 키 생성(RSA 공유/사설키 쌍과 대칭키), HMAC 엔진(인증 프로토콜에서 쓰는 알고리즘), 무작위 숫자 발생기(RNG, Random Number Generator), SHA-1 엔진, 전원 탐지, Opt-in(고객 의도대로 TPM을 활성/비활성화할 수 있는 안전한 메커니즘), 실행 엔진(I/O포트에서 받은 TPM명령어 실행), 비휘발성 메모리, 휘발성 메모리

**13절. 클라이언트 보안**

**악성 소프트웨어(악성 코드)**

**바이러스:** 자기복제 O, 파일/부트섹터 등 감염 대상 필요, 사용자가 감염된 파일 옮김, 시스템 및 파일 손상을 목적으로 함

**트로이목마:** 자기복제 X, 정상 프로그램으로 위장 or 정상 프로그램에 코드로 삽입, 사용자가 실행시킴, PC 성능 저하나 통제권 획득을 목적으로 함

**웜:** 자기복제 능력이 매우 강함, 독자적으로 존재하며 네트워크를 통해 스스로 전파한다. 네트워크 성능 저하를 목적으로 한다.

**악성 소프트웨어 분류:**

1. **독립형과 기생형:** 호스트를 필요로 하는지 아닌지 여부로 구분. 전자는 기생형. 독립적으로 존재할 수 없다. 바이러스, 논리폭탄, 백도어 등이 해당된다. 후자는 자체적으로 구동 가능한 프로그램으로 운영체제가 스케줄 해 구동할 수 있다. 웜과 좀비가 해당
2. **자기 복제 여부:** 바이러스성 – 웜, 바이러스 / 비-바이러스성 – 트로이목마, 백도어

**바이러스:** 타 프로그램을 변형시켜 감염시키는 프로그램. 자신을 변형시켜 실행 가능하기도 하다. 변형이란 원래 프로그램에 루틴을 주입해 바이러스 프로그램의 복제를 만들어내는 걸 말한다. 바이러스는 자신을 완벽히 복제할 수 있는 비법을 가진 명령 코드를 가진다. 또 전형적으로 프로그램이나 실행 콘텐츠의 전송자 속에 내장된다. 세대별로 분류하면 다음과 같다.

1. **제1세대 원시형 바이러스(Primitive Virus):** 단순하고 분석이 쉬운 바이러스. 코드의 변형이나 변화 없이 고정된 크기를 가지며 기억장소에 상주해 부트영역이나 파일을 감염시킨다.
2. **제2세대 암호화 바이러스(Encryption Virus):** 백신 프로그램이 진단할 수 없도록 바이러스 프로그램의 일부 또는 대부분을 암호화시켜 저장한다. 실행 시작점 부분에 위치하는 암호해독 데이터는 항상 일정하다.
3. **제3세대 은폐형 바이러스(Stealth Virus):** 기억장소에 존재, 감염된 파일의 길이가 증가하지 않은 거처럼 보이게 한다. 백신 프로그램이 감염된 부분을 읽으려 하면 감염되기 전의 내용을 보여줘 바이러스가 없는 것처럼 속인다. 이렇게 스스로 모습을 바꾸는 바이러스는 다형성 바이러스라고 한다.
4. **제4세대 갑옷형 바이러스(Armor Virus):** 백신 프로그램으로부터 숨기보다 여러 단계의 암호화와 다양한 기법을 동원해 바이러스 분석을 어렵게 하고 백신 개발을 지연시킨다.
5. **제5세대 매크로 바이러스:** MS 오피스, Visio, AutoCAD 등 Visual Basic Script를 지원하는 다양한 프로그램에서 활동해 현재 등장하는 바이러스 중 가장 높은 비중을 차지한다.

**매크로 바이러스:** 매크로 바이러스는 플랫폼과 무관하며 쉽게 퍼지고(이메일) 실행 파일(COM, EXE)을 다룰 때 주의를 덜 하기 때문에 피해가 크다. 특히 어도비 사의 PDF에 자바스크립트 같은 다양한 스크립트를 포함하는 형태로 많이 유포된다.

**바이러스 방지책:**

1. **안티 바이러스:** 탐지, 식별, 제거를 통해 바이러스 예방 방지
2. **안티 바이러스 필터링 방법:** 대표적으로 virus signature와 behavioral virus scanning이 있다.

* **signature scanning:** 특정 바이러스만이 가진 시그내쳐를 찾아낸다. 이는 실행 가능한 코드(excutable code) 상에서의 이진 스트링, 스크립트에서는 명령어의 나열로 파악할 수 있다. 대부분의 안티 바이러스 프로그램들이 채택하는 방법이다.
* **behavioral virus scanning:** 바이러스가 수행 중에 어떤 행동을 보이는지 추적하는 방법. 시그니처 스캐닝에서 대처가 불가능했던 새 바이러스와 웜에 대한 대처 능력을 기를 수 있다.

1. **바이러스 예방:** 신뢰성 있는 업체에서 구매한 상업용 소프트웨어만 사용해야 하며 정기 바이러스 스캐닝, 백신 업데이트 등이 필요하다. 이때 Windows Script Host, ActiveX, VBScript, JavaScript는 비활성화한다.

**웜:** 자가 복제해 네트워크 연결을 통해 컴퓨터에서 컴퓨터로 복제본을 전송한다. 이메일 자체가 시스템에서 시스템으로 자신을 확산시키기 때문에 웜의 성격을 가지고 있다. 다른 시스템에 직접적인 영향을 미치지 않아 트로이목마와 구분되며 다른 프로그램에 기생하지 않는다는 점에서 컴퓨터 바이러스와 구분된다.

**웜 확산 모델:** 확산은 3단계로 이뤄진다. 감염된 호스트가 각각 두 개의 호스트를 감염시킨 뒤 다시 감염된 호스트들은 지수적으로 증가한다. 이후 고점을 지나면 바이러스가 서서히 퍼지며 종료 단계에 접어든다.

**웜의 실행:** 버퍼 오버플로우, 포맷 스트링, SQL 인젝션, PHP 인젝션 등 공격 가능한 시스템 취약점을 이용해 시스템의 접근 권한을 확보하고 자신을 실행시킨다. 접근 권한을 확보하면 웜을 백도어로 설치하거나 DDoS 공격에 사용할 수 있는 봇이나 좀비, 키로거 프로그램을 다운로드해 실행한다.

**웜 대응책:** 바이러스 대응책과 유사하다. 안티 바이러스와 네트워크 모니터링을 통해 기본적인 방어를 할 수 있다.

* **네트워크 기반 웜 방어:** 웜 모니터링 소프트웨어이다. 상호 연결된 LAN의 집합체로 구성된 사이트의 엔터프라이즈 네트워크에 두 가지 유형의 모니터링 소프트웨어가 필요하다.

1. **진입 모니터(Ingress Monitors):** 엔터프라이즈 네트워크와 인터넷 사이 경계에 위치한다. 경계 라우터나 외부 침입차단시스템이나 독립된 수동 모니터의 진입 필터링 소프트웨어의 일부이다.
2. **진출 모니터(Egress Monitors):** 나가는 트래픽에 대한 스캐닝 흔적이나 의심스러운 행동을 모니터링해 출처를 잡아내도록 설계되었다.

**트로이목마(Trojan Horse):** 정상 프로그램으로 가장하거나 정상 코드에 삽입된 형태로 다운/실행된다. 이후 권한을 획득해 해당 시스템을 통제하는 것을 목적으로 한다. 데이터 파괴를 목적으로 하기도 한다. 대표적으로 Back Orifice는 1998년 9월 CDC(Cult of the Dead Cow)에서 발표한 Win 95/98/2000 관리자용 도구이며 백도어로 이용하기 적합하도록 개발되었다.

**트로이목마 대응책:** 신뢰할 수 없는 파일을 설치/실행하지 않는다. 취약점 패치를 항상 최신 상태로 유지한다. 안티 – 트로이 목마 도구도 있으니 활용하면 된다.

**스파이웨어:** 민감한 정보를 수집해 주기적으로 원격지의 서버로 보내는 프로그램이다. 신원 도용, 스패밍, 사기 등과 같은 악의적 활동에 사용. 브라우저 사용 정보를 수집해 표적 광고를 보내는데 활용되기도 한다.

**기타 알아둬야할 악성 소프트웨어:** 논리폭탄, 모바일 코드, Exploit, Downloaders, Auto-rooter, Kit, 스패머 프로그램, 플러더, Rookit, 공격 킷, 크라임웨어, 브라우저 하이재커, 다이얼러, 조크, Hoax

**인터넷 활용 보안**

**브라우징 보안:** 사이트에 따라 서로 다른 보안 요구사항이 필요할 때가 있다. 브라우저는 이를 관리하기 위해 다양한 보안영역을 설정하고 사용자가 정의할 수 있게 해야 한다.

**설정 가능한 영역:** 인터넷, 로컬 인트라넷, 신뢰할 수 있는 사이트, 제한된 영역

**쿠키:** 1994년 넷스케이프에서 처음 사용. 사용자가 웹 사이트에 방문할 때 생기는 4KB 이하의 파일이다. 사용자와 웹 사이트를 연결해주는 정보가 담겨 서버가 클라이언트의 신원을 알 수 있게 한다.

**쿠키를 이용한 통신:**

1. 사용자가 웹사이트 방문시 웹사이트는 사용자의 컴퓨터에 쿠키를 만든다. 브라우저를 확인하는 ID 번호를 쿠키에 넣어 사용자의 컴퓨터에 저장하는 식이다.
2. 사용자가 해당 서버를 다시 방문하면 쿠키를 서버로 다시 전송하는 단계이다. 사이트는 신원이 확인된 사용자의 정보를 이용해 쾌적한 서비스를 제공할 수 있다.

**쿠키의 일반적인 용도:**

1. **사이트 개인화:** 아이디, 비밀번호, 성향을 파악할 수 있다.
2. **장바구니 시스템**
3. **웹 사이트 이용방식 추적**
4. **타겟 마케팅**

**쿠키의 구조:** 4개의 속성, 하나의 데이터로 구성된다. 서버는 응답에 Set-Cookie 헤더를 포함시켜 쿠키를 설정한다. 기본 구조는 다음과 같고 값을 추가할 경우 세미콜론을 구분자로 추가한다.

**Set-Cookie: name=value; expires=[Date]; domain=[Domain]; path=[Path]; [secure]**

* **유효기간:** 쿠키 데이터는 브라우저를 끄기 전까지 사용할 수 있지만 기간이 설정되어 있으면 브라우저를 재시작한 뒤에도 적용할 수 있다. 유효 기간이 지난 쿠키는 소멸된다. 파일이 지워지지 않더라도 이를 브라우저에서 읽을 수 없다.
* **패스:** 한 번 쿠키 데이터를 생성하면 쿠키 데이터를 생성한 웹 페이지만 해당 데이터를 읽을 수 있다. 하지만 Path 항목을 지정해주면 해당 Path에서는 그 쿠키 데이터를 공유할 수 있다.
* **도메인:** 패스 속성을 확장한 것이다. 패스가 하나의 사이트에서 쿠키를 읽고 쓰는 권한을 설정하는 것이면 도메인은 도메인 단위에서 쿠키 데이터를 읽고 쓰는 권한을 설정하게 된다.

**HTTP 쿠키 관련 보안 속성:**

* **httpOnly 속성:** Set-Cookie 응답 헤더에 설정하는 속성으로 클라이언트에서 스크립트를 통해 해당 쿠키에 접근하는 것을 차단해주는 속성. 세션 쿠키(세션ID 보유)를 탈취하는 XSS(Cross Site Script)공격에 대응하기 위해 사용한다.
* **secure 속성:** Set-Cookie 응답 헤더에 설정하는 속성으로 클라이언트에서 HTTPS 통신일 경우에만 쿠키를 전송하고 HTTP일 경우 전송하지 않는다. 전송구간 암호화를 통해 평문 전송을 막기 위함이다. 기밀성 보장 목적

**쿠키에 대한 오해:**

* **바이러스 전파:** 쿠키는 텍스트 파일로 실행되지 않는다. 4KB 이하의 파일로 바이러스를 전파할 수도 없다.
* **사용자 컴퓨터에 피해 입히기:** 읽기/쓰기를 할 수 없는 텍스트이기에 불가능하다.
* **다른 웹사이트에서 읽기:** 이름, 값, 유효기간, 도메인, 경로 등으로 이루어져 저장된 도메인 이름을 가지고 있는 사이트에서만 유효하다.

**쿠키 보안 취약점:**

1. **XSS(Cross-Site Scripting) 공격:** 서버사이드 스크립트가 아닌 자바스크립트나 비주얼베이직 스크립트 등이 클라이언트의 컴퓨터에서 실행된다는 점을 이용한 공격이다. document.cookie라는 명령어는 사이트에서 쿠키값을 활용해 다양한 서비스를 제공할 수 있도록 하지만 공격자들은 쿠키값을 유출하기 위해 사용한다.
2. **스니핑 공격을 이용:** 쿠키값을 암호화하지 않으면 스니핑을 통해 쿠키를 탈취할 수 있다.
3. **공용 PC에서 쿠키값 유출:** 쿠키는 하드에 저장되어 접근이 열려있기 때문에 같은 컴퓨터 이용시 유출될 수 있다.

**세션:** 개별 클라이언트 상태정보를 서버에 저장하는 기술이다. 개별 클라이언트를 식별하기 위해 세션 아이드를 부여하고 세션 아이디를 세션 쿠키를 통해 주고받는다. HTTP 세션 하이재킹은 공격자가 사용자의 세션 ID를 탈취해 사용자인 것처럼 위장 접근하는 공격이다.

**안전한 세션 ID관리:** 안전한 서버에서 생성 후 사용해야 하며 최소 128비트 길이로 생성되어야 한다. 또 난수 알고리즘을 적용해 예측 불가능한 값이어야 한다. URL Rewrite 기능 사용시 아이디가 URL에 노출될 수 있어 사용하지 않도록 설계하고 장시간 접속시 세션 ID 노출 위험이 점점 커지므로 주기적으로 세션 ID를 재할당하도록 설계한다.

**14절. 윈도우 서버 보안**

**윈도우 시스템의 구조**

윈도우 운영체제는 5개의 링 기반이다. 하드웨어 – HAL(Hardware Abstraction Layer) – 마이크로 커널 – 각종 관리자 – 응용 프로그램 순으로 구분한다. 관리자 계층 까지를 커널 모드로 구분하고 마지막 응용 프로그램 계층을 사용자 모드로 구분한다. 드라이버 개발자와 하드웨어 개발자는 윈도우에서 제시한 표준인 HAL만 따르면 된다. HAL은 하드웨어와 소프트웨어 간의 통신을 번역한다.

~187p