ETRI

전자통신동향분석 제 23권 제 4호 2008년 8월

모바일 단말 보안 운영체제 기술 동향

The Trends of Secure Operating System Technology for Mobile Platform

21세기를 대비하는 정보보호 특집

배근태 (G.T. Bae) 임베디드보안기술연구팀 연구원
김기영 (K.Y. Kim) 임베디드보안기술연구팀 팀장

목 차

PC, 노트북 컴퓨터, 스마트폰, PDA, PMP, 텔레매틱스 단말 등 다양한 정보단말은 소형화, 모바일화, 복합화 추세에 있으며 이에 따라 다루는 정보, 접근하는 인터페이스, 서비스의 종류가 다양해지고 단말 자체의 이동성이 증가하여 보안 위험이 높아지고 있다. 또한 단말의 컴퓨팅 성능과 서비스의 질적 향상으로 모바일플랫폼이 데스크톱 환경 수준의 일반적인 환경으로 발전하게 되어 요구되는 보안 수준 역시 높아지고 있다. 본 고에서는 Linux, Windows 등 데스크톱 운영체제를 비롯하여 WIPI 등 모바일 플랫폼까지 보안 요구사항과 기술 추세를 살펴보고 향후 발전방향에 대하여 논의해 보고자 한다.

I. 서론

정보기술(IT)의 출현 이래로 바이러스, 해킹, 정보유출 등으로 표현되는 보안 문제는 항상 따라다니는 숙제였다. 최근에는 유비쿼터스라는 키워드가 묘사하듯이 단독적으로 활용하던 정보 자원들을 네트워크를 통하여 공유하고 또한 과거의 데스크톱 또는 서버 환경이 강력한 유무선 통신기술을 통해 점차 다양한 장치 기반의 모바일 환경으로 확대되면서 보안 이슈 역시 그 범위가 걷잡을 수 없이 커져가고 있다. 단적인 예로 이제는 휴대폰에서도 바이러스를 걱정해야 하는 시대가 된 것이다[1]. 또한 서비스 관점에서도 모바일 웹 2.0, AJAX, SOA, Java Mobile 등의 최신 기술로 설명되는 서비스 중심의 양방향 참여의 트렌드로 발전해가고 있다. 즉 과거 수동적으로 사용자가 서버로부터 정보를 제공받던 개념에서 벗어나 개개인이 모두 인터넷에 참여하여 다양한 미디어 콘텐츠를 양방향으로 자유롭게 소비, 활용, 생산하는 이용자 참여 중심의 인터넷 환경으로 변화하고 있는 것이다. 이에 따라 단순한 데이터뿐만이 아닌 애플리케이션, 사용자 컨텍스트 정보, 단말 상황 정보 등이 고수준(서비스 수준)에서 사용자가 인지하지 못하는 사이에 수없이 전파되어 정보가 유출, 침해될 확률이 높아지고 있다. 우연히 악의적인 Javascript 웹페이지에 접속하는 것만으로 단말의 중요정보가 유출될 수 있으며, 단말을 네트워크에 연결시켜 두는 것만으로도 모바일 코드 바이러스에 감염될 수 있는 것이다.

특히 스마트폰, PDA, PMP 등 개인용 모바일 단말의 경우 WLAN(WIFI), WiBro 등 인터넷 커넥티비티를 위한 통신인터페이스를 넘어 USB, 블루투스(Bluetooth), IrDA, CDMA 등 추가적인 각종 인터페이스까지 동시에 활용할 수 있는 단말로 발전하고 있다. 이에 따라 이종망간에 위협요소가 전파되기도 하고 크로스 서비스 공격(cross-service attack), 과금 공격, 유사 DoS 공격을 이용한 배터리 소모 공격 등 새로운 형태의 보안 위협이 생성되기도 하였다[2],[3].

이처럼 다양성과 연결지향성을 바탕으로 서비스 중심의 유비쿼터스 세상으로 미래가 점쳐지는 가운데 이를 위한 보안 기술 역시 보안 요소기술을 바탕으로 한 “통합 보안 서비스”의 관점에서 접근해야 한다는 견해가 설득력을 얻고 있다. 과거 보안 기술의 역사에 중요한 역할을 수행해 온 요소기술을 살펴보면 암호화 인프라, 인증인가 체계 등 대비형 보안 기술과 더불어 보안 위협이나 공격에 직접 적극적으로 대응할 수 있는 침해방지 기술이 발전하여 왔다. <표 1>에서 보는 바와 같이 네트워크 인프라의 관점에서 방화벽(firewall), 침입탐지시스템(IDS), 침입차단시스템(IPS) 등의 기술이 개발되었고, 호스트(장치, 단말)의 관점에서 보안 운영체제(Secure OS), 바이러스 스캐닝(virus scanning 혹은 anti-virus), 개인정보유출차단 등의 기술이 등장했다. 이 중 보안 운영체제 기술은 기존의 보안 제품이 막지 못하거나 탐지하지 못하는 공격을 운영체제의 커널 수준에서 효과적이고 근본적으로 탐지, 차단할 수 있고 내부 사용자의 공격, 실수 등의 상황에도 대응할 수 있는 강점을 가지고 있다. 호스트기반 침입탐지시스템, 바이러스 스캐닝 등의 기술은 기술의 활용 영역이 다를지라도 기본적으로 시그니처(signature) 매칭을 통하여 기존에 알려진 위협요소를 찾아내는 방식에 치중하고 있다. 이러한 방식은 이미 널리 알려져 데이터베이스에 추가되어 있는 위협상황(ex. Internet Worm)이 아니면 탐지할 수 없고, 스캐닝 성능의 제약으로 인해 실시간으로 위협을 탐지, 차단하는 데 한계가 있다. 아울러 애플리케이션과 서비스가 다양화되고 특히 오픈플랫폼 트렌드 속에서 개별 애플리케이션의 신뢰성을 확보하기가 점차 어

〈표 1〉 정보보호 기술분류와 보안 운영체제 기술영역

정보보호 기반기술	- 암호화(Encryption), 키 관리 - 전자서명 - 인증/인가
네트워크 보호기술	- IDS, IPS - Firewall - 가상사설망(VPN)
시스템 보호기술	- 보안 운영체제(Secure OS) - Anti-Virus - Application Securities

려워져 응용 계층의 보안 기술이 한계에 입각하고 있어 보다 근본적인 계층의 보안 해결책이 필요한 실정이다.

본 고에서는 이러한 측면에서 가장 근본적인 지점인 시스템 소프트웨어, 즉 운영체제의 관점에서 보안을 해결하고자 하는 노력의 필요성과 그 기술동향에 대하여 살펴보고, 또한 이 같은 측면에서 모바일 플랫폼 등 앞으로 펼쳐질 새로운 환경에서의 보안 기술에 대해서 예측해보고자 한다.

Ⅱ. 보안 운영체제와 접근제어

접근제어(access control)란 사용자(user) 혹은 프로세스(process) 등 접근 행위의 주체(subject)가 파일, 디렉토리, I/O 인터페이스, 네트워크 리소스 등과 같은 정보 객체(object)에 접근하고자 할 때 각 주체-객체 쌍의 접근 권한, 보안 설정, 보안 상황 등으로 판단하여 접근을 통제하는 것을 말한다. 특히 운영체제의 커널 수준에서 이러한 접근제어 기능을 제공하면 근본적인 보안 수준을 획득하여 애플리케이션이 침해를 당하거나 사용자가 실수를 해도 커널 수준에서 방지할 수 있으므로 보안성을 크게 향상시킬 수 있다.

"운영체제가 보안성을 확보하지 못한다면 어떠한 형태의 보안 노력도 모래 위의 성을 쌓는 결과를 초래하게 될 것이다." (The inevitability of Failure, National Security Agency, USA, 1998)[4]

악성코드, 해킹 등 기존의 보안 침해 사례가 대부분 모든 접근이 가능한 동일한 권한(ex. root, administrator)으로 애플리케이션을 실행했기 때문에 발생한 문제였다. 그러므로 주체의 ID에 따라 혹은 정보의 종류에 따라 세밀하게 권한 부여와 접근을 통제할 수 있다면 이 문제점을 해결하거나 크게 개선할 수 있는 것이다.

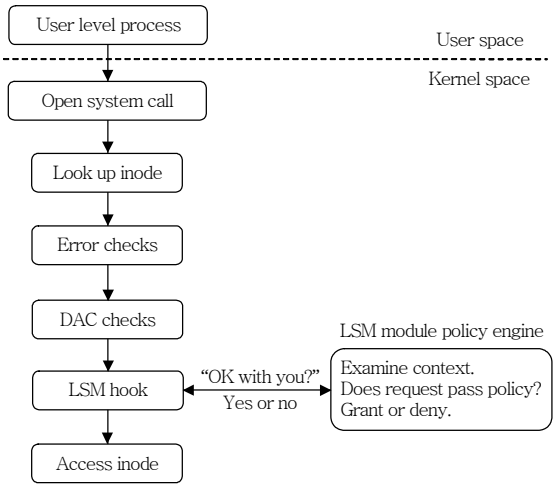
보안성 있는 운영체제를 구축하기 위해 기존에 제시된 접근제어 방법으로는 DAC와 MAC(MLS)가 있다. DAC는 객체의 소유자(user)에 근거를 두어 객체에 대한 접근을 제어하는 방법으로써 대부분의 전통적인 서버 혹은 데스크톱 운영체제 사용자에게 익숙한 가장 기초적인 사용자기반 접근제어 모델이다. 유닉스 시스템의 경우 각 객체에 사용자, 그룹 및 기타에 대한 퍼미션(permission)을 할당하여 그 규칙에 따라 해당 객체에 대한 접근을 결정하며 이 규칙의 설정권 역시 소유자에게 귀속된다. 그러나 DAC에서는 사용자가 실행하는 모든 프로그램이 그 사용자가 가진 권한과 동일한 권한을 가지므로 보안 수준의 정밀도가 매우 떨어진다. 즉 예를 들어 사용자가 자신도 모르는 사이 악성코드를 실행하게 될 경우 그 코드가 포함된 프로세스가 소유자와 동일한 권한을 가지므로 그 사용자가 실행하고 있는 모든 다른 프로세스와 동일한 권한으로 자원에 접근하게 되어 침해를 피할 수 없게 된다. 특히 많은 시스템에서 administrator, normal user 두 개의 사용자 권한을 기반으로 접근제어를 수행하거나 심한 경우 사용자 편의를 극대화하기 위하여 항상 최고권한자(ex. administrator)로 실행하는 경우가 있는데 이런 경우 보안성을 거의 기대할 수 없게 된다. DAC는 다수의 사용자가 하나의 서버 시스템에 동시에 접속하여 사용할 때 파일 등 각종 자원(객체)에 대하여 사용자별 소유권을 명확히 해야 하는 경우에 적합한 접근제어 방식이다. 따라서 보안성을 유지해야 하는 다양한 단말을 안전하게 보호할 수 있는 완전한 보안기술이라 보기 어렵다.

이러한 문제점을 해결하기 위하여 MLS 방법이 등장하게 되었다. MLS는 객체(파일 등 정보자원)의 비밀성(security level)과 주체(user, process)의 권한(clearance)에 근거하여 주체의 객체에 대한 접근을 제한하는 강제적 접근제어(mandatory access control) 방법이다. 그러나 MLS는 객체의 비밀성과 주체의 권한을 일일이 설정해줘야 한다는 특수성 때문에 일반적인 사용에 적합하지 않은 단점이 있다. 특히 역사적으로 미국 정부나 군 기관의 접근제어 정책 요구사항에 부합하기 위해 설계된 방식으로 일반 단말 보호를 위한 보안기술로 그대로 쓰이기엔 많은 문제가 있다.

1. 리눅스 운영체제 접근제어

과거의 DAC, MAC 방법의 장점을 취하고 단점을 보완하여 보다 일반화된 쓰임에 적합하도록 설계된 운영체제 접근제어 보안 기술 중에서 가장 성공적으로 적용되고 있는 구현물은 리눅스 운영체제의 SELinux이다[5]. SELinux는 NSA에서 수행한 보안 OS 프로젝트의 MAC 구현본인 flask를 리눅스 OS에 적용한 것이다. SELinux는 리눅스 커널에서 기본적으로 제공하는 LSM 프레임워크를 통하여 Linux 커널 코드의 전면적인 수정 없이 추가적인 보안 기능의 형태로 구현되었다[6].

LSM은 파일 등 자원에 접근할 때 사용되는 각종 시스템콜의 적정한 지점을 정하여, 이 지점에서 추가하고자 하는 기능을 수행하는 코드로 콜백(call back)이 일어날 수 있도록 하는 후킹 포인트를 미리 제공해 놓은 커널 프레임워크이다. (그림 1)에서 보는 바와 같이 응용프로그램이 자원에 접근하는 시스템콜에 대해서 DAC check 등 기본적인 접근 통제 결정을 한 뒤 LSM을 통하여 접근을 통제하는 부가적인 로직으로 프로시저를 넘겨주는 구조이다. 리눅스 커널이 제공하는 이 구조를 활용하여 접근제어 등 강력한 보안기능을 커널 코드의 수정 없이 loadable한 커널모듈의 형태로 구현할 수 있다. 이 LSM은 2001년 NSA에서 SELinux를 발표했을 때 이와



(그림 1) Linux Security Module[6]

같은 추가 보안 기능을 지원하기 위한 커널 프레임워크를 리눅스 커널측에서 제공하기로 결정하게 되면서 포함된 것으로 알려져 있다. 즉 커널 코드가 보안 기능이 활용할 수 있는 기초 골격을 제공하고 SELinux와 같은 보안 기능은 이 구조를 활용하는 구조이다.

SELinux[5]는 Domain-Type Enforcement, Role-Based Access Control, Multi-Level Security와 같은 강제적 접근제어(MAC) 모델들을 모두 지원하는 일반화된 모델의 구현으로써 fedora 등 리눅스 배포판에 포함되어 제공된다. SELinux는 보안정책로직(security policy logic)과 적용모듈(enforcement module)을 분명하게 구분해 놓았는데, 이는 다양한 보안 정책들을 유연하게 지원하기 위함이다. 보안정책로직에 적용할 수 있는 접근제어모델은 TE, RBAC, MLS 등 기존에 제안된 모델들을 다양하게 선택할 수 있다. 이 접근제어모델들은 모두 사용자, 프로세스 등 주체(subject)가 파일 등 정보 객체에 접근하는 것을 어떻게 허용할지에 관한 관계를 정적(static)으로 정책(policy)을 구축해 놓고 그 정책을 바탕으로 접근제어 판단을 적용(enforcement)하는 방식을 취한다. 이렇게 함으로써 사용자가 원하는 접근제어 모델을 그 모델에 맞는 정책 설정을 적합하게 구축해 놓았을 때 그 설정의 보호를 받는 객체가 보안 위협상황으로부터 잘 보호받을 수 있다.

SELinux는 여러 보안기능을 빠짐없이 제공하기 위한 일반화된 설계로써 중요한 의미가 있지만 이에 따른 설정의 복잡도가 실제 사용시 큰 단점으로 작용하고 있다. 즉 접근제어를 수행하기 위해 기설정되어 있어야 하는 정책을 표현하기가 매우 복잡하고 각각의 주체와 이로부터 보호받을 객체에 대해 미리 정책을 세밀하게 설정해놓지 않으면 접근제어의 보호를 전혀 받을 수 없는 것이다. 또한 SELinux의 디폴트 설정에 의해 정상적인 작업이 제한을 받는 등 사용자 편의성이 떨어져 SELinux 기능 자체를 해제(disable)하고 사용하는 경우가 많다. 즉 보안설정의 다양성을 지원하나 지나치게 일반화된 설계로 인하여 복잡한 세부 설정 사항들을 관리자(보안 사용자)

(그림 2) Policy Management Tools(fedora Core)

가 떠맡게 되고 이를 실시간으로 변하는 개개인의 특정 보안 상황에 맞게 잘 활용하기가 매우 어렵다.

최근의 리눅스 배포판(ex. fedora)에는 정책설정 파일(policy file)을 직접 수동으로 작성할 필요 없이 자동으로 생성해주는 SELinux Policy Generation Druid를 제공한다((그림 2) 참조). 이러한 GUI tool을 이용하여 정책 설정의 편의를 추구할 수 있지만 기본적으로 통합된 정책 설정 환경을 제공하는 것이 아니라 개별 정책 설정파일 작성을 GUI를 통해서 할 수 있도록 도와주는 정도에 불과하다. 그러나 SELinux를 포함하는 리눅스의 LSM 아키텍처는 향후 리눅스기반 플랫폼의 가장 강력한 접근제어 보안 기능으로써 중요한 기반을 차지할 것으로 예상된다.

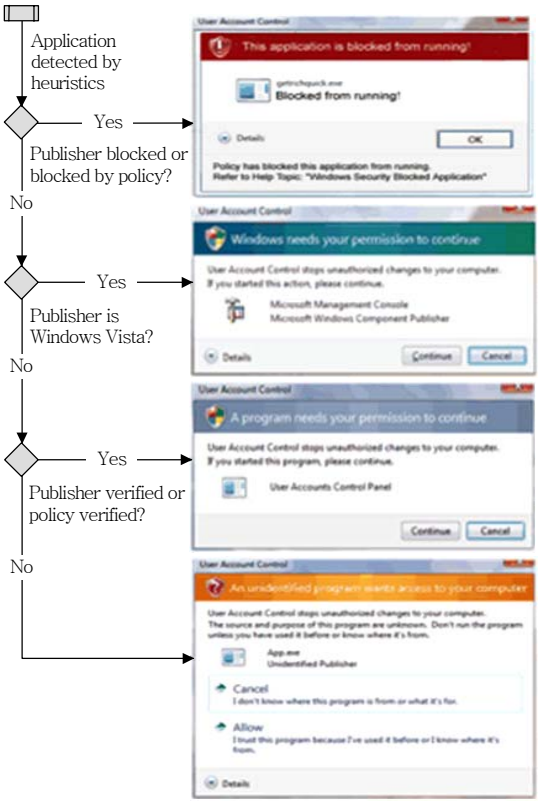
2. 윈도 운영체제 보안

마이크로소프트는 소스코드에 대한 비공개 정책, 보안 취약점 발견시 빠른 업데이트 등 기술적인 해법보다 정책적인 해법을 앞세워 보안 문제를 해결해 왔다. 또한 anti-virus 등 애플리케이션 수준의 보안 제품으로 그 역할을 미루어 온 것도 사실이다. 그러나 최근 들어 운영체제가 근본적으로 지원해야 하는 보안 기능(ex. 계정 관리)에 문제점이 있다는 것에 인식을 하고 이와 관련된 기능을 운영체제에서 제공하기에 이르렀다.

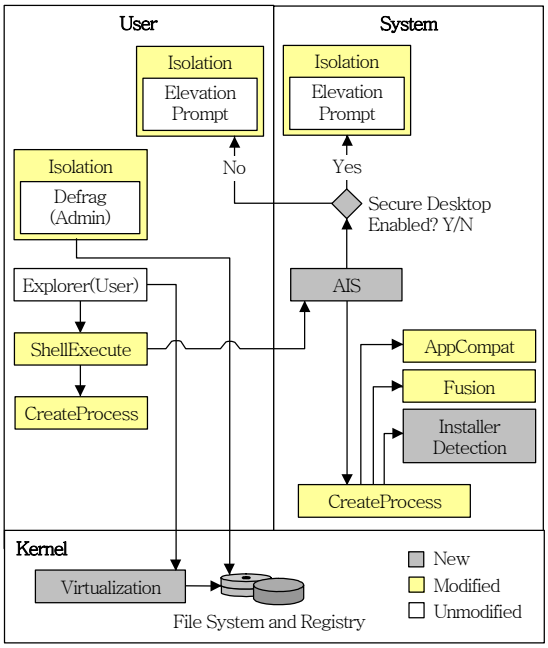
마이크로소프트는 윈도비스타(Windows VISTA)에서부터 사용자 계정 컨트롤(UAC)이라는 기능을 제공한다[7],[8]. Windows XP, Windows 2000, 2003 등 대부분의 마이크로소프트 운영체제의 가장 높은 우선순위의 보안 권고사항은 "최소권한 사용자 계정(LUA)"을 사용하라는 것이었다. 과거 바이러스, 스파이웨어 등 악성코드(malware)가 설치되어 시스템을 장악하게 되는 패턴을 분석한 결과 대부분이 이들이 관리자 권한으로 실행되었다는 것이 밝혀졌기 때문이다. 사용자가 가장 편하게 시스템을 사용할 수 있는 방법이 administrator 계정으로 모든 프로그램을 사용하는 것이고, 그 편리함만큼 사용자도 모르는 사이에 악의적인 코드들이 관리자 권한으로 실행되고 있는 것이다. 반대로 권고사항대로 최소권한의 사용자 계정으로 로그온 했을 경우 정상적인 프로그램을 실행하고 있는 중에도 권한 부족으로 불편함을 겪을 수 있기 때문에 사용자는 최소권한 사용을 경험적으로 꺼리게 된다. UAC는 이러한 사항을 고려한 일종의 절충 기법으로, 평상시에는 표준 사용자 권한(users)으로 프로그램을 실행하고 필요시에만 관리자 권한을 부여 받는다. 단, 이 경우 확인 절차가 필요하게 되므로 데스크톱 팝업 창(elevation prompt)을 띄우는 방법을 사용한다((그림 3) 참조). 이와 같은 과정이 필요한 관리자 권한 작업은 운영체제에 미리 정의되어 있고 다음과 같은 작업들이 이에 해당한다.

- %Systemdrive%에 대한 핸들링
- %Systemroot%(Windows), Program Files 폴더에 대한 쓰기 작업
- 레지스트리 중 HKEY_LOCAL_MACHINE에 대한 쓰기 작업
- 응용프로그램 설치, 삭제
- 장치 드라이버 설치
- Internet Explorer에 대한 각종 설정 변경
- 액티브엑스 컨트롤 설치
- Windows 시스템의 각종 설정 변경

이러한 행위(behavior)는 시스템 서비스인 AIS가 모니터하여 처리하는 구조로 되어 있다((그림 4) 참조). 이러한 작업들은 주로 운영체제 자체가 위협



(그림 3) Windows VISTA UAC Elevation Prompt[7]



(그림 4) UAC Architecture[8]

받을 수 있는 상황을 구체화한 것으로, 레지스트리 등 시스템 설정이나 파일들을 보호하거나 악성 소프트웨어의 설치를 방지하기 위함이다. 즉 사용자에 따라 달리 요구사항이 정의되는 개인 중요정보 파일 보호 등의 효과를 기대할 수는 없다. 따라서 이러한 보안 아키텍처를 개인 정보보호, 유출방지 기술로 확장하여 적용한다면 더 강력한 보안 플랫폼을 구축할 수 있을 것으로 기대된다.

3. 보안 커널 모듈

운영체제가 자체적으로 보안기능을 제공하지 않을 경우 보안개발자가 보안 기능을 추가하고자 할 수 있다. 이때 커널의 소스코드를 수정하여 기능을 추가하기가 매우 까다로우므로 보통 LKM 방식을 사용한다. 이 방식은 리눅스의 LSM과 같이 기존의 커널을 그대로 두고 추가적으로 커널 보안 기능을 구현할 수 있는 형태를 말한다. 주로 "보안 OS"라 불리는 보안 전문업체가 개발하는 독립적인 보안소프트웨어 제품이 이에 해당한다. 국내 제품으로 티맥스소프트(SysKeeper), 시큐브(Secuve TOS), 레드게이트(Red Castle), 안랩시큐브레인(Hizard SecureOS), 티에스온넷(REDOWL), CA(e-Trust) 등이 있으며 국외 제품으로 IBM(Brocade Advanced Security), Digital Equipment, HP Secure OS, Silicon Graphics(Trusted IRIX), Unisys(Secure OS 2200 Systems) 등이 있다. 독립적인 소프트웨어의 형태로 제공하는 제품은 운영체제 기술 등 기반기술의 부재로 호환성, 적용성 등에 제한이 있으며 주로 OS, 시스템 구축을 할 때 필요에 의해 시스템 구축의 차원에서 제공되므로 보안 기능의 독립화, 특성화가 어려운 단점이 있다. 또한 대부분이 서버 제품군을 위한 보안 기술로써 향후 다양한 단말에 직접적으로 적용 가능한 기술은 아니다. 이 같은 기술들은 운영체제에 독립적으로 제공할 수 있는 보안 구조로 의미를 둬야 할 것으로 보이며 향후 다양한 단말과 운영체제(OS)에 적용할 수 있는 보안 서비스 기술로 중요한 기반이 될 수 있을 것으로 보인다.

Ⅲ. 다양한 모바일 단말과 보안

스마트폰을 필두로 하여 PMP, PDA, 휴대용 게임기, UMPC, 인터넷단말, 텔레매틱스 단말 등 수많은 모바일 단말이 그 활동반경을 넓혀가고 있다. 각각의 단말들은 고유한 영역과 그 특수성을 지니고 있으면서도 동시에 일반화되고 복합화된 기능을 구비해가고 있다는 공통점이 있다. 스마트폰의 경우 이미 오래 전에 단순한 휴대전화의 개념에서 벗어나 디지털 카메라, MP3 플레이어, email, DMB, 전자수첩, 게임기, 인터넷 브라우징 등 무수한 기능을 제공하는 내 손안의 PC가 되어가고 있다. 또한 데스크톱 환경에서 출발한 노트북 컴퓨터, UMPC 등의 기기도 소형화, 복합화 추세로 이동성, 편의성에 있어 스마트폰, PDA 등의 기기와의 격차를 줄여나가고 있을 뿐만 아니라, 기존의 카메라와 같은 특수한 장치도 점차 네트워크에 연결되는 하나의 유비쿼터스 단말이 될 것이라는 견해가 지배적이다.

이러한 모바일 단말의 기반이 되는 운영체제를 비롯한 소프트웨어 체계를 일반적으로 모바일 플랫폼이라 부른다. 모바일 플랫폼으로 Linux, Windows 등 범용 기반 OS를 비롯하여 WIPI, BREW, J2ME, Symbian, Windows Mobile 등 다양한 솔루션이 있으며, 최근 애플의 iPhone과 구글의 Android가 이 대열에 합류하여 주목을 받고 있다[9]-[11]. 복합적인 기능을 추구하는 시장의 요구에 더불어 하드웨어의 성능 향상으로 모바일 플랫폼은 점차 범용 운영체제와 유사한 구조로 발전하여 기존에 펌웨어 수준의 하위 서비스를 벗어나 메모리 프로텍션이나 멀티태스킹과 같은 범용 운영체제 서비스를 제공하게 될 것으로 예측하고 있다[10].

이러한 추세에 따르면 모바일 단말의 보안 수준과 기능 역시 현재의 데스크톱이나 서버시스템에서 요구되는 수준 또는 그 이상이 될 것임은 명확하다. 특히 물리적인 이동성이 크고, 유무선 통신인터페이스가 더 다양해지는 환경을 고려할 때 보안 문제가 데스크톱 환경보다 오히려 더 극대화될 확률이 높다. 이러한 미래 단말들을 보안성의 관점에서 특성화한다면 다음과 같은 특징을 들 수 있다.

- 물리적인 이동이 잦음
- 통신 인터페이스의 종류가 다양
- 주로 단일 사용자가 사용
- 입출력, 디스플레이 등의 제한으로 복잡한 UI 적용불가

이러한 보안 요구사항에 따라 각종 모바일 플랫폼은 자체적인 보안환경을 제공하고 있다. 대표적으로 WIPI의 경우 다음과 같은 정책적, 기술적 대책을 제공한다.

WIPI 플랫폼의 보안 기법은 기본적으로 플랫폼의 각 자원별 정의된 보안 수준별 접근허용 여부와 응용 프로그램의 보안 수준에 대한 정보를 바탕으로 수행된다[12].

- Public: 가장 신뢰할 수 없는 수준, 공용 응용프로그램 등
- CP: 일정 수준 이상 혹은 전폭 신뢰가 가능한 수준
- System: 플랫폼이 응용 프로그램을 완전히 신뢰 가능한 것으로 간주하고 모든 자원에 대한 접근을 자동 허용하는 수준

보안 수준에 대한 자원의 종류에는 최소한 다음 그룹이 포함되어 각 그룹별로 제어 가능해야 한다[12].

- Storage: Private director/Application Shared Directory/System Shared Directory
- Network: Connection Oriented, Datagram, HTTP
- Secured Network Connection: HTTPS
- Serial Device: RS-232C, USB3
- Personal Information: 주소록, 촬영한 사진, 기타 개인 정보

이와 같은 접근 수준과 자원 그룹에 따라 <표 2>와 같은 예제의 접근허용여부표를 작성할 수 있다[12].

이렇게 정의된 플랫폼의 접근인가여부표와 응용프로그램별로 설정된 보안 수준을 통해서 보안 관련

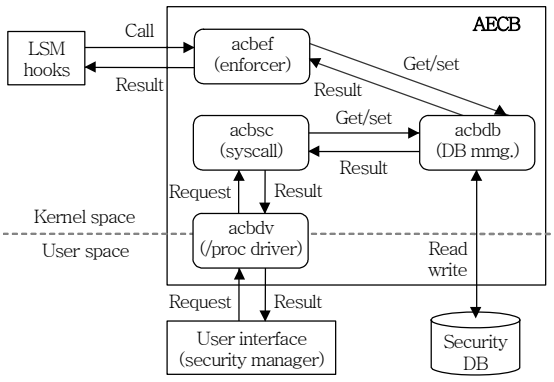
〈표 2〉 WIPI 보안수준별 접근인가표 예

	Graphic	Sound	Personal Info	System Settings
Public	Allow	Allow	Deny	Deny
CP	Allow	Allow	User	Deny
System	Allow	Allow	Allow	Allow

동작을 수행한다. 이와 같은 접근 통제 모델 이외에도 WIPI 표준은 SSL 보안 통신을 위한 API를 제공할 것을 명세하고 있다. 이와 같은 WIPI 등 모바일 플랫폼 표준의 보안 모델은 플랫폼이 기본적으로 제공하는 최소 보안기능으로서 시스템의 일반적인 보안성을 향상시킬 수 있으나, 사용자별 요구사항에 맞는 세밀한 보안 모델이라 보기는 어렵다. 예를 들어 특정 응용프로그램에 실수로 과도한 권한을 부여했을 경우 그 이후의 보안 문제에 대해 추적할 수 없고, 이를 위해서는 일일이 수동으로 보안 요구사항을 반영해야 하는 불편함이 있다.

한국전자통신연구원 정보보호연구본부에서는 이와 같은 기술요구 추세에 따라 "복합단말침해방지기술" 영역에서 향후 다양한 단말의 보안 요구사항에 맞는 보안 OS 기술(접근제어)을 개발하고 있다.

복합 단말을 위한 접근제어 기술은 리눅스 기반의 기술로써 데스크톱 환경을 포함하여 각종 미래형 모바일 단말에 적용 가능하다. (그림 5)에서 보는 바와 같이 리눅스 LSM architecture를 이용하여 커널 코드의 수정 없이 추가적인 보안 기능으로 구현되었다. RBAC 기반의 접근제어 모델을 통해서 사용자, 프로세스가 파일시스템, 네트워크 인터페이스 등의 정보자원에 접근하는 것을 세밀하게 통제할 수 있으며 이러한 정책들을 유저 인터페이스를 통하여 실시간으로 변경, 적용 가능하다. 향후 복합단말의 제한된 환경을 고려하여 사용자 편의성에 더욱 부합하면서 기존의 플랫폼보다 세밀하고 강력한 보안 기능을 제공할 수 있는 보안 OS 기술을 개발중이며 향후 Linux, WinCE 등 다양한 플랫폼에 적용 가능할 것으로 기대된다.



(그림 5) 복합단말침해방지기술 접근제어 블록

Ⅳ. 결언

정보 기술과 역사를 함께하는 정보보호 기술은 유비쿼터스 시대를 맞아 그 전환점에 와있다. 언제 어디서나 컴퓨팅 환경을 접할 수 있는 만큼 언제 어디서나 보안 이슈가 부각되는 것이다. 특히 많은 종류의 복합 단말이 그 활동영역을 넓혀가면서 단말의 시스템소프트웨어 보안이 문제의 핵심으로 떠오르고 있다. 응용계층 보안, 안티바이러스 등 시그니처 스캐닝 기반의 보안 기술은 근본적이지 않을 뿐만 아니라 그 역할에 한계가 있는 것이다. 가장 근본적인 시스템소프트웨어인 운영체제 수준에서 제공하는 보안 기능은 가장 강력하고 확실한 대책이다. 과거 펌웨어 수준의 모바일 플랫폼이 이미 범용 운영체제 수준의 정보처리기반을 제공하고 있는 바, 본고에서는 데스크톱용 보안 모델인 SELinux, Windows 최신 보안 기술 등이 앞으로 도래할 최신 모

● 용 어 해 설 ●

보안 운영체제: 운영체제의 커널에 부가적인 보안기능을 추가한 운영체제를 말한다. 주로 사용자나 응용계층이 파일 등 자원에 접근하는 것을 정밀하게 통제할 수 있는 접근제어 기능이 이에 해당한다. 해킹, 바이러스, 응용프로그램 취약성, 사용자 실수 등 각종 보안문제에 근본적으로 대응할 수 있는 장점이 있다.

모바일 플랫폼: 모바일 단말이 다양한 응용프로그램을 지원할 수 있도록 해주는 시스템소프트웨어 계층을 말한다. 주로 운영체제부터 응용프로그램에 직접 서비스를 제공하는 계층까지를 의미한다.

바일 플랫폼 보안 기술의 주춧돌로 작용할 가능성이 매우 높다고 판단한다. 앞서 살펴본 바와 같이 WIPI 등 전통적인 모바일플랫폼 표준이 데스크톱 접근제어 보안모델과 유사한 보안 정책을 이미 명세하고 있다.

이 같은 기술 요구사항의 추세로 볼 때 향후 더 세밀한 통제가 가능하고 사용자 편의성에 부합하는 보안 요구사항이 있을 것이라는 것은 자명하다. 또한 시스템 자체를 지키기 위한 일반적인 보안 정책을 넘어, 사용자 개개의 요구사항에 맞는 보안 필요성을 만족시킬 수 있는 운영체제 수준의 정교한 보안 서비스가 필수 불가결해질 것이다.

약 어 정 리

AIS	Application Information System
AJAX	Asynchronous JavaScript and XML
DAC	Discretionary Access Controls
IDS	Intrusion Detection System
IPS	Intrusion Prevention System
LKM	Loadable Kernel Module
LSM	Linux Security Module
LUA	Least-privileged User Account
MAC	Mandatory Access Control
MLS	Multi-Level Security
NSA	National Security Agency
RBAC	Role-Based Access Control
SELinux	Secure-Enhanced Linux
SOA	Service Oriented Architecture
TE	Type Enforcement
UAC	User Account Control
WIPI	Wireless Internet Platform for Interoperability

참 고 문 헌

[1] Cell phone virus tries leaping to PCs by Dawn Kawamoto Staff Writer, CNET News.com, Sep. 22, 2005, 10:02 AM PDT.

[2] Collin Mulliner, Giovanni Vigna, David Dagon, and Wenke Lee, "Using Labeling to Prevent Cross-Service Attacks Against Smart Phones," DIMVA 2006, LNCS 4064, 2006, pp.91-108.

[3] Timothy K. Buennemeyer, Michael Gora, Randy C. Marchany, and Joseph G. Tront, "Battery Exhaustion Attack Detection with Small Handheld Mobile Computers," Portable Information Devices, PORTABLE 07. IEEE, 2007.

[4] Peter A. Loscocco, Stephen D. Smalley, Patrick A. Muckelbauer, Ruth C. Taylor, S. Jeff Turner, and John F. Farrell, National Security Agency, "The Inevitability of Failure: The Flawed Assumption of Security in Modern Computing Environments," *the Proc. of the 21st National Information Systems Security Conference*, Oct. 1998, pp.303-314.

[5] Peter A. Los cocco, NSA, and Stephen D. Smalley, NAI Labs, "Meeting Critical Security Objectives with Security-Enhanced Linux," *the Proc. of the 2001 Ottawa Linux Symp.*, 2001.

[6] Chris Wright, Crispin Cowan, Stephen Smalley, James Morris, and Greg Kroah-Hartman, "Linux Security Modules: General Security Support for the Linux Kernel," oasis, p.213, Foundations of Intrusion Tolerant Systems(OASIS'03), 2003.

[7] Understanding and Configuring User Account Control in Windows Vista, http://technet2.microsoft.com/WindowsVista/en/library/00d04415-2b2f-422c-b70e-b18ff918c2811033.mspx?mfr=true

[8] Jennifer Allen, "Windows Vista Application Development Requirements for User Account Control Compatibility," Microsoft Whitepaper, June 2007.

[9] 오승희, 김기영, "리눅스 기반의 휴대단말 운영체제 동향 분석," 전자통신동향분석, 제23권 제3호, 2008. 6., pp.152-162.

[10] 윤민홍, 김선자, 리눅스모바일단말SW연구팀, "글로벌 모바일 단말 소프트웨어 플랫폼 동향," 전자통신동향분석, 제23권 제1호, 2008. 2., pp.144-153.

[11] 전종홍, 이승윤, "모바일 웹 2.0과 모바일OK 표준화 동향," 전자통신동향분석, 제22권 제6호, 2007. 12., pp.84-97.

[12] 한국 무선 인터넷 표준화 포럼, "모바일 표준 플랫폼 규격 2.0.1(Wireless Internet Platform for Interoperability 2.0.1)," 2004. 9., http://www.wipi.or.kr/