온라인 게임 최신 해킹기법 및 대응방안

2008. 9. 24

A3Security
정대근 선임컨설턴트

A3SECURITY

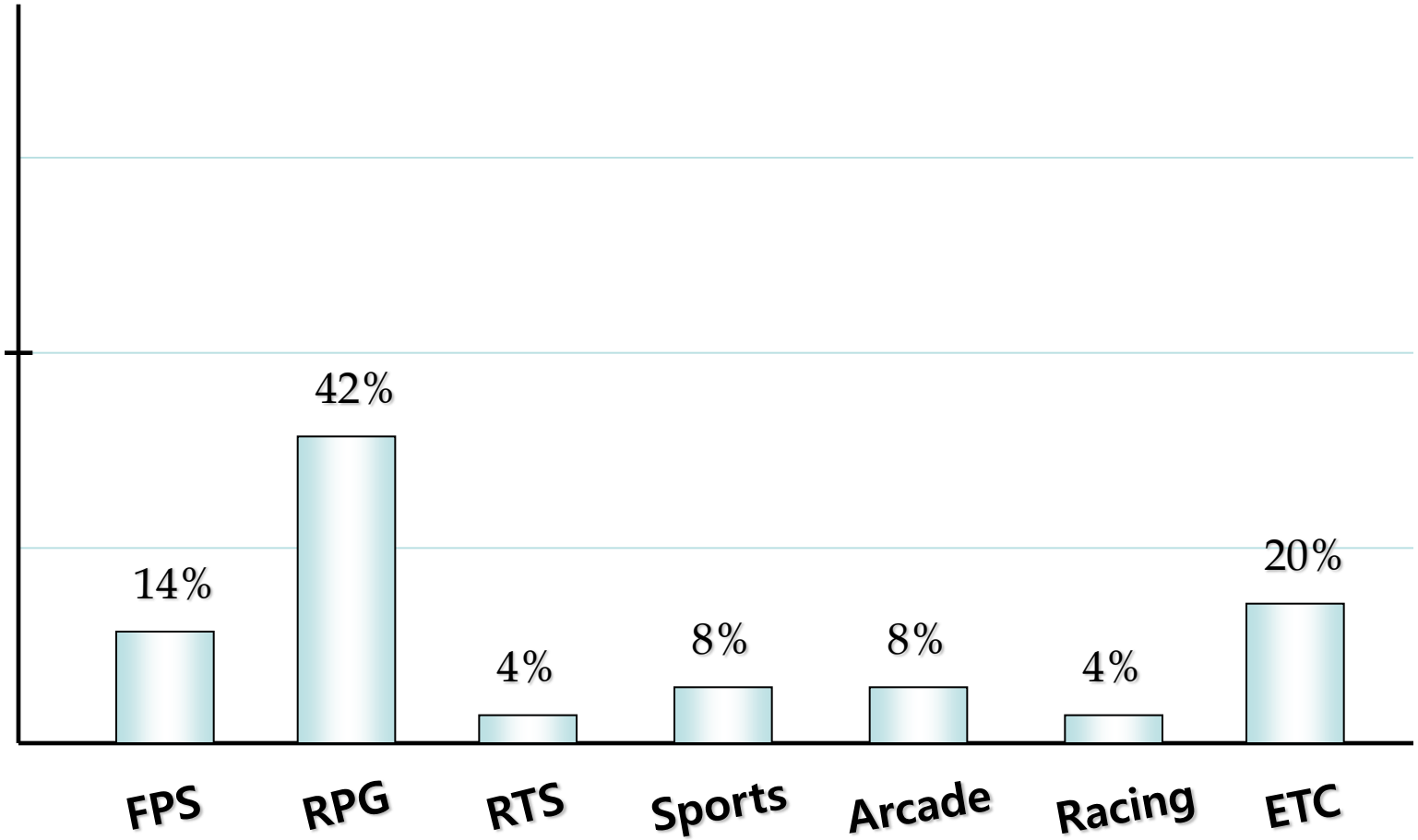
목 차

Case #1. DLL Injection

Case #2. Virtual Machine

- 최근 온라인 게임 동향

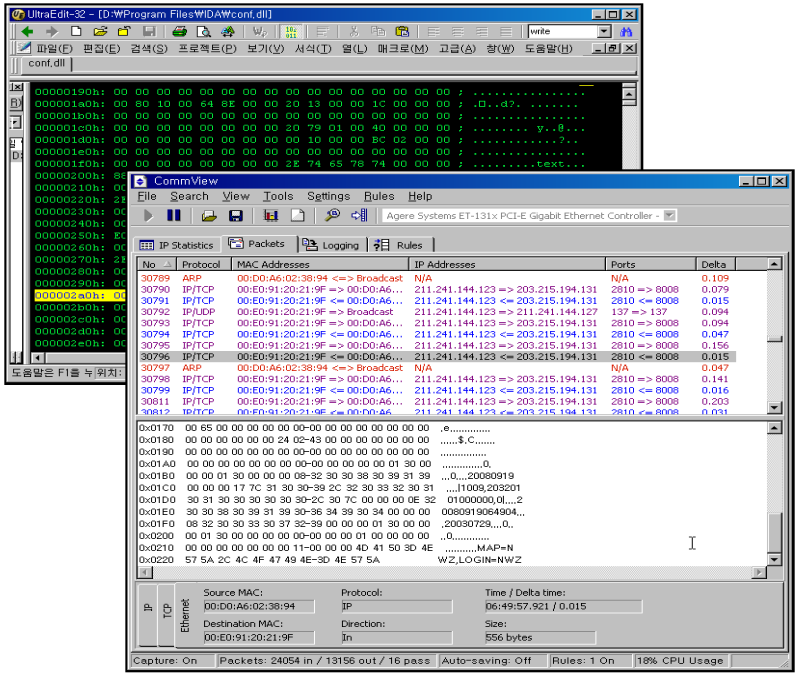
2008. 09. 17 기준



A3SECURITY

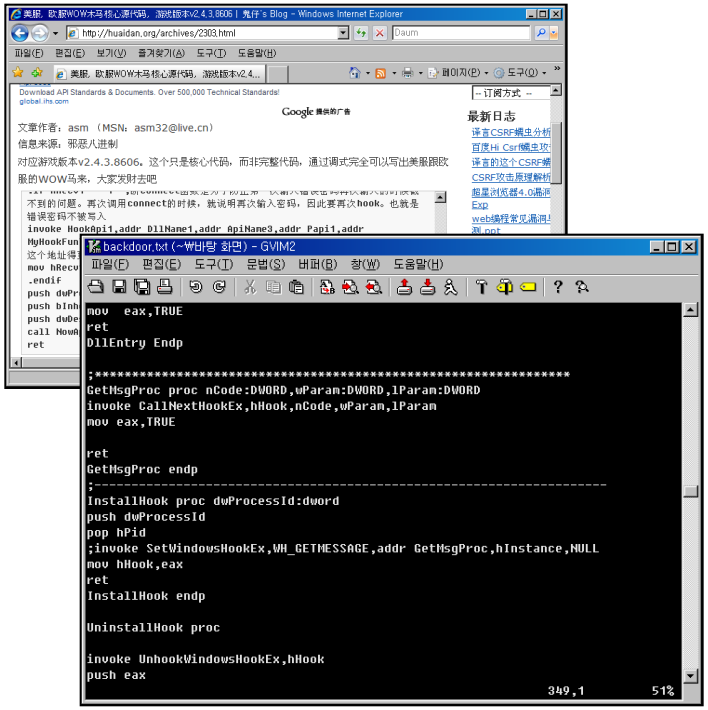
고전적인 온라인 게임 해킹 기법

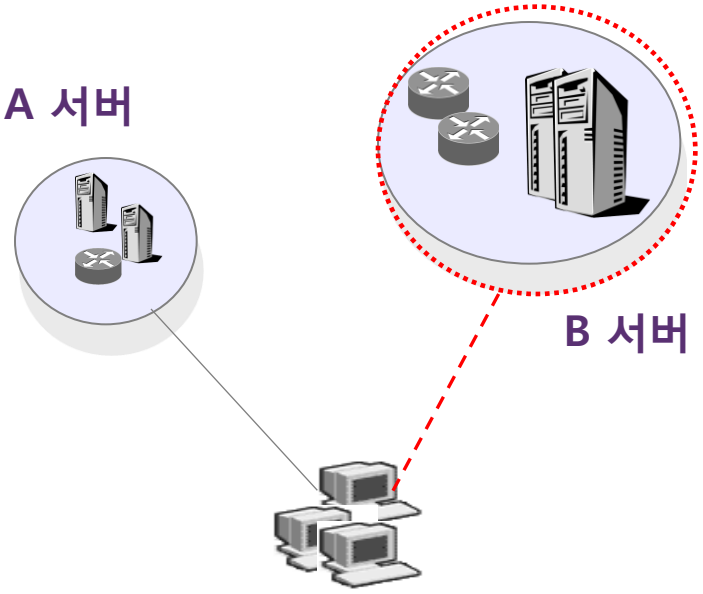
- Auto Play
- Speed Hack
- Bot Attack
- Data/Packet Modification



A3SECURITY

Case #1. DLL Injection





허용하지 않은 불법 서버 접근

Case #1. DLL Injection

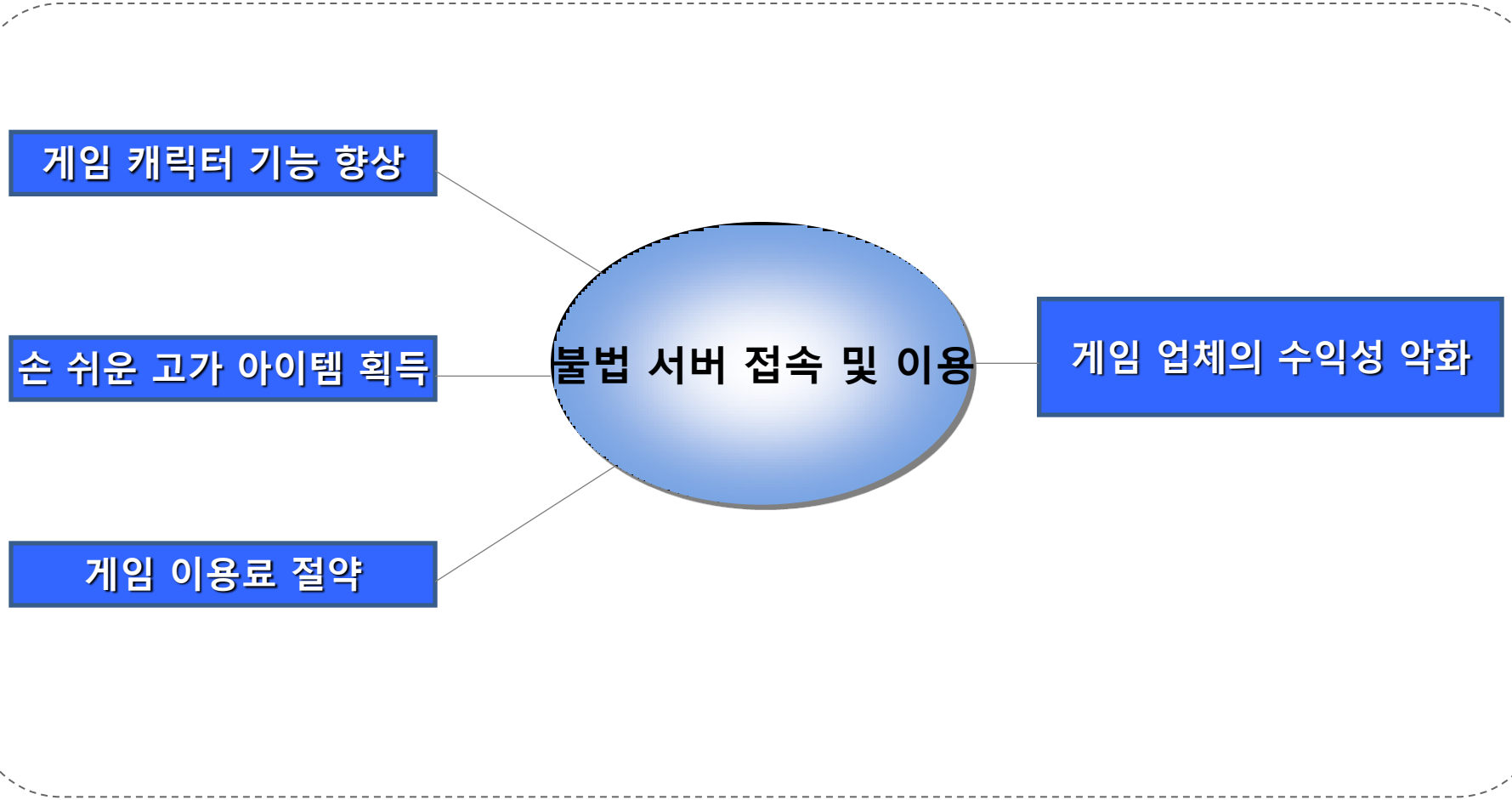
```
invoke ExtractFileName,addr szText
invoke wsprintf,addr szFileName,CTXT("%s"),eax
invoke EncryptString,addr szWowprocess ;解密game.exe????
invoke CompareString,LOCALE_USER_DEFAULT, NORM_IGNORECASE,addr szFileName, -1,add
s, -1;如果是wow, ?hook
.if eax == 2
invoke OutputDebugString,CTXT("Found...")
invoke LoadModuleEx,NULL
mov hExeModule,eax
invoke GetModuleImageSize,hExeModule
mov dwModuleSize,eax
invoke BytePos,hExeModule, dwModuleSize,addr szUserPassRealCode,sizeof szUserPass
ut ;搜索特征?的位置
```

프로세스 영역에
Hooking 설정

```
invoke DeleteFile,addr TempPath
invoke _WriteFile,addr $___GAMENAME___user,addr TempPath
mov        @dwSize,sizeof @szValue
invoke         _RegQueryValue,CTXT("
nstallPath"),addr @szValue,addr @dwSize,CTXT("REG_SZ") ;得到$___GAMENAME___安?路?
invoke lstrcat,addr @szValue,CTXT("
")
invoke lstrcat,addr @szValue,addr $___GAMENAME___user
invoke _FindFileToDel,addr @szValue ;?空?前用?下的所有文件?子目?
invoke         WriteProcessMemory,WProcess,hUserPassRealCode,addr szJmp,7,addr dwTem
MOV hRecv1,1
invoke WriteProcessMemory,WProcess,hSend,addr szBakSend,20,NULL  ;恢?send函?20字?
```

원하는 정보를
Process Memory에
Overwrite

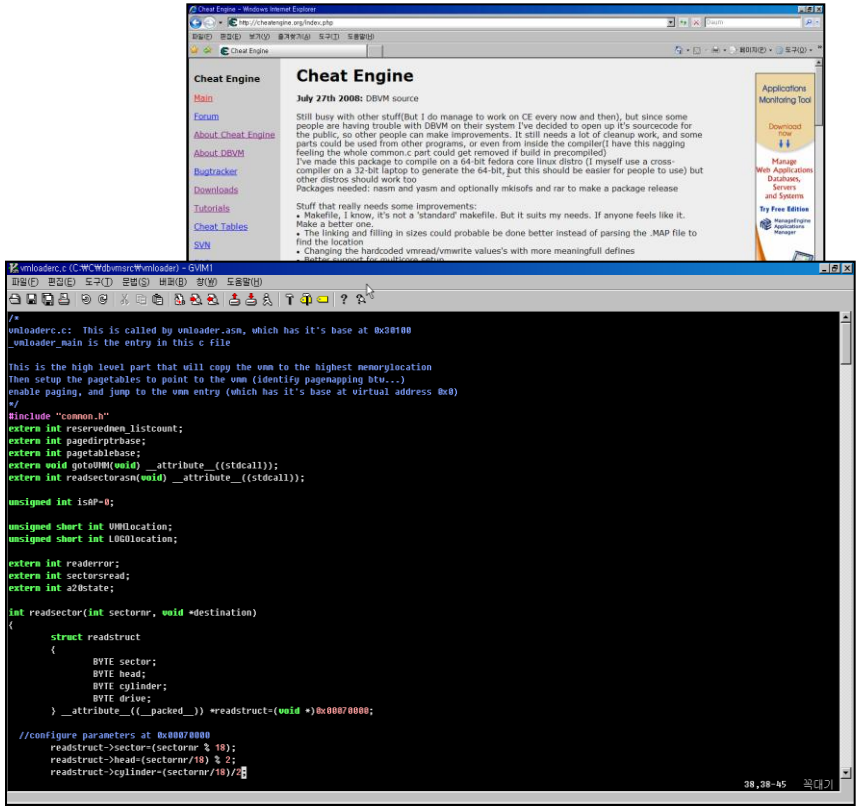
Case #1. DLL Injection



Case #2. Virtual Machine

게임 디버깅 프로그램의 가상화 활용

- CPU 에서 지원해야 가능
- System Kernel 모드에 쉽게 접근
- 기존 detection code는 무용지물
- 소스코드 공개



Case #2. Virtual Machine

```
printf("Opening and reading vmm.bin...");
if (stat("vmm.bin",&tempstat))
{
  printf("File can't be found¥n");
  return 1;
}
vmm_size=tempstat.st_size;
vmm=malloc(vmm_size);
fpVmm=fopen("vmm.bin","r");
fread(vmm,tempstat.st_size,1,fpVmm);
printf("done¥n");

printf("Creating vmdisk.img...¥n");
fpDisk=fopen("vmdisk.img","w");
if (fpDisk==NULL)
{
  printf("Failed creating file¥n");
  return 1;
}

printf("Writing VMMlocation in bootsector¥n");
VMMlocation=2+(1+vmloader_size/512);
*(unsigned short int *)&bootloader[0x3d]=VMMlocation;

fwrite(bootloader,bootloader_size,1,fpDisk);
fwrite(vmloader,vmloader_size,1,fpDisk);

//now seek to the VMM startsector
fseek(fpDisk, VMMlocation*512, SEEK_SET); //go to next sector pos
fwrite(vmm,vmm_size,1,fpDisk);

//fill till dividable by 512
bzero(sector,512);
fwrite(sector,512-((VMMlocation*512+vmm_size) % 512),1,fpDisk);
```

```
fpDisk=fopen("vmdisk.img","w");
if (fpDisk==NULL)
{
  printf("Failed creating file¥n");
  return 1;
}

printf("Writing VMMlocation in bootsector¥n");
VMMlocation=2+(1+vmloader_size/512);
*(unsigned short int *)&bootloader[0x3d]=VMMlocation;

fwrite(bootloader,bootloader_size,1,fpDisk);
fwrite(vmloader,vmloader_size,1,fpDisk);
```

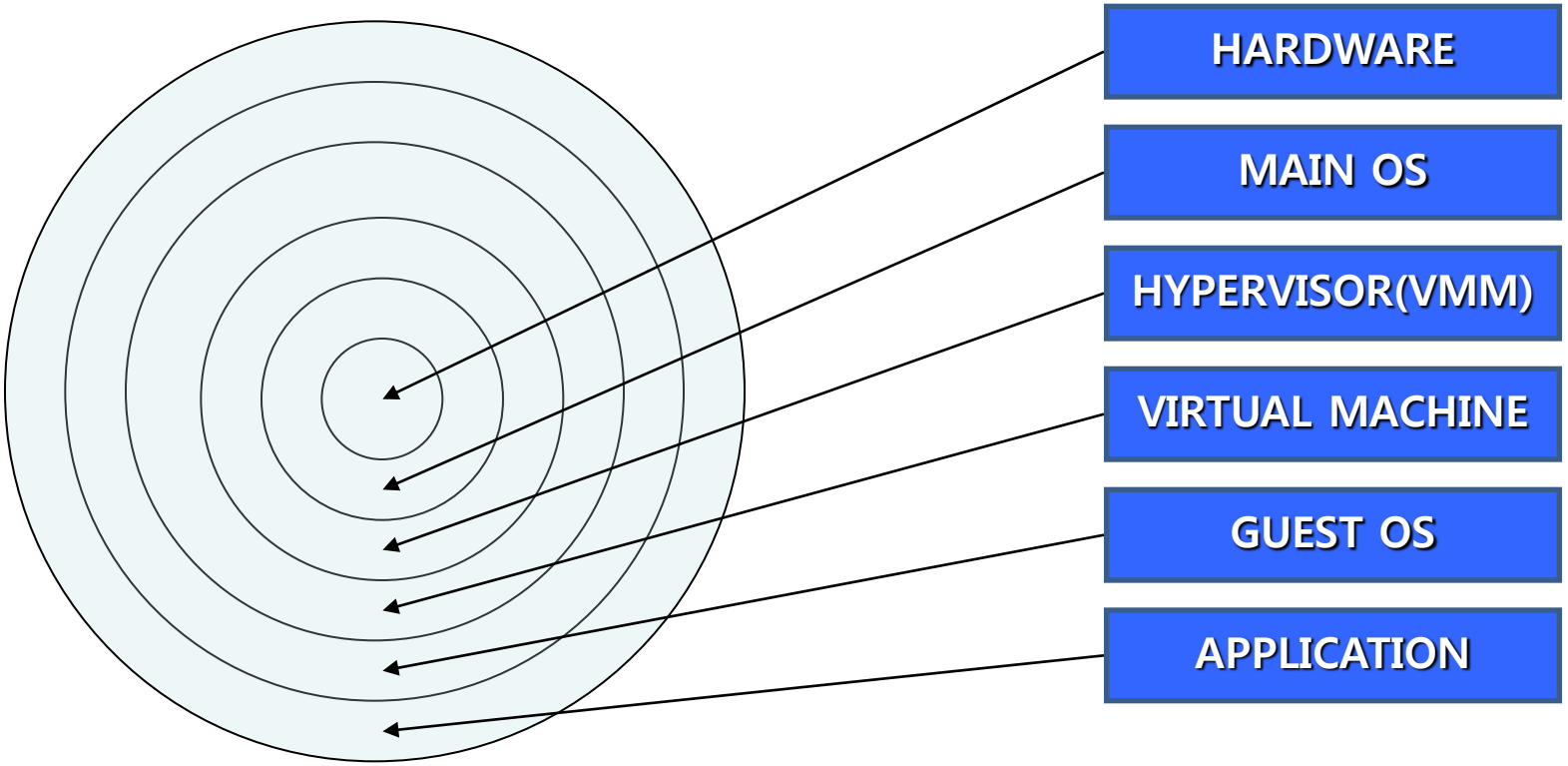
가상의 부트섹터 이미지를
만들어 VM 환경을 조성

Case #2. Virtual Machine

```
UINT64 VirtualToPhysical(UINT64 address)
{
  /* pagedirvirtual contains the virtual address where the pagetable is stored */

  unsigned int PML4=address >> 39 & 0x1ff;
  unsigned int Dirptr=address >> 30 & 0x1ff;
  unsigned int Dir=(address >> 21) & 0x1ff;
  unsigned int Offset=address & 0x1fffff;
  UINT64 startofpage;

  //this design doesn't use more than 4GB ram addressing, even though it is 64, bit, so only the level0 pagedirptr is enough


  PPDE2MB_PAE usedpagedir=(PPDE2MB_PAE)((UINT64)pagedirvirtual+Dirptr*0x1000);
  if (usedpagedir[Dir].P==1)
  {
    return (usedpagedir[Dir].PFN << 13)+Offset;
  }
  else
  {

    sendstringf("PML4=%d Dirptr=%d Dir=%d Offset=%x\n\r",PML4, Dirptr, Dir, Offset);
    sendstringf("pagedirvirtual=%6\n\r",(UINT64)pagedirvirtual);
    sendstringf("usedpagedir=%6\n\r",(UINT64)usedpagedir);
    sendstringf("usedpagedir[Dir].P==0\n\r");
    sendstringf("&usedpagedir[Dir]==%6\n\r",(UINT64)&usedpagedir[Dir]);


    return 0xffffffffffffffff;
  }
}
```

가상 메모리 주소를 VM 환경 하에서 사용 가능한 물리적 메모리 주소로 변경을 하는 코드 루틴

Case #2. Virtual Machine



VM 계층 구조도

온라인 게임 해킹에 대한 대응방안

- CreateThread detection

BaseThreadStartThunk()

lpThreadFunc

- LoadLibrary
- GetProcAddress
- FreeLibrary

※ CreateRemoteThread

- VMM Environment Detection

Detection Program Execution

감사합니다