한국산학기술학회논문지
Vol. 11, No. 12 pp. 5107-5112, 2010

가상환경을 위한 제어모델과 인증모델

유진호[1*], 이병철[2]
[1]백석대학교 정보통신학부, [2](주)센트루소프트웨어

The Control Model and The Authentication Model for a Virtual Computing Environment

Jinho Yoo[1*] and Byoungchul Lee[2]
[1]**Division of Information and Communication, Baekseok University**
[2]**Sentrue Software Inc.**

요 약 본 논문은 가상환경 상에서 어떤 작업을 할 때 가상환경 상의 자원을 관리를 제어하는 플랫폼 접근에 관한 것이다. 또한 이러한 플랫폼에서 제어권한을 위한 인증을 어떻게 다룰 것인가에 관한 연구이다. 하드웨어의 발전에 따라 하드웨어 환경의 현격한 발전은 사람이 요구하는 컴퓨팅을 충분히 처리하고 남는 단계에 있다. 이러한 경우 남는 컴퓨팅 자원들을 효율적으로 처리하기 위해 가상환경이 연구되어 플랫폼 제작이 활발히 연구되고 있다. 이러한 가상환경은 기존의 실제 환경과는 기능상으로는 거의 유사하지만 실제 구성과정은 많이 다르다. 이에 실제 컴퓨팅 환경에서 일어날 수 있는 문제 뿐만아니라 가상환경은 또 다른 문제를 가진다고 할 수 있다. 이에 본 논문에서는 이러한 실제시스템 환경과 다른 가상환경의 제어플랫폼과 인증방법을 제시할 것이다.

Abstract This paper is related to the access to the control platform who manages their resources on the virtual environment. The computing performance of hardware supersedes the computing performance human demands. In this case, making the platform on the virtual computing environment has been developed for the usage of computing resource's residues which is available. This virtual computing environment is quite similar with the real environment in aspect of some functions but is quite different with that in aspect of the real configurations. This paper found that the virtual computing environment has another problems besides the problems the real computer might be happening. This paper found that the virtual environment needs the proper control platform. This paper will suggest a control platform for managing the virtual computing environment properly.

Key Words : USB, Authentication, Yirtualization

1. 서론

하드웨어 환경의 급속한 발전에 따라 컴퓨터 시스템의 하드웨어 구성은 고성능화 되었고 범용 소프트웨어 수행을 위해 요구되는 컴퓨팅 능력을 충분히 처리할 수 있다. 이러한 이유로 컴퓨터 시스템 상에 하드웨어를 소프트웨어로 가상화하여 구현하게 되었다. 한 컴퓨터 하드웨어 상에 가상화된 여러 소프트웨어 머신을 탑재할 수 있게 된 것이다. 컴퓨터 주요기능수행에 있어서는 전혀 지장없이 하나의 컴퓨터 상에 소프트웨어 머신들을 탑재할 수가 있다. 소프트웨어 제작에 의한 가상머신을 통해 하드웨어가 분리된 것처럼 컴퓨팅 요소로 채워지게 되었다. 이와같은 기술을 통해 서버가 가상화 되고 개인의 컴퓨팅을 위해 컴퓨팅을 인가하는 기술적인 지원이 가능하게 되었다. 하드웨어 환경이 소프트웨어의 발전보다 훨씬 앞서는 형태는 결국 다시금 고성능 서버로부터 개인에게 필요에 따라 컴퓨팅을 빌려주는 형태로 발전하게 되었다. 즉 개인이 필요에 의해 컴퓨팅을 요구하면 그 요구에 맞는 소프트웨어 컴퓨터인 가상화 컴퓨터 머신을 구성하여 사용가능하도록 한다. 컴퓨팅 환경의 변화와 유비쿼터스

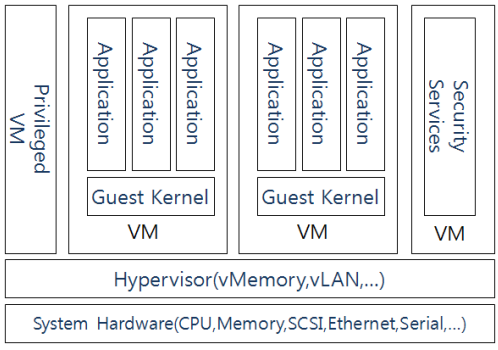
*교신저자 : 유진호(yoojh@bu.ac.kr)
접수일 10년 11월 17일 수정일 10년 12월 03일 게재확정일 10년 12월 17일

환경의 결합에 의해 어느 곳에서나 컴퓨팅 서버로부터 자원을 할당받는 컴퓨팅형태가 요구되었다. 가상 컴퓨팅 환경을 개발하는 상업적인 모델도 기술적으로 완성단계이며 기존의 하드웨어 역할을 완전히 담당함과 동시에 다분화되고 편의를 위한 여러 변형들이 현재 시도되고 있다. 가상화된 컴퓨팅 환경은 기능적으로 하드웨어 환경을 완벽하게 대치하지만 실제 시스템과는 다르다. 가상환경은 하드웨어화된 소프트웨어 머신이 하나의 프로그램으로 다른 프로그램에 의해 접근이 가능할 수가 있다. 하드웨어 컴퓨팅 환경과의 차이는 보안이나 제어에 더 각별히 주의해야 하는 차이를 가지고 있다. 기존의 제어모델에 추가되는 요구사항들이 생기게 되었다. 이러한 기술의 장점은 컴퓨팅 장치가 소프트웨어로 구성되어 있기 때문에 하드웨어 리소스가 허락하는 한 무한히 컴퓨팅 장치가 생성될 수 있고 소프트웨어 조작에 의해 컴퓨팅 장치를 추가, 제거, 수정 그리고 이동시킬 수 있다. 이러한 장점이 있지만 하드웨어이기 때문에 접근하지 못하던 하드웨어 컴퓨팅 장치가 소프트웨어로 구현되어 수정이 가능한 점은 제어구조 상 취약할 수 있다. 이에 많은 시장조사 기관에서는 가상환경으로 발전함에 따라 가지는 문제점에 대한 대비를 해야 한다는 지적들이 논의되고 있다. 시스템 플랫폼의 변화는 제어구조의 변화와 새로운 제시를 요구하게 되었다. 이에 본 논문에서는 vmware 가상화 시스템을 기반으로 하는 제어구조를 제안하고 인증을 위한 효율적인 방법을 제시하도록 한다.

2. 연구배경

2.1 가상화 컴퓨팅 환경

본 절에서는 제안하는 제어구조의 배경이 되는 가상컴퓨팅 환경에 대해서 설명하도록 한다. 가상화시스템은 하이퍼바이저라는 특수한 프로그램 상에 각각의 운영체제가 독립적인 가상 하드웨어 위에서 수행되는 구조를 갖고 있다. 그림 1과 같이 시스템 하드웨어 위에 하이퍼바이저라는 가상화 프로그램이 올라가 있다. 그리고 시스템 생성요구에 의해 가상머신을 할당하여 그 위에 운영체제 등의 소프트웨어를 올릴 수 있게 구성된다. 가상머신 위에 게스트 커널이 올라가고 그 위에 응용프로그램들이 올라가게 된다. 현재는 보안서비스 모듈과 특권화된 서비스를 지원하는 가상머신이 올라가 있다. 가상머신은 실제 시스템에 부착된 입출력 장치로 연결되어 구현되며 실제 시스템처럼 동작한다.



[그림 1] 가상화 시스템 구조

하이퍼바이저 상에 게스트시스템을 설치하면 가상 하드웨어에 대한 드라이버 역할을 수행하는 프로그램들에 의해 가상머신과 하이퍼바이저 간의 미리 약속된 특수한 입출력 포트를 통한 통신이 이루어 진다. 일반적으로 실제 시스템에서는 입출력 포트에 대한 접근권한은 특권명령에 의해서만 수행되어 사용자 모드에서 접근이 되지 않는다. 그러나 가상화 시스템에서는 가상화 시스템의 운영을 위한 필요로 인해 특정한 입출력 포트에 대해 특정 매직 넘버로 접근할 경우 사용자 모드에서도 접근이 가능하다.

2.2 대상 가상화 시스템

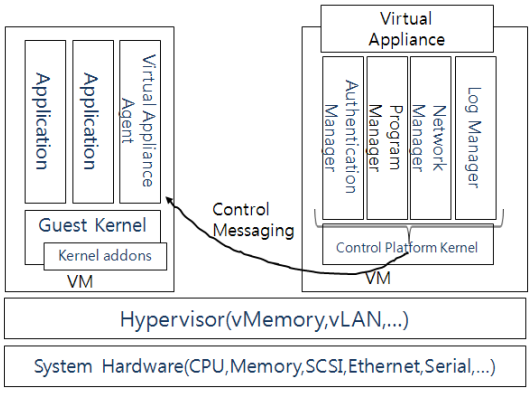
본 논문에서는 논문의 기본적인 수행환경이 가상화 시스템이다. 본 환경의 구성은 vCenter, vSphere, ESX 4.1 등으로 구성된다. ESX 위에 가상머신이 올라간다. ESX는 확장형 하이퍼바이저이다. ESX는 리눅스 운영체제로 구성되어 있다. 리눅스 위에 커널모듈의 형태로 가상머신 하이퍼바이저를 구현한다. 리눅스를 하드웨어 환경을 위한 운영체제로 사용하고 그 위에 커널모듈 형태로 하이퍼바이저를 구현하여 사용하며 그 위에 가상머신이 올라가고 가상머신은 하드웨어와 같은 구성을 가지므로 운영체제를 포함하여 응용구현의 기반이 된다. ESX의 구성에 접근하고 통제하기 위한 인터페이스가 vSphere이다. vSphere를 통해서 ESX 위에 게스트를 올리고 통제할 수 있으며 통계자료에 접근할 수 있다. vSphere상에는 콘솔도 마련되어 있다. ESX는 하나의 하이퍼바이저를 구성하여 여러개의 가상머신을 구성하고 그 위에 게스트 운영체제가 올려져 있다. vCenter는 여러개의 ESX로 구성이 되며 가상머신을 클러스터 형태로 할당하는 것이 가능하다. 가장 상위에 vCenter가 있고 하위에 ESX의 모임이 있고 ESX는 하이퍼바이저 역할을 하여 그 위에 수개의 가상머신이 탑재된다. 이러한 형태는 수개의 ESX 하이퍼바이저와 수개의 vCenter로 구성되어 수많은 소프트웨어

가상머신을 관리하게 된다. 실제 하드웨어의 수가 추가되고 성능이 좋아질수록 관리해야 할 소프트웨어 가상머신의 수가 늘어난다. 구조가 복잡해질수록 중앙에서 제어할 수 있는 제어구조가 요구된다.

3. 제안 가상화 시스템 제어구조

3.1 제안하는 제어구조와 기능

본 논문에서는 가상머신 상의 모든 제어가 가능한 형태의 가상머신 제어구조를 제안한다. 가상머신은 라이브 업데이트, 라이브 마이그레이션 등의 기능을 수행하다 보면 가상머신 시스템 내의 작동에 대해서 기능적인 장치 외에도 제어수단이 필요하다. 이러한 제어수단을 구현할 수 있는 제어구조를 가지는 제어플랫폼은 그림 2에서 보는 것과 같다. 시스템 하드웨어 상에 하이퍼바이저로서 ESX서버가 올라간다. ESX서버 상위 단에는 수개의 가상머신이 탑재될 수 있다[5,6]. 그러한 가상머신 중의 하나로 가상 어플라이언스인 제어구조 가상머신이 올라가게 된다. 가상머신 위의 제어플랫폼 운영체제와 기능모듈들이 탑재된다.



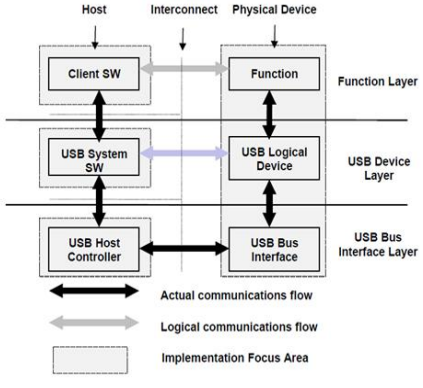
[그림 2] ESX 서버 내의 구조

제어 플랫폼 커널에는 로그 관리자, 네트워크 관리자, 프로그램 관리자, 인증 관리자가 올라가게 되며 이러한 관리자를 통해 ESX 서버 내의 전체 가상머신을 관리하고 예견되지 않은 접근을 차단 보호하게 된다[6]. 프로그램 관리자는 제어 시스템을 구성하는 모듈을 최신버전으로 갱신하고 에이전트나 관리자에 소프트웨어를 배포하는 기능을 한다. 네트워크 관리자는 기본적으로 가상 네트워크 상에서 방화벽의 기능을 수행하며 추가적으로 요구되는 네트워크의 기능을 구현한다. 로그 관리자는 각 컴포넌트로부터 발생한 로그를 수집 저장하고 외부 프로세스로 전달하는 기능을 수행한다. 제어 에이전트는 가상머신에 설치되어 가상머신을 인증하고 인증실패 시 정책을 수행하여 인증된 가상머신이 불법적으로 유출되어 사용되지 못하도록 하는 기능을 수행한다. 제어 커널은 인증된 가상머신의 커널 영역에 적재되어 에이전트에 의해 설정된 정책에 의해 접근 제어를 수행하는 기능을 한다. 추가로 에이전트 설치자는 에이전트가 있어야 할 곳에 에이전트 설치를 관리하는 역할을 한다. 모든 제어 기능 담당 프로그램은 그래픽 사용자 인터페이스를 사용하여 해당 초기값을 설정할 수 있고 구성을 초기화 할 수 있다. 결국 사용자 인터페이스를 통해서 세션 관리, 하이퍼바이저 정보조회, 인증정보 관리, 네트워크 관리, 이벤트 관리, 알람 관리, 사용자 관리, 에이전트와 관리자 갱신 등을 수행한다.

3.2 제안하는 인증모델

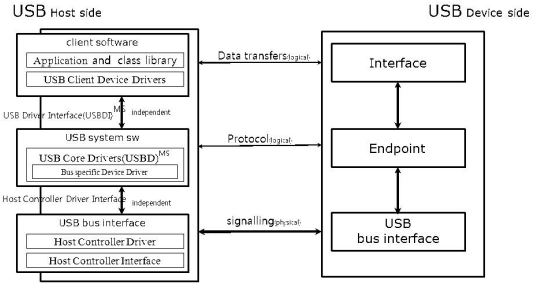
본 장에서는 가상머신에서 인증키를 사용하는 방법에 대해서 논한다. 가상머신 상의 게스트 운영체제에서 접근할 수 있는 형태의 인증을 제공해야 한다. USB장치는 인증키를 저장하기 위한 수단으로 많이 사용되는 장치이다. 즉 USB인증을 선택할 경우 기존의 인증방법을 그대로 사용할 수 있는 장점이 있다. 기존에 잘 마련된 검증된 인증방법을 사용할 수 있다는 뜻이다.

USB는 가장 널리 사용되는 컴퓨터 주변장치 중에 하나이다. 거의 모든 주변장치가 USB 인터페이스를 지원한다. 즉 USB 장치를 통해 입출력장치를 정의할 때 가장 일반적으로 사용되는 것이 될 수 있다는 뜻이다. USB는 물리적 장치 뿐만 아니라 소프트웨어 장치스택을 가지고 있으며 주종관계를 통해 상당히 통제되고 에러없는 사용방법을 제공한다. 그림 3은 USB 전체 스택에 관한 것이다[3].



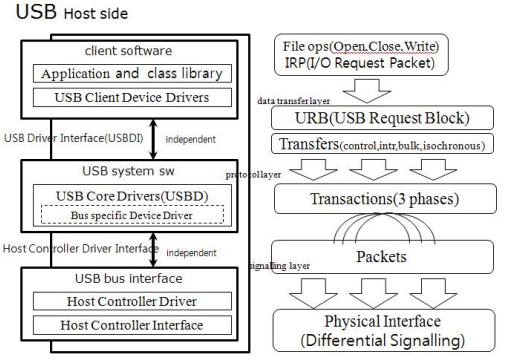
[그림 3] USB stack

그림 3에서 보는 것과 같이 가장 하부에는 호스트 제어기가 USB 버스 인터페이스를 통해서 주종관계로 연결되어 있다[1]. 이를 위해 호스트는 USB 장치를 구동시키는 시스템 소프트웨어와 그 위의 각 장치별 장치구동 소프트웨어로 스택이 구성된다. 이러한 호스트 스택은 여러 개의 USB 장치를 제어할 수 있는 형태로 제공된다. 종속적인 역할을 하는 USB 장치는 소수의 기능단위 구현이 된다. 그림 3의 스택과 같이 구성되는 USB의 계층 중 가장 하위 단인 USB 버스 인터페이스를 소프트웨어로 구현한다. USB의 경우 특성별로 구현하는 것보다 장치자체로 구현하는 것이 유용하다. 이로써 USB 장치의 모든 하드웨어에 대한 접근이 가능해진다. USB하위 단을 소프트웨어로 구현하면 하위 단의 하드웨어 접근을 소프트웨어를 통해 모사가 가능하다. USB 장치의 하부에서 소프트웨어 구현이 되므로 하부 단에 대한 접근이 정상적으로 인가되는 것이다. USB 장치 중 하나인 디스크의 경우 파일시스템 개념의 디렉토리를 보여주는 것이 아니라 USB 장치 자체를 접근할 수 있다. USB 디스크에 파일시스템 함수로 접근하는 것은 하부가 파일시스템 함수로 이루어지고 하부 파일시스템 아래로 접근할 수 없다. USB장치의 특정한 위치에 임의로 접근하는 것을 구현하려면, USB를 장치수준으로 접근해야 해결할 수 있다. 파일시스템은 파티션 위에 올라가는 소프트웨어 구조이다. 디스크 장치는 장치 위에 512바이트의 MBR(Master Boot Record)와 최대 4개의 파티션으로 구성된다. 파티션은 다시 부트 레코드 블럭, 슈퍼 블럭, 아이노드 리스트 그리고 데이터 블럭으로 구성된다. 파일시스템이 파티션 위에 올라간다 해서 파일시스템을 통해 파티션의 모든 정보에 접근할 수 있는 것이 아니다.



[그림 4] USB 호스트와 장치 프로토콜 스택

파일시스템에서 인가한 함수에 의해서만 파티션에 접근할 수 있다. 즉 파일시스템의 파일접근 함수만을 이용할 수 있다. 장치수준에서 접근하기 위해서는 USB 스택의 하위 단을 이뮬레이션해야 한다. 그림 4는 USB의 상세한 스택구성을 보여준다[3]. USB는 원래 그림 4와 같은 구성을 가지고 있고, 가상머신에서 USB 인증을 위한 장치 이뮬레이션은 그림 4에서 호스트 스택의 가장 하위 단인 하드웨어 바로 위에 위치한 소프트웨어에서 아랫단 하드웨어 기능을 대신 처리하게 된다.



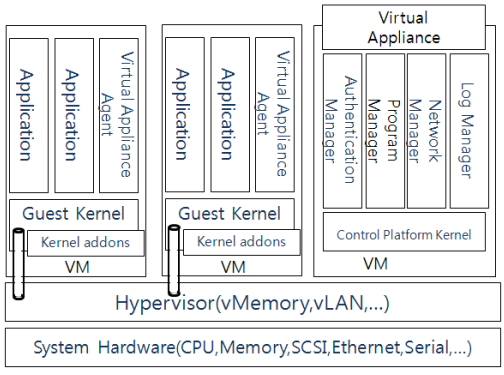
[그림 5] 호스트 제어기 상세

하드웨어는 전송단에 해당되며 상위에서 전달된 데이터를 USB장치로 전달하게 된다. USB호스트의 하위 하드웨어를 소프트웨어로 처리하여 USB장치의 연결을 소프트웨어적으로 임의로 변경할 수가 있게 된다. 이러한 구현을 위해서는 USB호스트의 하위 소프트웨어 부분인 호스트 제어기 드라이버를 새로 작성해 주어야 한다. 호스트 제어기의 상세내역은 그림 5과 같다. 그림 5는 USB 호스트의 상세한 데이터 이동 내용을 보여준다. 가장 상위에서는 파일을 접근하는 것처럼 접근을 한다. 아래로 내려오면서 트랜잭션들로 나누어지고 다시 조작하여 패킷을 만든다. 이러한 자료구조 지원을 위한 클라이언트 소프트웨어는 USB 시스템 소프트웨어에 연결이 되고 그 아래는 호스트 제어기로 연결되기 위한 호스트 제어기 드리이버로 연결된다. 가상 USB 장치를 연결하기 위해서 본 연구는 호스트 제어기 드라이버를 다시 프로그램하여 가상 USB 디스크를 연결할 수 있도록 한다. 즉 가상화 USB 장치를 통한 인증을 위해서는 USB 호스트 제어기와 USB 디스크 장치를 소프트웨어로 구현한다. 새롭게 구성된 USB호스트 제어기는 커널에 독립된 USB 호스트 제어기로서 등록 절차를 수행한다. 독립적으로 구성된 USB 호스트 제어기에 file-backed storage를 가상의 USB 디스크로 인식되는 파일을 설정할 수 있다. vmware의 경우 하이퍼바이저 모듈에서 파일시스템 접근을 지원하지 않는다. 본 연구에서는 USB 디스크 인식을 위한 파일 제공을 커널 모듈에 정적으로 고정하여 제공하도록 한다. 이를 통해 호스트 제어기로 연결되는 커널 모듈의

정적 데이타를 USB 디스크로 보이게 하여 인증을 위한 가상 USB 디스크로 사용한다. USB 저장 장치는 커널에서는 파일로 구성하는 것을 모사하는 것이기 때문에 커널 설정에서는 기존의 기법인 file-backed storage gadget(FSG)를 사용하는 것으로 아래와 같이 커널 설정을 한다.

```
Configure kernel compile option in 'Make menuconfig'
 Device  Driver ->  USB Support -> USB Gadget ->
                <*>  Support for USB Gadget
 Device  Driver ->  USB Support -> Driver Mode ->
                <*> Gadget Stack
 Device  Driver ->  USB Support -> USB Gadget ->
                 USB peripheral controller
                <*> Inverntra (M) USB peripheral
                <M> USB Gadget driver
                <M> File-backed Storage gadget
```

커널은 USB 디스크를 USB 파일로 인식되게 모사하고 USB 파일은 커널 모듈에서 다시 커널 모듈 상의 정적 데이터로 구현한다. 정적 데이터는 파일시스템과 같은 구성을 갖게 하기 위해 우선 디스크 모사 파일을 만들고 그 모사된 파일을 이용해서 정적 데이터를 만든다.



[그림 6] USB 인증연결 시스템 구성

가상 USB 구성에 의한 인증구현은 그림 6과 같다. 가상 USB는 ESX 하이퍼바이저 상에서 구현되고 게스트 가상머신으로 연결될 수 있다. 가상 USB가 ESX 서버에서 구현되어 게스트 머신에서 참조되려면 USB 저장장치 구현만으로 구현할 수 없다. 기본적으로 USB 장치가 참조되려면 USB 호스트 콘트롤러 자체가 ESX 서버 상에 설치되어야 한다. 이것은 상업용 가상머신 서버의 요구사항이다. 즉 가상 USB 장치를 통해서 인증을 구현하려면 가상 USB 뿐만아니라 USB상위에 USB 호스트제어기가 구현되어야 하고 ESX에서 가상머신 구성할 때 USB 호스트 제어기를 가상머신 구성요소로 할당하는 것에 의해 USB 디스크를 구성할 수 있다.

4. 결론

본 연구는 가상화 시스템의 시스템 제어 플랫폼과 인증방법에 관한 것이다. 가상화 시스템은 컴퓨터 기술의 진보와 더불어 점차로 확산되고 있는 기술이다. 실제로 가상화 시스템을 통해서 스마트워크 환경을 구성하고 있고, 가상화 시스템을 통해 그린 컴퓨팅의 일환으로 연간 111만톤의 탄소 배출량을 절감할 수 있다는 조사가 이루어진 바도 있다. 이에 기존 시스템에 가상화 시스템의 적용은 미래 컴퓨터 구성의 모습이라 할 수 있다. 이러한 추세에 걸맞게 가상화 시스템의 구성은 점점 복잡해지고 있으며 양적으로도 성장하고 있다. 상업용 가상화 시장의 매출이 2010년에 전년에 비해 50% 정도의 성장을 보였다. 기타 서버와 데스크탑 시장에서의 가상화 시장 점유는 컴퓨팅 시장의 괄목할 성장을 보이고 있다. 이에 따라 시장 조사기관은 날로 발전하는 가상화 기술의 보안과 제어에 대한 우려를 나타내고 있고 시스템 제어 대책의 마련이 발전을 순탄하게 지속해 줄 것으로 예측하고 있다. 이러한 이유로 본 연구는 가상화 시스템의 보안과 제어에 대한 한 방법을 제시하였다. 그리고 사용자가 할당받은 가상머신 상에서의 인증을 위한 방법도 제시하였다.

제어 플랫폼 커널에는 로그 관리자, 네트워크 관리자, 프로그램 관리자, 인증 관리자가 올라가게 되며 이러한 관리자를 통해 ESX 서버 내의 전체 가상머신을 관리하고 예견되지 않은 접근을 차단 보호하느 역할을 한다. 할당받은 가상 머신의 인증 수단을 위해서 ESX 하이퍼바이저 시스템 상에 새로운 USB 호스트 제어기를 만들고 USB 디스크 형태를 제공하였다. USB 디스크 형태의 지원을 위해 ESX 커널 상에서는 USB 디스크를 하나의 파일 구성이 될 수 있게 구성했다. ESX 하이퍼바이저 모듈에서는 파일시스템 기능을 제공하지 않기 때문에 이를 위해 커널 모듈의 정적 데이터 형태로 구현하였다. USB 호스트 제어기나 USB 디스크를 소프트웨어로 구현하여 수정이나 인증 제어수단의 구현시 수월성을 제공하였다. 하드웨어적인 수정에 대해 소프트웨어 수정으로 가능하게 구성한 것이다. 가상머신은 실제 USB 메모리 등의 인증을 위한 장치를 가지지 않고 소프트웨어적으로 연결을 할 수 있게 되었다. 이러한 소프트웨어 구성은 ESX 시스템의 보안수준에서 보호받을 수 있으며 소프트웨어이므로 조작이 간편하다. 본 논문에서는 가상화 시스템이 날로 확산되고 있는 시점에서 제어에 대한 플랫폼을 제공

하고 인증을 위한 수월한 방법론을 제시하였다.

참고문헌

[1] USB Implementers Forum, Inc., USB 2.0 specification, http://www.usb.org/developers/docs/
[2] Backing Storage for the File-backed Storage Gadget, http://www.linux-usb.org/gadget/file_storage.html
[3] 유진호, “USB/IP를 이용한 원격장치공유에 대한 연구”, 산학기술학회 논문지, 제11권, 제11호, pp. 4592-4596, 11월, 2010.
[4] D. Bem and E. Huebner, Computer Forensic Analysis in a Virtual Environment: University of Western Sydney, 2007
[5] VMWare, "VMWare," 2007; http://www.vmware.com
[6] VMWare Server, "VMWare Server," 2007; http://www.vmware.com/product/server

유 진 호(Jinho Yoo) **[정회원]**

- 1996년 2월 : 서강대학교 컴퓨터공학과 (전산학석사)
- 2006년 8월 : 충북대학교 컴퓨터학과 (전산학박사)
- 1996년 1월 ~ 1998년 12월 : 엘지정보통신연구소 전임연구원
- 1999년 1월 ~ 2008년 2월 : 한국전자통신연구원 선임연구원
- 2008년 3월 ~ 현재 : 백석대학교 정보통신학과 교수

<관심분야>
임베디드시스템, HCI, 가상화 시스템

이 병 철(Byoungchul Lee) **[정회원]**

- 1996년 2월 : 한국과학기술대학교 전산과 (전산학학사)
- 1999년 1월 ~ 2001년 2월 : (주)노아에이티에스 대표이사
- 2001년 2월 ~ 2008년 10월 : (주)시큐브레인 대표이사/연구소장
- 2009년 4월 ~ 현재 : (주)센트루소프트웨어 대표이사/연구소장

<관심분야>
가상화시스템, 클라우드컴퓨팅 보안, 시스템 보안