反虚拟机技术

一、初级检测手段

- 在进程列表、服务列表、注册表中搜索`VMware`字符串。

  `net start | findstr VMware`

  `SYSTEM\CurrentControlSet\Control\DeviceClasses`

- 查看MAC地址是否以固定数值开头，例如`00:0c:29`。查询函数`GetAdaptersInfo`
- 查看硬件的版本

绕过方法

- 卸载VMware Tools或者停止该服务`net stop "VMware Tools Service"`
- 在IDA中阻止恶意代码的检测流程。找到相关字符串的引用位置，修复。
  - 在调试过程中让相关的跳转指令无法实现
  - 使用十六进制编辑器修改比较的字符串`VMware`
  - 卸载软件/停止服务

二、高级检测手段

- Red Pill：使用`sidt`指令检查IDTR寄存器(6字节)的第六个字节(内存基地址)是否是虚拟机使用的`0xFF`。

  ```
  sidt    fword ptr [eax]
  mov     al, [eax+5]
  cmp     al, 0FFh
  jnz     short loc_401E19
  ```

  绕过：使用多核处理器，用NOP指令替换这部分指令

- No Pill：使用`sldt`指令检查LDTR寄存器中的内容是否为零，不为零说明使用的虚拟机。

  绕过：`VM->Settings->Processors->Disable Acceleration`

  绕过后检测：使用`smsw`指令检查返回值的高位比特

- 使用`in`指令检测I/O接口`'VX'`

  ```
  004014FE        mov     eax, 'VMXh'          ; magic number
  00401503        mov     ebx, [ebp+var_1C]    ; return information
  00401506        mov     ecx, 0xA             ; operation"get VMware version
  type"
                                               ; 0x14: get the memory size
  00401509        mov     dx, 'VX'
  0040150E        in      eax, dx
  ...
  00401541        cmp     ebx, 'VMXh'
  00401546        jnz     short loc_40155C
  ```

绕过：NOP-out `in`指令，修改跳转指令

- 使用`str`指令检查task register(TR)寄存器中保存的segment selector，它指向了当前任务的task state segment(TSS)，在虚拟机和正常主机上该指令返回的数值不同。

  绕过：使用多核处理器

```
str word ptr [ebp+var_4]
movzx   ecx, [ebp+var_4]
test    ecx, ecx            ; 检查第一个字节是不是0
jnz short loc_401276
movzx   edx, [ebp+var_3]
cmp     edx, 40h            ; 检查第二个字节是不是40h
jnz short loc_401276        ; 如果前两个字节是0040h，就说明是虚拟机
```

- 使用工具：ScoopyNG进行检测，原理和上述相同
- 利用虚拟机的漏洞进行逃逸

总结常用检测指令

`sidt`、`sgdt`、`sldt`、`smsw`、`str`、`in`、`cpuid`