情 報 保 護 學 會 誌
第19卷 第2號, 2009. 4

시스템 보안을 위한 가상화 기술 활용 동향

김 인 혁*, 김 태 형*, 김 정 한*, 임 병 홍*, 엄 영 익*

요 약

가상화 기술은 소프트웨어적으로 독립된 가상환경을 제공함으로써 관리 수단 통합, 자원 활용 극대화, 안정된 서비스 제공 및 비용 절감 등 많은 이점을 부여하며 다양하게 활용되고 있다. 가상화를 적극적으로 활용하고 있는 분야로는 서버 가상화, 실시간 시스템, 시스템 모니터링, 클라우드 컴퓨팅 등을 대표적으로 꼽을 수 있으며, 시스템 보안 분야에서도 가상화의 이점을 적용하여 한층 높아진 보안 서비스를 제공하고 있다. 이러한 동향에 따라 본 논문에서는 먼저 가상화 기술에 대해 전반적으로 살펴보고, 이를 기반으로 시스템 보안 분야에서 가상화 기술이 어떻게 활용되고 있는지 알아본다.

Ⅰ. 서 론

최근 들어 세계적으로 많은 기업들이 안정된 서비스를 제공하고 비용절감을 위해서 가상화 환경을 구축하여 운영하고 있는 추세이다. 국내 기업들도 가상화 기술에 대해 관심을 갖고 적극적으로 접근하고 있으며, 가상화 환경을 제공하기 위해 시험 운용 준비하는 등 관련 업계들은 새로운 국면을 맞이하고 있다. 또한 관리 효율성을 높이면서 최근 이슈가 되고 있는 그린 IT를 위한 기반기술로도 기대를 모으고 있기에, 가상화 기술은 IT 각 분야에서 앞다투어 연구되고 있으며 많은 영향력을 끼치고 있다.

이와 마찬가지로 시스템 보안 및 관리 분야에서도 보안 향상 및 관리 기술 혁신을 위해서 가상화 기술을 적극적으로 반영하고 있다. 기존 악성코드 탐지를 위한 침입 탐지 시스템의 경우 시그너쳐 기반 정적 분석이 주류를 이루며 오탐지가 잦았던 반면, 가상화 기술을 적용한 경우 독립된 환경에서 안정적으로 실시간 탐지가 가능해지면서 더욱 향상된 서비스를 기대할 수 있게 되었다. 그리고 시스템 재생을 위해서 가상화 기술을 적용하면서 기존에 어려웠던 결정/비결정 이벤트들에 대한 로깅 및 재생이 가능해졌다. 또한, 가상환경을 이용하여 허니팟과 같은 악성코드 분석 시스템을 제공할 수 있게 되었으며, 도메인 보안 수준에 따라 상이한 서비스를 제공할 수 있게 하는 등 보안 분야에서는 가상화 기술의 장점들을 충분히 활용하여 다양하게 이용하고 있다.

본 논문에서는 가상화 기술이 시스템 보안에 미치는 영향에 대해 살펴보고자 한다. 2장에서는 가상화 기술 및 응용분야에 대해 설명하고 3장에서는 시스템 보안을 위한 가상화 활용 사례에 대해 살펴본다. 4장에서는 향후 전망에 대해 기술하며 결론을 맺는다.

Ⅱ. 가상화

2.1 가상화 소개

최근 IT 분야에서는 하드웨어와 네트워크 성능이 향상되고, 이를 기반으로 더 나은 서비스를 제공하기 위해 가상화 기술을 적극적으로 활용하기 시작하면서 가상화가 뜨거운 이슈로 떠오르고 있다. 서버, 운영체제, 애플리케이션, 스토리지 및 네트워크 등 IT 인프라 거의 모든 측면에서 가상화 기술을 이용하여 성능 향상을 꾀하고 있다. 각 분야 마다 가상화 기술을 적절히 활용하여 복잡했던 환경을 단순화시키고 작업처리의 분산, 관리 수단의 통합 등을 통해 비용 절감과 관리 능률 향상 등 긍정적 효과를 얻고 있다.

가상화 기술을 이용하는 방식은 크게 세 가지로 나누어 볼 수 있다. 시스템 자원의 추상화, 분배 그리고 병합으로 구분한다. 먼저, 자원의 추상화는 자원을 이용하는 주체에게 복잡한 물리적인 속성을 숨기고, 주체가 필

* 성균관대학교 정보통신공학부 ({kkojiband, kim15m, gtgkjh, dd8562, yieom}@ece.skku.ac.kr)

요로 하는 형태로 논리적인 자원을 제공하는 것으로 자바 가상머신을 대표적인 예로 들 수 있다. 그리고 서버 가상화와 같이 하나의 시스템 자원을 다수의 접근 주체들에게 독립적인 자원으로 인식시켜 이용할 수 있도록 하는 자원의 분배 방식이 있다. 또한, 분리되어 있는 자원들을 하나의 자원으로 인식 할 수 있도록 하여 방대한 자원을 효율적으로 이용할 수 있도록 가상환경을 만들어 주기도 한다.

이와 같은 가상화 기술은 근래 새롭게 대두된 분야는 아니다. 과거 이슈화되었다가 다시금 주목 받게 된 기본 계기에는 고속 유무선 네트워크 환경 구축과 시스템 하드웨어 자원의 성능향상 그리고 경제성 확보가 이루어진 데 있다. 이를 바탕으로 새로운 기술들을 접목시키면서 과거 가상화 활용에 걸림돌이 되었던 제약 사항 들이 많은 부분 해소됨에 따라, 현재 당면한 각종 과제들은 가상화 기술을 적용하여 효과적으로 극복하고 해결할 수 있게 되었다. 무엇보다 하드웨어 자원으로부터의 의존성을 탈피시켜줌으로써 기존 기법과는 다른 혁신적인 접근 방식을 통해 활로를 개척하여 새로운 문제해결 방식을 가능하게 되었다.

더불어 PC, 또는 서버에서 요구되는 프로세스 처리량이 급격하게 증가함에 따라 소프트웨어 개발 및 점검을 보다 손쉽게 제공할 수 있는 가상화 기술이 강력한 핵심 기술로 요구되고 있다.

이와 같은 이유로 가상화가 핵심기술로 활용되면서 많은 시스템들에 대한 운용 및 관리에 유연성과 효율성을 더하게 되었다. 특히, 시스템을 운영하는 과정에서 발생하는 자원의 고갈이나 성능저하, 업그레이드 등과 같은 문제들을 해결하기 위해 시스템 이주 작업이 필연적으로 요구 된다. 이 과정에서 시스템 중단이나 관리 어려움과 같은 문제가 존재하는데, 가상 환경에서는 하드웨어 추상화를 통해서 운영체제와 애플리케이션의 이주가 편리하기에 시스템 관리를 효율적으로 운용할 수 있게 되었다. 이처럼 가상화 기술을 활용한다면 시스템 중단 및 점검 시간을 최소화하려는 노력 없이도 시스템을 유지 보수할 수 있기 때문에 안정된 서비스를 원활히 제공하는 등 많은 이점이 있다.

또한, 가상머신 간 독립된 환경을 구성하여 편리한 개발 환경 및 보안 인프라를 구축할 수 있도록 한다. 그리고 외부로부터의 공격이나 내부 결함으로 인하여 가상머신 자체 문제가 발생하거나 서비스 장애가 발생하더라도 가상 환경에서는 이에 대해 신속하게 대응하고 복구할 수 있도록 지원하고 있다.

2.2 가상화 기법

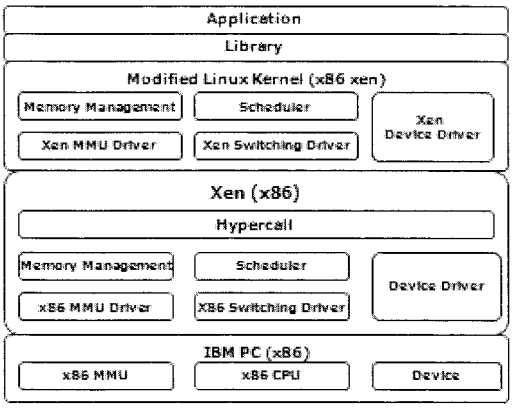
가상화 기술은 시스템 구성 계층구조 상에서 어느 계층에서 추상화를 통해 가상화를 구현하느냐에 따라 그 기법이 구분된다. 이번 장에서는 일반적인 가상화 기법 네 가지를 구분하여 소개하고, 각각의 장단점을 비교하여 설명한다[1].

2.2.1 OS-level 가상화

가장 먼저 OS-level 가상화의 경우, 프로세스에게 필요한 자원들을 분배하여 각각의 프로세스들이 독립적인 환경에서 동작할 수 있도록 지원하는 기법이다. 하나의 운영체제를 기반으로 동작하며, 전가상화나 반가상화와 같이 특권 명령어 처리에 대해서 고려할 필요가 없기 때문에 우수한 성능을 나타내며 동작하는 특징을 갖고 있다. 반면 가상 머신들이 독립된 환경을 영위하기 위해서는 운영체제 수정이 요구되는 단점을 갖고 있다.

2.2.2 반가상화 (para-virtualization)

기존 하이퍼바이저(hypervisor)를 이용한 전가상화의 경우 binary translation을 이용하여 명령어들을 변환하는 과정에서 성능과 효율성이 떨어지는 취약점을 안고 있기에, 반가상화는 이를 개선하는데 초점을 맞추어 개발된 기법이다[2]. 반가상화의 경우, 가상 머신에서 특권



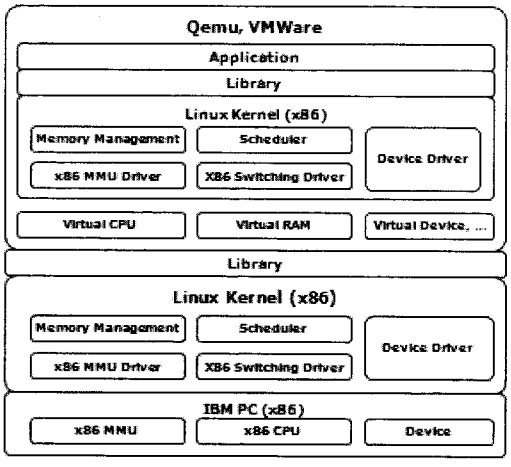
〔그림 1〕 Xen의 구조

명령어가 발생하면 이를 하이퍼바이저가 처리할 수 있는 형태로 변형시켜서 가상머신이 하이퍼바이저의 영역을 침범하지 못하도록 하고 있다. 이를 통해, 반가상화 기법은 가상머신들이 기존 시스템에 가까운 성능을 나타내도록 지원하며, 다양한 운영체제들이 동시에 동작할 수 있도록 설계되었다. 그러나 가상머신이 하이퍼바이저에 맞게 수정되어야 하는 단점이 있다. [그림 1]은 대표적인 반가상화 기법을 사용하는 Xen 구조를 보여준다.

[그림 1]에서 Xen의 구조를 살펴보면, 앞서 설명한 바와 같이 가상머신에는 Xen 위에서 동작할 수 있도록 수정된 커널이 올라가 있다. 그리고 수정된 커널들은 높은 실행권한이 필요한 작업을 수행하기 위해 Xen이 제공하는 하이퍼콜(hypercall)을 이용한다.

2.2.3 전가상화 (full-virtualization)

전가상화 기법은 하이퍼바이저가 가상머신의 명령어들을 일정기준에 따라 묶음 단위로 읽어서 수행하는 binary translation을 지원한다. Binary translation에 의해 가상머신이 동작하면 가상머신 운영체제에 대한 수정이 필요 없어지기 때문에 기존 운영체제를 그대로 가상머신에서 이용할 수 있게 된다. 그리고 가상머신의 명령어들을 하나하나 소프트웨어적으로 에뮬레이션하는 에뮬레이터보다 빠르지만 기존 시스템보다 느린 성능을 보인다. [그림 2]는 전가상화 기법을 사용하는 VMWare의 구조를 보여준다.

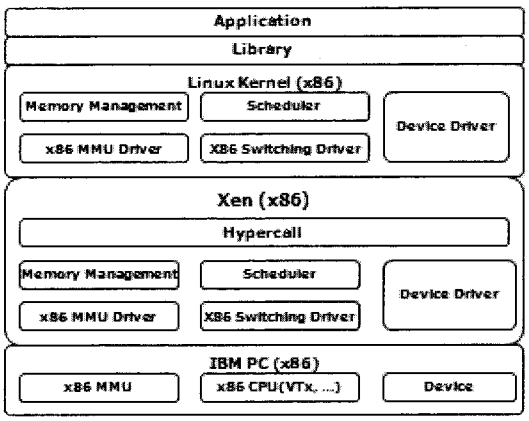


〔그림 2〕 VMWare의 구조

[그림 2]에서와 같이 VMWare는 하드웨어 위에서 수정되지 않은 기본 커널 기반으로 동작하므로 다양한 시스템을 손쉽게 운영할 수 있다는 장점을 갖는다.

2.2.4 가상화를 지원하는 하드웨어 기법

앞서 설명했던 기법들은 가상머신 마다 독립된 환경을 마련해주기 위해서 소프트웨어적 기법들을 적용하여 가상화를 구현하였다면, 지금 설명할 기법은 하드웨어적으로 가상화를 지원하는 기법이다. Intel은 프로세서에 VT-x, VT-i 기능들을 첨부하여 손쉽게 가상화를 실현할 수 있도록 하고 있다[3]. VT-x, VT-i 기능들은 가상머신과 하이퍼바이저의 실행명령 권한 계층을 4단계로 각각 구분하여, 가상화에 걸림돌이 되는 특권 명령어들에 대한 관리를 손쉽게 할 수 있도록 지원한다. 하지만 이 기능을 추가한 것만으로 기존 시스템과 같은 성능을 나타내지 못하기 때문에 다른 가상화기법들이 추가적으로 반영되어야 성능개선을 선보일 것으로 예상되고 있다. [그림 3]은 하드웨어 지원 기능을 이용하는 Xen의 구조를 보여준다.



〔그림 3〕 하드웨어 지원 기반 Xen의 구조

VT-x 기능을 이용하는 Xen의 경우는 [그림 3]과 같이 가상머신위에 수정되지 않은 커널이 올라가 있는 것을 확인할 수 있다. 이처럼 하드웨어 가상화 지원 기능을 이용하면 기존 전가상화 기법보다 개선된 성능으로 다양한 가상 환경을 지원할 수 있게 된다.

2.3 가상화 응용분야

새로운 패러다임으로 각광받고 있는 가상화 기술은 IT 각 분야에서 요구사항에 따라 다양한 용도로 이용되고 있다. 대표적으로 활용되고 있는 분야로는 서버 시스템, 실시간 시스템, 시스템 모니터링, 클라우드 컴퓨팅 등을 들 수 있다. 본 장에서는 각각 분야가 안고 있는 한계점 및 요구사항들을 알아보고 가상화 기술이 어떠한 방식으로 활용되고 있는지 살펴본다.

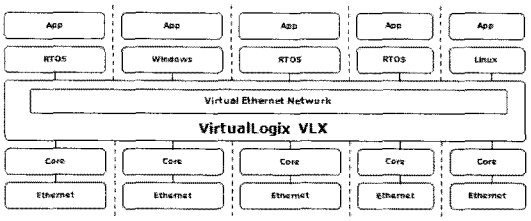
2.3.1 서버 가상화

IT 산업이 발전함에 따라 서버 수가 증가하고 이에 따른 유지 관리비용 또한 많은 부담으로 작용하고 있다. 더욱이 많은 예산을 들여 서버를 구축하여도 활용도가 낮아서 자원의 낭비가 심하고 불필요한 시스템 관리비용이 발생하고 있다.

이처럼 비효율적으로 운영되었던 다수의 서버 컴퓨팅 환경을 가상화 기술을 적용하여 단일 서버로 통합함으로써 시스템 자원의 활용도를 높이고 업무 처리 효율성을 극대화한다. 서로 다른 운영체제와 애플리케이션을 운용했었던 여러 서비스 실행환경을 하나의 플랫폼으로 통합하여 운영할 수 있게 됨으로써 비용 절감, 효율적인 서버 시스템 관리 및 서비스 수준 향상 등과 같은 효과를 얻게 되었다.

2.3.2 실시간 시스템

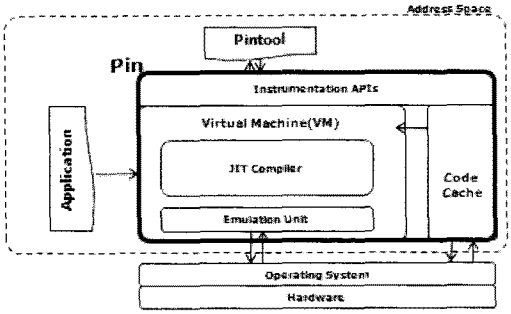
실시간 시스템에서는 가상화 기술을 접목시켜 다수의 가상머신 운영체제들이 싱글코어 혹은 멀티코어 프로세서 상에서 동시에 동작할 수 있도록 하고 있다. 이를 위해 VirtualLogix VLX에서는 가상머신과 하드웨어 사이에 VLX 가상화 계층을 두어서 중요한 자원을 분할 및 공유 할 수 있도록 지원하고 있다([그림 4] 참조). 이 시스템에서는 각 가상머신 운영체제들이 서로 독립적이면서도 실시간 성능 및 높은 서비스 품질을 유지할 수 있도록 보장해 준다.



〔그림 4〕 VirtualLogix VLX의 구조

2.3.3 시스템 모니터링

시스템 모니터링 기법에 가상화 기술을 더하게 되면서 기존과 달리 관리 대상 시스템 환경과 모니터링 시스템 환경을 분리할 수 있게 되었다. 이로 인해, 실시간 시스템 모니터링이 가능해졌으며 모니터링 결과에 따라 실시간으로 대응할 수 있게 되었다. 또한 스냅샷 같은 기술을 이용하여 시간에 따른 파일의 변화를 기록할 수 있게 되었다. 그 밖에도 이전에는 시스템 사용자가 알지 못했던 IO 관련 모든 동작들도 모니터링할 수 있게 되었으며, 시스템 변화 과정을 기록하고 롤백(rollback)할 수 있도록 지원한다. 이처럼 가상화 기술을 통해 시스템 안정성, 디버깅, 악성코드 공격 감시 및 피해복구를 위해 중요한 역할을 할 수 있게 되었다. [그림 5]는 시스템 모니터링을 지원하는 Pin의 구조를 보여준다.



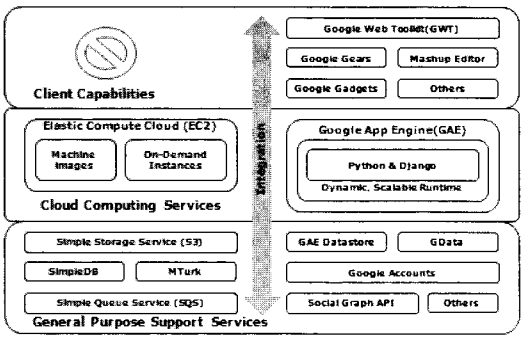
〔그림 5〕 Intel Pin의 구조

Pin은 다양한 API를 통해, 모니터링 대상인 실행파일에서 호출하는 API 들을 확인하거나 레지스터 값을 보여 주는 등 많은 기능을 지원하며 실행파일 분석에 필요한 다양한 정보를 제공해 준다.

2.3.4 클라우드 컴퓨팅

클라우드 컴퓨팅은 가상화 기술을 통해 진정한 의미를 갖게 된다. 가상화 기술을 이용하여 많은 수의 컴퓨

터를 하나의 컴퓨터처럼 묶어주고, 또는 한 대의 슈퍼 컴퓨터를 여러 사용자들이 상호 네트워크를 통해 자신만의 독립된 환경을 구축하여 고성능 컴퓨팅 능력을 이용할 수 있게 된다. 클라우드 컴퓨팅이 가능해지면 장소에 구애받지 않고 컴퓨팅 작업을 수행할 수 있으며, 시스템 관리 부담을 덜고 필요한 자원을 할당하여 사용하므로 자원 활용 효율성을 높일 수 있다. 또한 대부분의 작업을 중앙 시스템 안에서만 처리하게 되어 데이터 보안 관리 능력도 높아진다. [그림 6]은 아마존의 EC2와 구글 GAE의 구조를 비교하여 보여준다.



〔그림 6〕 아마존EC2와 구글GAE의 구조

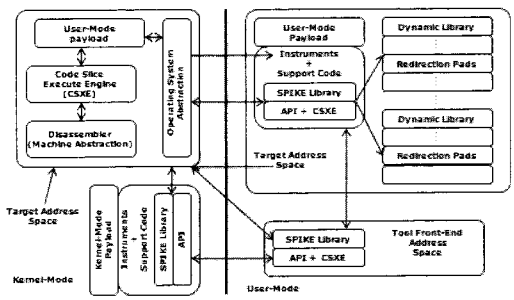
Ⅲ. 시스템 보안을 위한 가상화 기술 활용 사례

3.1 동적코드분석기를 이용한 악성코드 탐지기법

인터넷에서 유통되는 정보가 많아지면서 악의적인 목적을 가진 사용자로부터 개인정보를 보호하는 일이 중요시되고 있다. 악의적인 사용자에 의해 개발 및 배포되고 있는 악성코드는 단순히 네트워크 트래픽을 넘어서 사용자의 비밀정보를 가로채고, 시스템 자원을 인위적으로 조작하는 등 심각한 문제를 발생시킨다. 이에 악성코드를 판단하고 차단하는 일이 현대 컴퓨터 보안에서 가장 큰 이슈가 되고 있다.

악성코드를 탐지하는 방법은 여러 가지가 있다. 대표적으로 악성코드를 정적으로 분석하여 특정 시그너처로 판별해내는 방법이 있다. 이는 악성코드를 실행하지 않고 단순히 읽어서 판단하기 때문에 시스템에 영향을 주지 않고 속도도 빠르지만, 변종 악성코드에 대해 취약점을 보이고 있다. 다음으로는 악성코드를 직접 실행하면서 분석하는 동적 분석 기법이 있다. WIN32 API 후킹으로 악성코드의 시스템 자원 요청 패턴을 분석하여 판단하기 때문에, 정적 분석 기법에 비해 변종 악성코드에 대해 강점을 보이지만, WIN32 API 후킹 모듈 설치 및 악성코드 탐지 후 롤백의 어려움이 있다. 마지막으로 최근 가상화 기술이 발달함에 따라 시스템 자원 요청 패턴 뿐만 아니라, 메모리 접근 패턴 등 보다 정밀한 분석이 가능해졌다. 여기에 샌드박스라는 제한된 환경에서 실행하는 가상화 기법이 추가된다면, 시스템에 영향을 주지 않고 악성코드를 정밀하게 분석하는 것이 가능해진다[4].

SPiKE는 악성코드 탐지를 위한 동적코드분석기이다. 이는 API 후킹, 코드 브레이크포인트, 데이터 추적, DLL 후킹 등 다양한 기능을 제공하고, 최근 악성코드들의 안티-알리어스 기술을 우회하는 기법이 적용되었다. SPiKE의 구조는 [그림 7]과 같다[5].



〔그림 7〕 SPiKE의 구조

위의 [그림 7]은 악성코드 분석을 위한 유저-모드 라이브러리와 커널-모드 드라이버로 구성된 SPiKE의 프레임워크를 보여주고 있다. 유저-모드에서는 SPiKE 관리 툴과 DLL 후킹 등을 지원하고, 커널-모드에서는 스텔스 브레이크포인트, 코드 실행, 메모리 접근 제어 등의 기능을 제공한다.

가상화 기술을 이용한 악성코드 탐지 기법은 가상화 기술의 발전에 힘입어 더욱 발전될 것으로 기대된다. 인텔과 AMD에서 지원하는 하드웨어 가상화 기능은 이런 발전에 더욱 기여할 것이고, 샌드박스를 위한 섀도우 기법 등은 악성코드 실행 환경을 제한함으로써 탐지 후 롤백이 필요없게 되는 등 가상화 기술을 이용한 다양한 악성코드 탐지 기법이 나올 것으로 기대된다.

3.2 가상머신의 시스템 재생 기능을 이용한 악성코드 동작분석

시스템 재생 기능은 임의의 시점의 시스템 동작을 그대로 재현함으로써 다양한 분야에서 시스템 분석을 하는데 용이하다. 제품 개발시 발생 빈도가 낮은 심각한 문제 디버깅, 시스템 취약성 분석, 프로파일링 등에 활용되고, 악성코드에 의해 피해를 입은 시스템을 분석함으로써 악성코드의 정확한 동작 과정을 이해하는데 활용된다.

시스템 재생 기능은 시스템의 결정/비결정 이벤트들을 로깅하므로써 가능해진다. 결정 이벤트는 실행코드와 데이터, 메모리 정보 등 주기억장치와 보조기억장치에 저장되어 있는 정보들이다. 이는 특정 시점의 정보를 알고 있으면 이후의 연속적인 정보들은 결정이 되어있으므로, 주기적인 스냅샷을 통해 로깅이 가능하다. 비결정 이벤트는 인터럽트, 외부 장치 상태 등을 말하며, 이 정보들은 임의의 시점에 임의의 값이 발생하기 때문에 모든 정보들을 로깅하고, 재생시 동일한 시점에 동일한 값을 발생시켜야 한다. 가상화 기술이 적용되기 전에는 결정/비결정 이벤트들을 로깅하고 재생하는데 많은 어려움이 있었기 때문에 명령어 단위의 시스템 재생은 불가능했다. 하지만 가상화 기술의 발전에 따라 에뮬레이션, 코드 변환 기법 등을 통해 명령어 단위의 시스템 로깅 및 재생이 가능해짐에 따라 다양한 시스템들이 제안되고 있다.

ReTrace는 VMWare에 적용된 시스템 재생 기능이다. VMWare는 코드 변환 기법을 사용하기 때문에 명령어 단위의 비결정 이벤트 로깅 및 재생이 가능하지만, 코드 변환에 따른 오버헤드가 크다. 또한, 로깅은 VMWare에서 수행하고, 재생은 Qemu에서 하므로써 로깅에 드는 비용은 최소화하고, 분석은 자유롭게 했지만, 서로 다른 가상 머신에서 로깅 및 재생을 하므로 시스템 간의 호환성 문제가 발생한다. 최근에는 다양한 하드웨어 가상화 지원 기술과 프로세서의 디버깅 기능을 이용하여 에뮬레이션과 코드 변환 기법을 사용하지 않는 시스템 재생 기법이 제안되고 있다[6].

가상화 기술을 이용한 시스템 재생 기능은 악성코드 동작 분석에 있어 새로운 지평을 열 것으로 보인다. 임의의 시점에 발생한 악성코드의 침입을 명령어 단위로 재생/분석하므로써 새로운 악성코드에 대한 백신 개발에 많은 도움을 줄 것으로 기대된다.

3.3 가상머신을 이용한 악성코드 수집 허니팟 운영

허니팟(honeypot)은 자원에 대한 권한이 없는 공격자가 시스템 내의 보호되고 있는 자원들을 침해하고 있는 것처럼 속이거나 침해를 저지하는데 사용된다. 또한 허니팟 안에서 탐지되는 대부분 작업은 의심스러운 행위로 간주되므로, 공격자의 행위를 상세히 분석 및 침입 탐지할 수 있다[7].

그러나 실제로 특정한 호스트를 목표를 하는 공격자의 침입에 대해서 탐지 및 방어하기에는 어려움이 있다. 그래서 웜(worm)이나 봇넷(botnet)과 같은 큰 규모의 무차별적인 공격의 침입에서 유용하게 활용되고 있다.

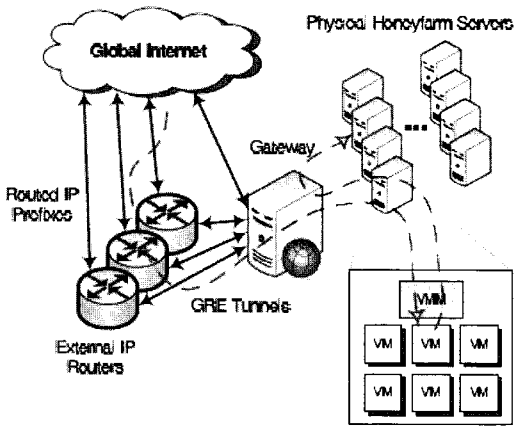
허니팟은 실제 호스트 시스템과 동일하거나 비슷한 실행환경을 제공해야 하므로, 허니팟마다 각각의 서버와 개별적인 IP주소를 필요로 한다. 따라서 규모와 관리에 비용이 많이 든다. 이러한 문제점을 해결하기 위해서 VMWare, Xen, Virtual PC와 같은 가상 머신을 구동하여 하나의 서버에 많은 허니팟을 구성한다.

가상 머신 환경은 허니팟의 구성에 여러 이점을 제공한다. 가상 머신 모니터는 가상 머신의 적재, 저장 등의 관리를 쉽게 한다. 그리고 메모리와 디스크 할당, 시스템 콜의 패턴과 네트워크 흐름 등의 컨텐츠를 제공한다. 또한 실제 서버와 다른 외부 환경에서 실행되기 때문에 커널 루트킷(kernel rootkit)과 같은 숨겨진 멀웨어(malware)도 분석 및 탐지할 수 있다..

허니팟과 관련된 연구들 중에서 캘리포니아 대학의 컴퓨터공학과에서 발표된 "Scalability, Fidelity, and Containment in the Potemkin Virtual Honeyfarm" 논문에서는 가상화 기술을 사용하여 허니팟의 성능을 향상시키고자 하였다[8]. 전통적인 허니팟 시스템은 독립적인 컴퓨터 목적이 아닌, 단지 외부의 어떠한 흐름에 의해서 작업이 부과되었다. 따라서 프로세서나 메모리 자원이 거의 사용되지 않고 낭비되는 문제점이 있었다.

이런 문제점을 해결하기 위해서 구현된 Potemkin 시스템은 특수화된 네트워크 게이트웨이와 가상 머신 모니터인 Xen을 기반으로 구성된다. 특수화된 네트워크 게이트웨이는 IP 주소를 동적으로 관리하며, 허니팟의 활동을 위해서 하나의 이미지를 참조하여 경량의 가상 머신을 생성한다. 따라서 프로세서와 메모리의 낭비를

줄이며, 하나의 물리 서버가 수많은 허니팟을 지원할 수 있도록 하였다. 다음의 [그림 8]은 Petemkin 시스템의 구조도이다.



〔그림 8〕 Petemkin 시스템의 구조

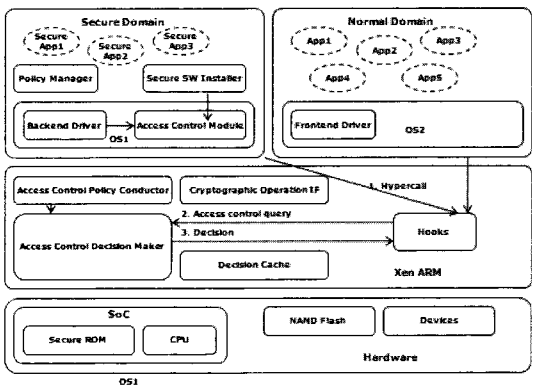
Gateway Routers는 허니팜(Honeyfarm) 서버 내부로 오는 트래픽을 감독, 외부로 나가는 트래픽을 조정, 허니팜 서버들과의 장기적인 자원 관리 이행, 탐지 및 분석과 유저 인터페이스의 기능을 지원한다.

시스템을 구성하기 위에 많은 자원이 요구되어 제약이 많았던 허니팟은 가상화 기술의 발전으로 최근에 더욱 주목 받고 있다. 앞으로 가상화 기술이 더욱 발전됨에 따라 이를 이용한 허니팟은 공격자의 행위를 분석 및 침입 탐지에 많은 도움을 줄 것으로 기대된다.

3.4 가상머신을 이용한 보안 수준에 따른 운영

하드웨어 집적도의 향상으로 인하여 최근의 모바일 디바이스들은 다양한 컴퓨팅 기능을 탑재 하고 있다. 하지만 이로 인하여 시스템의 복잡도가 크게 증가함에 따라, 시스템의 신뢰성을 보장하기가 더욱 어려워지고 있다. 특히 무선 환경에서 동작하는 모바일 디바이스의 경우 악성코드로부터의 공격이나 DOS공격과 같은 다양한 보안 취약점을 안고 있기 때문에, 전자 상거래와 같이 특정 서비스를 이용할 때에는 위험성을 갖고 있다. 이와 관련한 많은 연구 중에서 삼성전자의 "Xen on ARM"은 가상화 기술을 통하여 위의 문제를 해결하고자 하였다.

"Xen on ARM"은 삼성에서 연구 중인 오픈소스 프로젝트로써 Xen의 반가상화 기술을 이용하여 모바일 환경을 보안 수준에 따라 격리 운영하는 기술이다[9]. 즉, 가상 머신을 보안 레벨에 따라 보안도메인과 일반도메인으로 나누어 관리한다. 이를 위해서 ARM프로세서를 가상화하여 동작 모드별로 권한을 나누었으며, 메모리 가상화를 통해 도메인 간의 메모리 보호를 제공한다. 또한 입출력 가상화를 통해 신뢰성 있는 입출력을 제공하고 있다. 다음의 [그림 9]는 "Xen on ARM"의 시스템 구조를 보여준다.



〔그림 9〕 Xen On ARM의 구조

위의 시스템 구조는 다음의 세 가지 보안 기능을 제공하고 있다.

- Secure Domain

전자 상거래, 인터넷뱅킹 등의 프로그램 운영 시 다른 사용자의 임의로 접근을 막는다.

- Secure Boot

Secure Rom을 이용하여 부트스트랩(bootstrap) 시 보안 기능을 제공한다.

- Access Control

하이퍼바이저에 Access Control Module(ACM) 을 적용하여, 도메인으로 부터의 모든 입출력 요 청을 관리한다.

Xen on ARM은 더 향상된 보안 기능을 제공하기 위해 보안 통신 프로토콜을 개발 중에 있다. 또한 보안 기능을 제공하는 과정에서 발생하는 오버헤드를 줄이기

위해 성능 분석과 최적화 작업도 진행 중이다. 이와 같은 가상화 기술 연구를 통해서 앞으로의 모바일 디바이스에서는 보다 향상된 보안 기능을 제공 할 것으로 기대된다.

Ⅳ. 결 론

컴퓨터 시스템이 사회 전반의 기본 전제 조건으로 자리매김함에 따라 시스템 보안의 중요성은 날로 높아지고 있다. 시스템 보안 업계에서는 한층 향상된 신뢰 서비스를 안정적으로 제공하기 위해 다양한 기법들이 제안되고 있다. 그 가운데 가상화 기술은 기존의 시스템 보안 수준을 한 단계 높이며, 핵심 역할을 담당할 것으로 기대를 모으고 있다. 이를 위해서는 보다 안정된 가상환경 구축하고, 실용성을 높일 수 있는 성능개선 기법들에 대한 연구가 필연적으로 필요할 것이다. 또한, 가상화 기술이 보편화 되면서 지금까지와는 다른 새로운 유형의 보안위협들이 파생될 것으로 예상됨에 따라, 이에 대한 대책도 시급히 해결해야할 과제로 남아 있다.

참고문헌

[1] VMware, Inc. "Understanding Full Virtualization, Paravirtualization, and Hardware Assist," http://www.vmare.com, 2007.

[2] P. Barham, B. Dragovic, K. Fraser, S. Hand, T. Harris, A. Ho, R. Neugebauer, I. Pratt, and A. Warfield, "Xen and the Art of Virtualization," Proc. of the 19th ACM Symposium on SOSP, 2003.

[3] G. Neiger, A. Santoni, F. Leung, D. Rodgers, and R. Uhlig, "Intel Virtualization Technology: Hardware Support for Efficient Processor Virtualization," Intel Technology Journal, pp. 167-177, 2006.

[4] S. Sidiroglus, J. Ioannidis, A. Keromytis, and S. Stolfo, "An Email Worm Vaccine Architecture," Proc. of the 1st Information Security Practice and Experience Conference(IPSEC), 2005.

[5] A. Vasudevan, R. Yerraballi, "SPiKE: engineering malware analysis tools using unobtrusive binary-instrumentation," Proc. of the 29th ACM International Conference, Vol. 171, 2006.

[6] M. Xu, V. Malyugin, J. Sheldon, G. Venkitachalam, and B. Weissman, "ReTrace: Collecting Execution Trace with Virtual Machine Deterministic Replay," Proc. of 2007 Workshop on Modeling, Benchmarking and Simulation, 2007.

[7] J. K. Jones, G. W. Romney, "Honeynets: an educational resource for IT security," Proc. of the 5th conference on Information technology education, 2004.

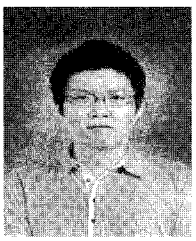
[8] M. Vrable, J. Ma, J. Chen, D. Moore, E. Vandekieft, A. C. Snoeren, G. M. Voelker, S. Savage, "Scalability, fidelity, and containment in the potemkin virtual honeyfarm," Proc. of the twentieth ACM symposium on Operating systems principles. 2005.

[9] J. Hwang, S. Suh, S. Heo, C. Park, J. Ryu, S. Park, C. Kim, "Xen on ARM: System Virtualization using Xen Hypervisor for ARM-based Secure Mobile Phones," In IEEE CCNC, pp. 257-261, Jan 2008.

〈著者紹介〉

김 인 혁 (Inhyuk Kim)

2006년 2월: 성균관대학교 전자전기컴퓨터공학과 졸업
2008년~현재: 성균관대학교 전자전기컴퓨터공학과 석사과정
<관심분야> 운영체제, 가상화, 정보보호

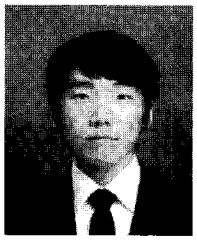


김 태 형 (Taehyoung Kim)

2007년 8월: 성균관대학교 컴퓨터공학과 졸업
2009년 2월: 성균관대학교 전자전기컴퓨터공학과 석사
2009년~현재: 성균관대학교 전자전기컴퓨터공학과 박사과정
<관심분야> 운영체제, 가상화, 정보보호

김 정 한 (Junghan Kim)

2008년 2월: 세종대학교 컴퓨터소프트웨어학과 졸업
2008년~현재: 성균관대학교 전자전기컴퓨터공학과 석사과정
<관심분야> 운영체제, 가상화, 정보보호



임 병 홍 (Byounghong Lim)

2009년 2월: 충남대학교 컴퓨터학과 졸업
2009년~현재: 성균관대학교 임베디드소프트웨어학과 석사과정
<관심분야> 운영체제, 가상화, 정보보호

엄 영 익 (Young Ik Eom)

종신회원

1983년 2월: 서울대학교 계산통계학과 졸업
1985년 2월: 서울대학교 전산과학과 석사
1991년 8월: 서울대학교 전산 과학과 박사
2000년 9월~2001년 8월: Dept. of Info. and Comm. Science at UCI 방문교수
1993년 3월~현재: 성균관대학교 정보통신공학부 교수
<관심분야> 시스템 소프트웨어, 미들웨어, 가상화, 시스템 보안