클라이언트 가상화를 이용한 중요정보 보호

임세정* · 김광준** · 강태근***

Important Information Protection using Client Virtualization

Se-jung Lim* · Gwang-jun Kim** · Tae-geun Kang***

요 약

본 논문에서는 클라이언트 가상화 기술을 이용하여 로컬 컴퓨팅 환경의 성능 저하를 최소화 하고 가상화된 사용자 영역에서 사용자가 필요한 기능을 사용할 수 있도록 효율적으로 제공하며, 로컬 컴퓨팅 환경의 중요 정보 보호와 성능의 안정성 및 지속성을 유지하였다. 또한 로컬 컴퓨팅 환경 뿐만 아니라 악성코드와 같은 공격으로부터 가상화된 영역을 보호하기 위한 방법을 제안함으로서 가상화된 영역에 있는 데이터들의 암호화를 통하여 가상화된 사용자 영역의 보안을 극대화시켰다. 가상화를 통해 로컬 컴퓨팅 자원을 그대로 사용하면서 효율적으로 로컬 컴퓨팅 시스템으로부터 하나의 사용자 컴퓨팅 리소스를 분리시키는 것과 같은 효과를 얻을 수 있다.

ABSTRACT

In this paper, proposed client virtualization technology to minimize degradation of the local computing environment, efficient and qualified users in the area of virtual functions needed to enable the user to provide important information in the local computing environment protection and performance, stability and continuity was important to keep. As well as the local computing environment from malicious code attacks such as methods for protecting virtualized domain also can not be overlooked as a major problem area in a virtualized, virtualized data through the encryption of user-space security, maximized. In addition, through virtualization using local computing resources efficiently while still a local computing system separate from the computing resources to a single user can get the same effect.

키워드

Client Virtualization, Stability, Continuity, Encryption, Local Computing Resource

I. 서 론

가상화 기술의 급격한 발전은 고속 유무선 네트워크 환경 구축과 시스템 하드웨어 자원의 성능향상 그리고 경제성 확보가 이루어진 데 있다. 또한 하드웨어 자원으로부터의 의존성을 탈피시켜줌으로서 기존 기법과는 다른 혁신적인 접근 방식을 통해 새로운 문제해결 방식을 가능하게 되었다. 가상화(Virtualization)는 컴퓨터 환경에서 컴퓨터 자원의 추상화를 일컫는 범용 용어로서 실제 존재하는 물리적 자원들의 특징을 숨기고 논리적인 자원의 집합을 구성하여 이러한 자원과 상호작용하는 다른 시스템, 응용 프로그램, 사용자에게 제공하는

* 고려대학교 컴퓨터전파통신공학과(limsejung@korea.ac.krr)
** 교신저자 :전남대학교 컴퓨터공학과(kgj@jnu.ac.kr)
*** 전남대학교 컴퓨터공학과(xorms123@jnu.ac.kr)
접수일자 : 2011. 12. 27
심사(수정)일자 : 2011. 01. 28
게재확정일자 : 2011. 02. 09

기술로 정의할 수 있다. 이는 서버, 운영체제, 응용 프로그램, 저장장치 등과 같은 물리적인 자원을 여러 개의 논리적인 자원으로 보이도록 할 수 있으며 여러 개의 물리적인 자원을 하나의 논리적인 자원으로 나타낼 수도 있다[1][2].

PC, 또는 서버에서 요구되는 프로세스 처리량이 급격하게 증가함에 따라 소프트웨어 개발 및 점검을 보다 손쉽게 제공하며, 클라이언트의 개발 환경 및 중요정보 보안 인프라를 구축할 수 있는 새로운 가상화 기술이 필요하다. 가상화 환경에서 데스크탑 가상화는 시스템을 운영하는 과정에서 자원의 고갈이나 성능저하, 업그레이드 등과 같은 문제를 해결하고, 시스템 관리 및 클라이언트의 중요한 정보가 보호될 수 있어야 한다. 또한 가상머신의 독립된 환경을 구성하여 편리한 개발 환경 및 보안 인프라를 구축하며, 외부로부터 공격이나 내부 결함으로 인해 가상 머신 자체 문제가 발생하거나 서비스 장애가 발생하더라도 신속하게 대응하고 복구할 수 있도록 지원하여야 한다[3][4].

제안된 클라이언트 가상화 기술 기반의 소프트웨어 가상화는 가상 하드웨어 없이 로컬 컴퓨팅 환경의 하드웨어를 그대로 사용하기 때문에 하드웨어 가상화에서 발생하는 가상 하드웨어의 오버헤드 발생 없이 로컬 하드웨어의 모든 기능이 사용 가능하다. 또한 호스트 운영체제와 응용 프로그램을 그대로 사용하므로 가상화된 사용자 영역에서 요구하는 가상 영역의 크기를 최소화하면서 백업 및 네트워크 전송시 유리하다.

로컬 컴퓨팅 환경에 가상화된 사용자 영역을 구축함으로써 외부로부터의 로컬 컴퓨팅 환경에 접근하려는 침입을 차단하여 안전하게 보호하고 것 뿐만 아니라 인터넷을 통하여 로컬 컴퓨팅 환경에 원치 않는 프로그램이 설치되거나 나쁜 영향을 주는 악성 코드나 바이러스들로부터 로컬 컴퓨팅 환경을 보호 할 수 있다. 또한, 가상화된 사용자 영역에 저장되어 있는 중요 정보가 외부로 유출되는 것을 막기 위해 암호화된 가상의 사용자 영역을 구축하였다.

II. 클라이언트 가상화

2-1. 하드웨어 가상화

하드웨어 가상화는 그림 1에서와 같이 로컬 컴퓨팅 환경의 하드웨어와 호스트 운영체제 안에 각각의 가상 하드웨어가 존재한다. 가상 하드웨어에는 가상 환경에 따라 운영 체제와 응용 프로그램 설치가 필요하여 운영체제와 응용 프로그램의 라이센스 비용과 설치가 필요하다.

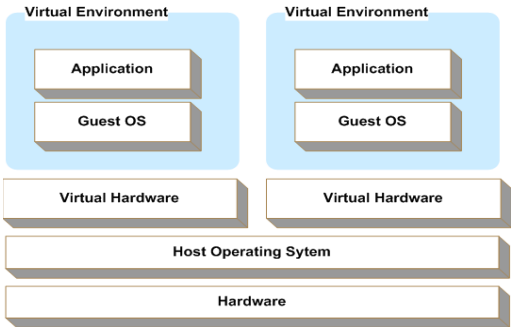

그림 1. 하드웨어 가상화
Fig. 1 Hardware Virtualization

하드웨어 가상화의 가상 환경 크기는 운영 체제와 응용 프로그램이 설치되기 때문에 운영체제와 응용 프로그램을 포함하여 최소 500MB에서 1GB를 필요하므로 백업 및 네트워크 전송시 부하를 발생 시킬 수 있다[5][6]. 하드웨어 가상화는 가상 하드웨어의 오버헤드 발생으로 로컬 하드웨어의 GPU 하드웨어 가속 등을 사용하지 못한다.

2-2. 소프트웨어 가상화

소프트웨어 가상화는 가상 하드웨어 없이 로컬 컴퓨팅 환경의 하드웨어를 그대로 사용하기 때문에 하드웨어 가상화에서 발생하는 가상 하드웨어의 오버헤드 발생 없이 로컬 하드웨어의 모든 기능이 사용 가능하다. 호스트 운영체제와 응용 프로그램을 그대로 사용하므로 가상 환경의 크기는 최소 10MB~50MB로 하드웨어 가상화에서 요구하는 가상 환경의 최소 크기에 비해 매우 작아서 백업 및 네트워크 전송시 유리하다. 소프트웨어 가상화 기술은 다시 커널 레벨 가상화 기술과 유저 레벨 가상화 기술로 구분된다. 커널 레벨 가상화는 각각의 가상 환경이 생성하는 시점에 커널을 구성하고 응용 프로그램을 설치해야 한다.

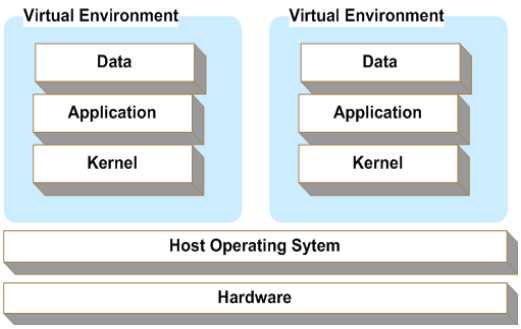

그림 2. 커널 레벨 가상화
Fig. 2 Kernel Level Virtualization

그림 2와 같이 커널 레벨 가상화를 통해 생성된 가상 환경에서는 커널을 공유하지 않으므로 로컬 컴퓨팅 환경에 설치되어 있는 응용 프로그램 뿐만 아니라, 서로 다른 가상 환경의 응용 프로그램을 공유하는 것이 어렵다. 커널 레벨 가상화는 운영체제 내부 구조에 의존적이어서 운영체제의 버전에 따라서 구조적인 변경이 필요하다.

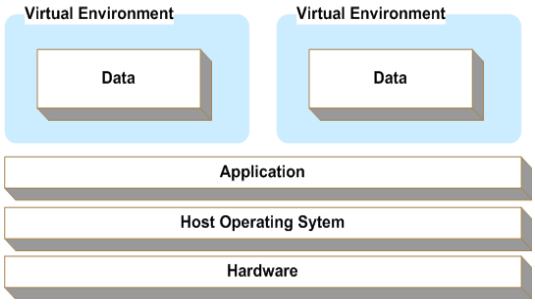

그림 3. 유저 레벨 가상화
Fig. 3 User Level Virtualization

커널 레벨 가상화와 유저 레벨 가상화의 운영 체제의 의존성을 비교해 보면 유저 레벨 가상화는 운영체제의 API에 의존적이다. 그림 3과 같이 유저 레벨 가상화는 운영체제의 버전에 따라서 구조적인 변경이 필요한 커널 레벨 가상화와 다르게 버전이 다른 운영체제도 API의 호환성은 유지되므로 가상 환경의 구조적인 변경이 필요하지 않다. 유저 레벨 가상화는 가상 환경을 생성할 때 초기화만으로 생성되므로 사용자의 요청시 실시간 생성과 사용이 가능하고 로컬 컴퓨팅 환경에 설치되어 있는 응용 프로그램을 그대로 사용하므로 응용 프로그램에 대한 라이센스 비용과 추가 설치가 필요하지 않다.

III. PC 보안을 위한 가상화

3-1. 커널 모드와 유저 모드의 구조

로컬 컴퓨팅 환경에서 동작하는 잘못된 응용 프로그램으로부터 시스템 보호를 위해 운영체제는 운영 체제의 코드를 실행하는 커널 모드와 사용자 응용 프로그램을 실행하는 유저 모드를 사용한다. 유저 모드에서 실행되는 응용 프로그램은 제한된 인터페이스의 집합만이 가능하고 시스템 데이터에 대한 액세스를 제한하고 하드웨어에 대한 직접적인 접근이 불가능하다. 유저 모드 프로그램이 시스템 서비스를 호출할 때 프로서서는 호출을 가로채서 스레드를 커널 모드로 전환하도록 호출한다. 시스템 서비스가 완료될 때 운영체제는 스레드 컨텍스트를 다시 유저 모드로 전환하고 호출자가 계속해서 실행하도록 한다. 윈도우는 OS/2, POSIX, Win32의 환경 서브시스템을 가지고 있다. 윈도우가 운영체제로 동작하기 위해 모든 코드를 가진다면 많은 양의 중복 함수를 가지기 때문에 시스템 크기와 성능에 대한 부정적인 영향을 고려하여 Win32가 주 서브시스템으로 설계되었다. 다른 두 서브 시스템들은 요구에 의해 구성되는 반면 Win32 서브 시스템은 항상 실행해야 한다. 윈도우의 환경 서브시스템 프로세스에서는 콘솔 윈도우들, 프로세스와 스레드의 생성과 종료, 16비트 가상 도스 머신 프로세스에 대한 지원, 내부 언어 지원함수들을 지원한다.

커널 모드 장치 드라이버는 윈도우 디스플레이 제어, 화면 출력 관리, 키보드와 마우스를 비롯한 다른 장치들로부터의 입력 수집, 응용 프로그램에 사용자 메시지 전달을 지원하고 GDI(Graphics Device Interface) 함수를 포함한다.

그래픽스 장치 드라이버들은 하드웨어 종속적인 그래픽스 디스플레이 드라이버, 프린터 드라이버, 비디오 미니포트 드라이버 등이다. 로컬 PC에서 실행되는 응용 프로그램은 직접 Win32 주 서브시스템을 호출하지 않고, 하나 이상의 서브 시스템 DLL들을 통해서 호출한다. 서브시스템 DLL들은 문서화된 윈도우 API 함수들을 문서화 되지 않은 커널 모드 시스템 서비스를 호

출로 변환한다.

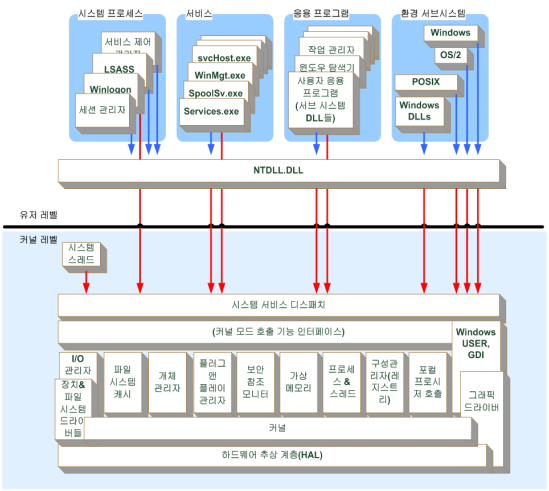

그림 4. Windows 아키텍쳐 처리과정
Fig. 4 Processing of Windows Architecture

3-2. 유저 레벨의 메시지 후킹

Kernel32.dll, User32.dll, Advapi32.dll, Gdi32.dll과 같은 윈도우 서브시스템 DLL들은 윈도우 API 함수들을 구현해 놓은 것이다. 그림 5는 유저 레벨에서 윈도우 서브시스템 DLL내의 함수를 호출하여 커널 레벨로 진입하여 실행하는 과정을 나타낸다. 로컬 PC에서 동작하는 응용 프로그램은 윈도우 서브시스템 DLL을 로드하고 그것의 임포트 어드레스 테이블(IAT)에서 각 함수들의 메모리 주소를 복사한다. 디렉토리 내의 파일을 나열하기 위해 응용 프로그램이 Kernel32.dll에서 제공하는 FindFirstFile API를 호출하면 IAT에 있는 FindNextFile 함수의 주소로 이동하여 Ntdll.dll을 호출한다.

Ntdll.dll은 윈도우 서브시스템 DLL을 사용하기 위한 특별한 주요 시스템 지원 라이브러리이며 윈도우 실행부 시스템 서비스에 대한 시스템 서비스 디스패치 스텁과 서브시스템, 서브시스템 DLL, 다른 네이티브 이미지에 의해 사용되는 내부 지원 함수들을 포함한다. Ntdll.dll은 FindNextFile에 대응하는 커널 레벨 함수인 NtQueryDirectoryFile의 번호를 EAX 레지스터에 로드하고 EDX 레지스터에는 FindNextFile 인자의 유저 레벨 주소를 로드한다. SYSENTER와 INT 2E 명령을 이용하여 Ntdll.dll은 커널 레벨로 진입한다.

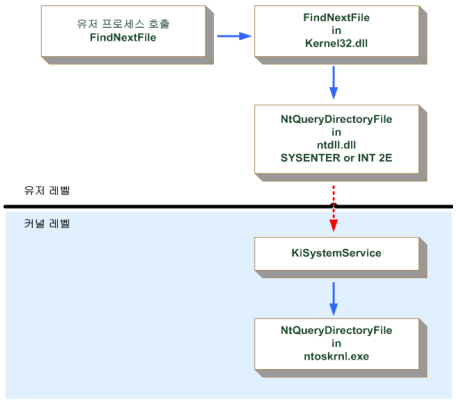

그림 5. FindNextFile의 실행 흐름
Fig. 5 Execution Flow of FindNestFile

윈도우 서브시스템 DLL이 응용 프로그램 메모리 주소 공간에 로드되기 때문에 응용 프로그램의 메모리 주소 공간에 접근하게 되면 API를 변경시키거나 응용 프로그램 임포트 테이블 변경이 가능한데 이것이 API 후킹이다. API 후킹을 이용하면 프로세스나 네트워크 포트를 숨길 수 있고 파일에 대한 쓰기가 다른 파일에서 이뤄지도록 할 수 있으며 응용 프로그램이 특정 프로세스의 핸들을 구하지 못하게 할 수 있다. 또한, API 호출을 로그로 남긴다면, 응용 프로그램이 어떤 식으로 동작하는지 알 수 있다.

API 후킹 방식을 동적인 API 후킹 방식과 정적인 API 후킹 방식으로 구분할 수 있다. 동적인 API 후킹 방식은 런타임시에 사용자의 요구에 따라 API를 상황에 따라 다르게 후킹하는 것이 특징이다. 본 논문에서는 특정 코드로 특정 API만을 후킹하는 정적 API 후킹 방식을 사용 하였다.

정적인 API 후킹 방식은 API 후킹 코드를 작성하면 다시 컴파일 해야 하는 단점을 갖는다. 하지만, 동적인 API 후킹 방식과 비교하면 구현이 쉽고 안정적이며 API에 따라서 후킹을 한 후의 동작 구현이 다양하다.

Ⅳ. 가상화 사용자 영역 실험 결과 및 분석

가상화된 사용자 영역 테스트는 Microsoft Windows XP Professional Version 2002 Service Pack 3에 1.0GB RAM의 테스트 PC에서 진행한다.

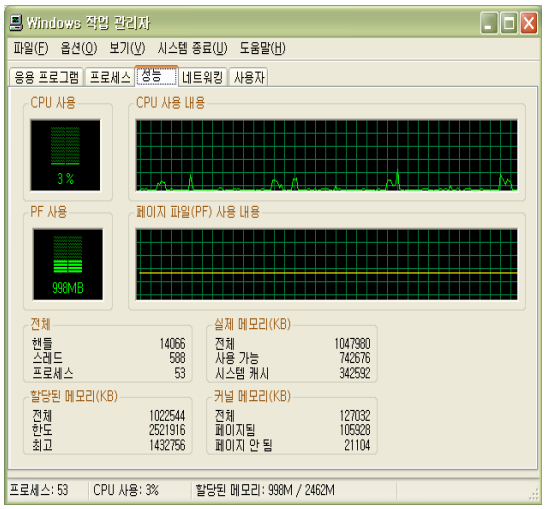

그림 6. 디스크 등록정보 (가상화된 사용자 영역 테스트 전)
Fig. 6 Disk Properties (Before Test of Virtualized User Domain)

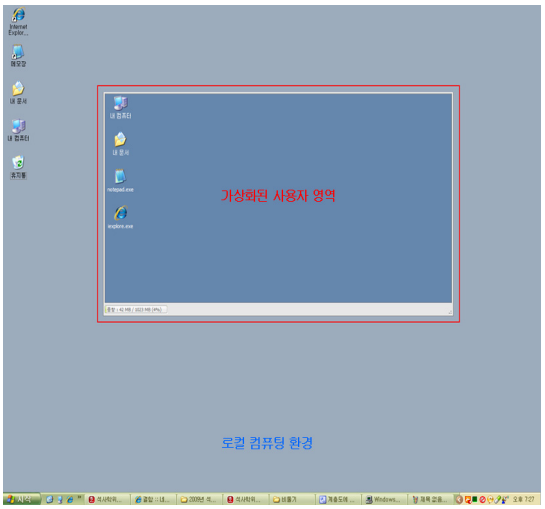

그림 7. 가상화된 사용자 영역 실행
Fig. 7 Execution of Virtualized User Domain

그림 6은 가상화된 사용자 영역의 테스트를 진행하기 전 로컬 컴퓨팅 환경의 성능 변화를 나타내는 것으로서 가상화된 사용자 영역의 테스트를 진행하는 과정에서의 성능 변화와 비교할 수 있는 중요한 척도가 된다. 가장 먼저 로컬 컴퓨팅 환경에서 가상화된 사용자 영역의 테스트를 위하여 가상 데스크톱으로 사용하기 위한 가상화된 사용자 영역의 생성이 필요하다. 이렇게 생성되는 가상화된 사용자 영역은 로컬 컴퓨팅 환경에서 사용 가능한 공간의 크기를 초과해서 생성할 수 없다.

가상화된 사용자 영역에서의 사용 공간은 생성 시간을 고려하여 1G Byte의 크기로 생성하도록 하였다. 이렇게 생성된 가상화된 사용자 영역은 사용자의 삭제 명령을 통해 실시간 삭제 및 재생성도 가능하지만, 중복 생성은 허용하지 않는다. 이것은 로컬 컴퓨팅 환경에서 가상화된 사용자 영역을 오직 하나만 존재할 수 있도록 설계하였기 때문이다. 가상화된 사용자 영역이 정상적으로 구축되면 그림 7과 같이 로컬 컴퓨팅 환경에 가상화된 사용자 영역이 응용 프로그램의 형태로 구동된다.

가상화된 사용자 영역의 생성이 완료되면 실행할 응용 프로그램을 등록해야 한다. 응용 프로그램 등록은 로컬 컴퓨팅 환경에 존재하는 .exe 확장자의 실행 파일을 가상화된 사용자 영역에 등록하거나 로컬 컴퓨팅 환경에 존재하는 .exe 실행 파일을 가상화된 사용자 영역으로 드래그하여 등록할 수 있다.

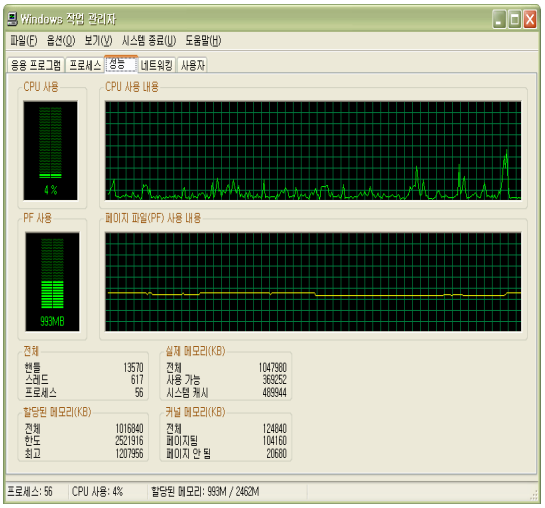

그림 8. 테스트 PC의 성능 (가상 디스크 생성 중)
Fig. 8 Performance of Test PC (Creating Virtual Disk)

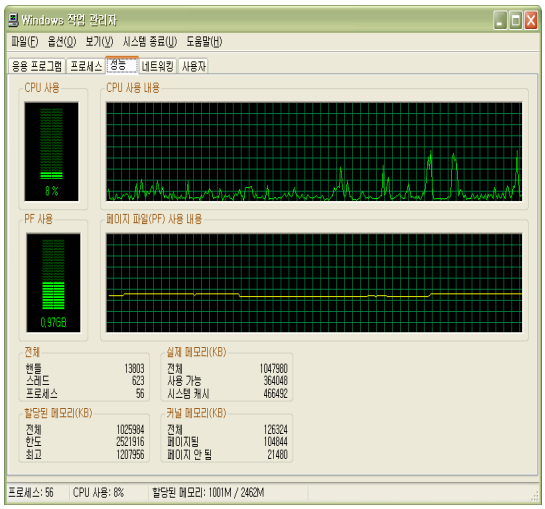

그림 9. 테스트 PC의 성능 (가상 디스크 생성 완료)
Fig. 9 Performance of Test PC (Virtual Disk Creation Completed)

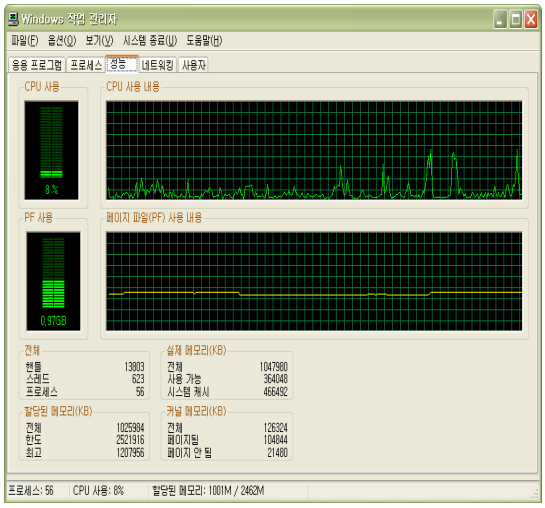

그림 10. 테스트 PC의 성능 (응용 프로그램 등록)
Fig. 10 Performance of Test PC (Application Program Registration)

테스트를 위하여 iexplore.exe와 notepad.exe를 가상화된 사용자 영역에 추가하면 그림7과 같이 로컬 컴퓨팅 환경에서 사용하는 아이콘이 가상화된 사용자 영역에도 생성되고 추가된 응용 프로그램의 목록을 관리하고 실행 파일에 대한 자세한 정보를 확인할 수 있다.

그림 8과 9는 가상화된 사용자 영역에서 실행할 응용 프로그램을 등록하기 전, 즉 가상 디스크 생성이 정상적으로 종료된 상태의 테스트 PC의 성능을 보여주고 있으며, 그림 10은 테스트를 위하여 응용 프로그램 Windows Internet Explorer와 Notepad를 등록한 후의 테스트 PC의 성능을 나타내고 있다.

가상화된 사용자 영역의 생성과 동작할 응용 프로그램의 실행 파일의 등록이 완료되면 로컬 컴퓨팅 환경의 실제 데스크톱과 구별되는 가상화된 사용자 영역에서 데스크톱이 테스트 PC에서 동작한다.

V. 결 론

본 논문에서는 가상화된 사용자 영역에서 높은 신뢰성과 실시간 성능을 보장하고 자원의 효율적인 활용을 위한 새로운 클라이언트 기반의 가상화 기술을 적용하였다. 가상화 기술의 방대한 기술에서 필요한 기술 요소를 찾아내고 로컬 컴퓨팅 환경에 사용자에게 보다 편리하고 안정된 새로운 가상화 기술을 제안하였다. 제안된 가상화 기술로 로컬 컴퓨팅 환경의 성능 저하를 최소화 하여 효율적으로 가상화된 사용자 영역에서 필요한 기능을 사용할 수 있도록 제공하는 과정에서 로컬 컴퓨팅 환경의 중요 정보 보호와 성능의 안정성 및 지속성을 유지하는 것이 중요하였다. 또한, 가상화된 사용자 영역은 악성코드와 같은 공격으로부터 보호하기 위하여 데이터 암호화를 통하여 보안을 극대화시켰다. 이것은 가상화를 통해 사용자 로컬 컴퓨팅 자원을 그대로 사용하면서 효율적으로 로컬 컴퓨팅 환경으로부터 하나의 사용자 컴퓨팅 리소스를 분리시키는 것과 같은 효과와 로컬 컴퓨팅 환경에 저장되어 있는 중요 정보의 보호와 가상화된 사용자 영역의 보호에 효과가 있었다.

참고 문헌

[1] 김인혁, 김태형, 김정한, 임병홍, 엄영익, “시스템 보안을 위한 가상화 기술 활용 동향”, 정보보호학회지, Vol. 19, No. 2, pp. 26-34, 2009.
[2] 홍대영, 고원석, 임성수, “보안과 신뢰성 있는 컴퓨팅을 위한 가상화 기술”, 정보과학회지, Vol. 26, No.10, pp. 50-57, 2008.
[3] B. Lampson, M. Abadi, M. Burrows and E.

Wobber, “Authentication in Distributed Systems: Theory and Practice”, ACM Transactions on Computer Systems, pp. 265-310, 1992.
[4] Heiser Gernot, Elphinstone, Kevin Kuz, Ihor Klein, Gerwin, Petters, and Stefan M. “Towards Trustworthy Computing Systems: Taking Microkernels to the Next Level”, Operating Systems Review. Vol. 41, No. 3, pp. 3-11, 2007.
[5] Samuel T. King, Peter M. Chen, Yi-Min Wang, Chad Verbowski, Helen J. Wang and Jacob R. Lorch, “SubVirt:Implementing malware with virtual machines”, Proceedings of the 2006 IEEE Symposium on Security and Privacy, 2006.

저자 소개

임세정(Se-jung Lim)

2008년 전남대학교 컴퓨터공학과(공학사)
2010년 전남대학교 대학원 컴퓨터공학과 졸업(공학석사)
현재 고려대학교 대학원 컴퓨터·전파통신공학과 (박사과정)
※ 관심분야 : 가상화, ATM망, 실시간 데이터통신, 컴퓨터 네트워크, TCP/IP 혼잡제어, 생체정보 및 의료정보, 무선인터넷, 이동통신 등

김광준(Gwangl-jun Kim)

1993년 조선대학교 컴퓨터공학과 졸업(공학사)
1995년 조선대학교 대학원 컴퓨터공학과 졸업(공학석사)
2000년 조선대학교 대학원 컴퓨터공학과 졸업 (공학박사)
2000년~2001년 Dept. of Electrical & Computer Eng. Univ. of California Irvine Postdoc.
2003년~현재 전남대학교 컴퓨터공학과 조교수
※ 관심분야 : 가상화, ATM망, 인터넷 통신, 컴퓨터 네트워크, 실시간 통신 프로그래밍, 영상 처리 및 통신, 프로그래밍 언어(Visual C++, Java), 의료정보통신 등

강태근(Tae-Guen Kang)

2009년 전남대학교 컴퓨터공학과 (공학사)
현재 전남대학교 대학원 컴퓨터공학과 (석사과정)
※ 관심분야 : 가상화, ATM망, 컴퓨터 네트워크, 실시간 데이터 통신 등