网络攻击与防御

刘智
西南石油大学 计算机科学学院
zhi.liu@swpu.edu.cn

本节内容与目标

- 理解恶意代码不同分类方法、命名方式
- 了解恶意代码发展趋势
- 熟悉计算机病毒工作原理

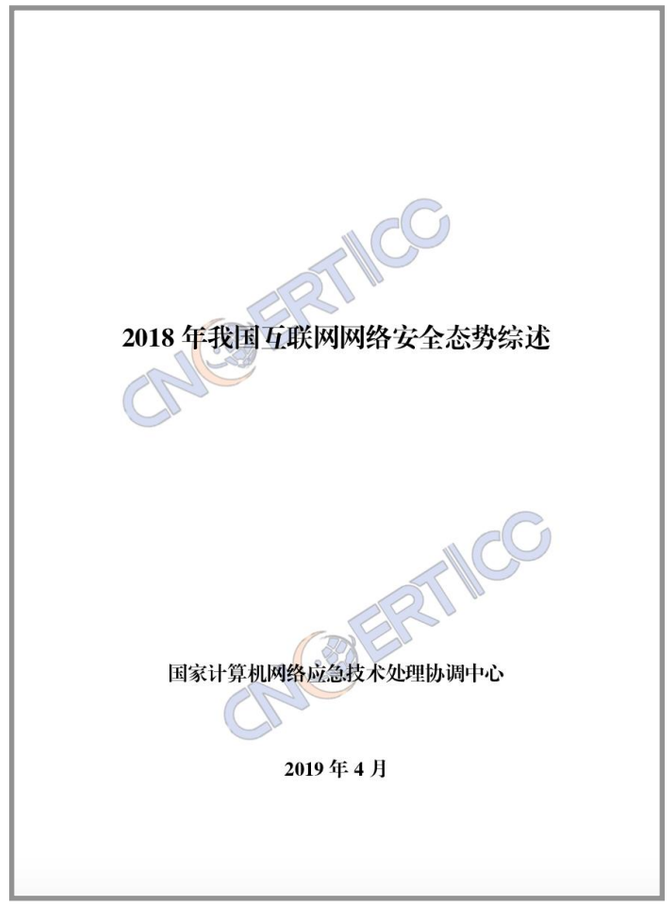
恶意代码定义

- “运行在目标计算机上，使系统按照攻击者意愿执行的一组指令” – “Malware: Fighting malicious code” by Ed Skoudis and Lenny Zeltser
- “恶意代码是指故意编制或设置的、对网络或系统会产生威胁或潜在威胁的计算机代码” – 百度百科

Malware = Malicious software

相关数据

- CNCERT捕获**1亿个**恶意代码
- 境内遭到攻击的IP达**5946万个**
- 境内感染恶意代码主机数**655万台**
- 4.9万个恶意程序控制了国内**526万台主机**

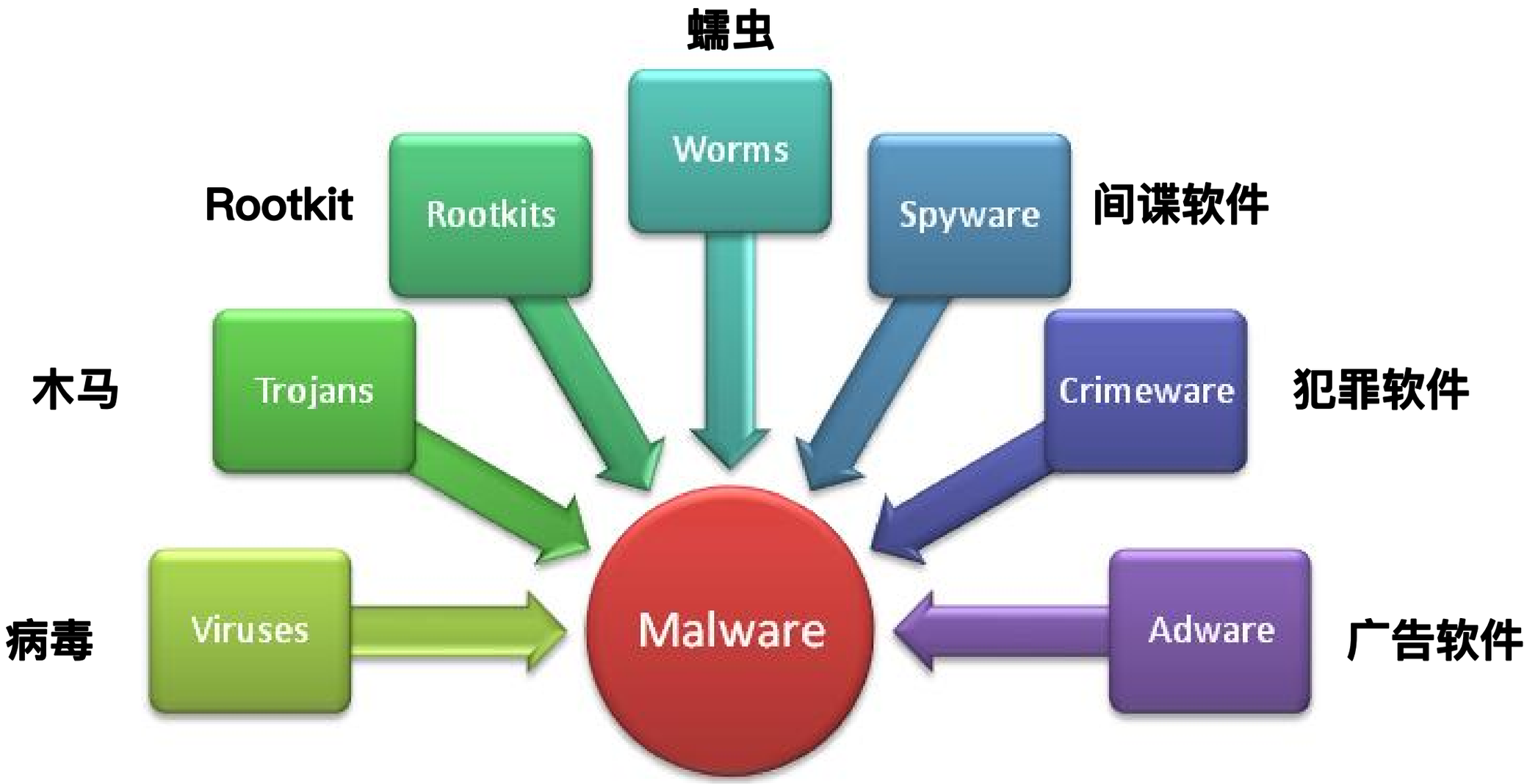


恶意代码表现

- 机器运行速度变慢
- 网络变慢
- 异常网络访问
- 注册表修改
- 突然死机或重启
- 文件被修改

....

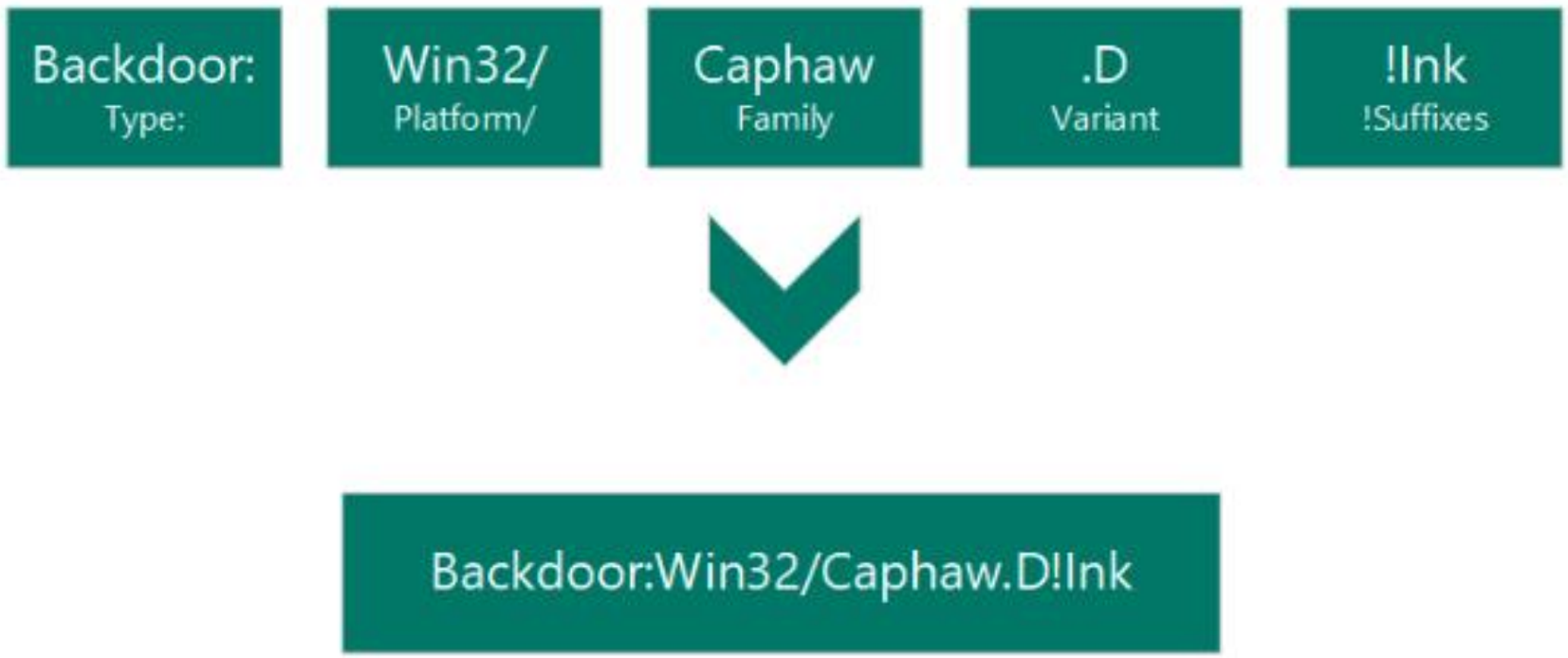
恶意代码分类



恶意代码定义及特征

类型	定义	典型特征
病毒(virus)	植入到计算机程序中的非法指令或者程序， 破坏计算机功能和数据，具备自我复制能力	感染性(本地)
蠕虫(worm)	通过网络自我复制进行破坏的程序	感染性(网络)
木马(trojan)	伪装成正常代码进行破坏的程序	欺骗性、信息窃取
Rootkit	工作在内核态的恶意代码	隐蔽性(内核)
间谍软件(spyware)	偷偷安装在电脑上并收集受害者的敏感信息的软件。	信息窃取
广告软(adware)	未经用户允许下载安装并弹出广告的恶意软件	
Bot	作为僵尸网络的组成部分，在未经用户许可的情况 下进行各种方式的破坏或与master通信	隐蔽性、C&C通信
勒索软件	用户劫持用户资产或资源并以此向用户勒索软件的 一种软件	文件加密

恶意代码命名规范



类型: Backdoor, Ransome, Trojan, Virus, Worm…

平台: DOS, Win32, Win64, Linux…

后缀: exe, dll, ….

恶意代码主要传播途径

- 软盘
- 光盘
- USB
- 互联网
  - 邮箱、IM、文件下载

典型恶意代码

- 1983年，Fred Cohen研制第一款真实恶意代码
- 1986年，第一个感染PC的恶意代码Brain
- 1988年，Morris蠕虫
- 1992年，杀毒软件兴起
- 1998年，CIH病毒
- 2000年后： CodeRed(2001)，震荡波(2004)
- 2010年：Stuxnet
- 2016年后：物联网恶意代码、勒索软件

产生阶段的恶意代码

- 典型恶意代码：Morris蠕虫
- 主要特征
  - 攻击的目标单一，传染磁盘引导扇区或可执行文件；
  - 通过截获系统中断向量的方式监视系统的运行状态，并在一定的条件下对特定目标进行传染；
  - 传染目标以后有明显特征；
  - 不具有自我保护的措施，容易被分析和解剖。

综合阶段的恶意代码

- 主要特征
  - 目标趋于混合型，可同时传染磁盘引导扇区和可执行文件；
  - 以更为隐蔽的方法驻留内存和传染目标；
  - 开始采取自我保护措施，增加分析和查杀的难度；
  - 变种多，更新快，隐蔽性更强。

成熟发展阶段的恶意代码

- 主要特征
  - 具有**多态性或“自我变形”**能力，是恶意代码的成熟发展阶段。
  - **多态**：通过加密等方式改变“形态”绕过监测
  - **自变形**：动态修改代码

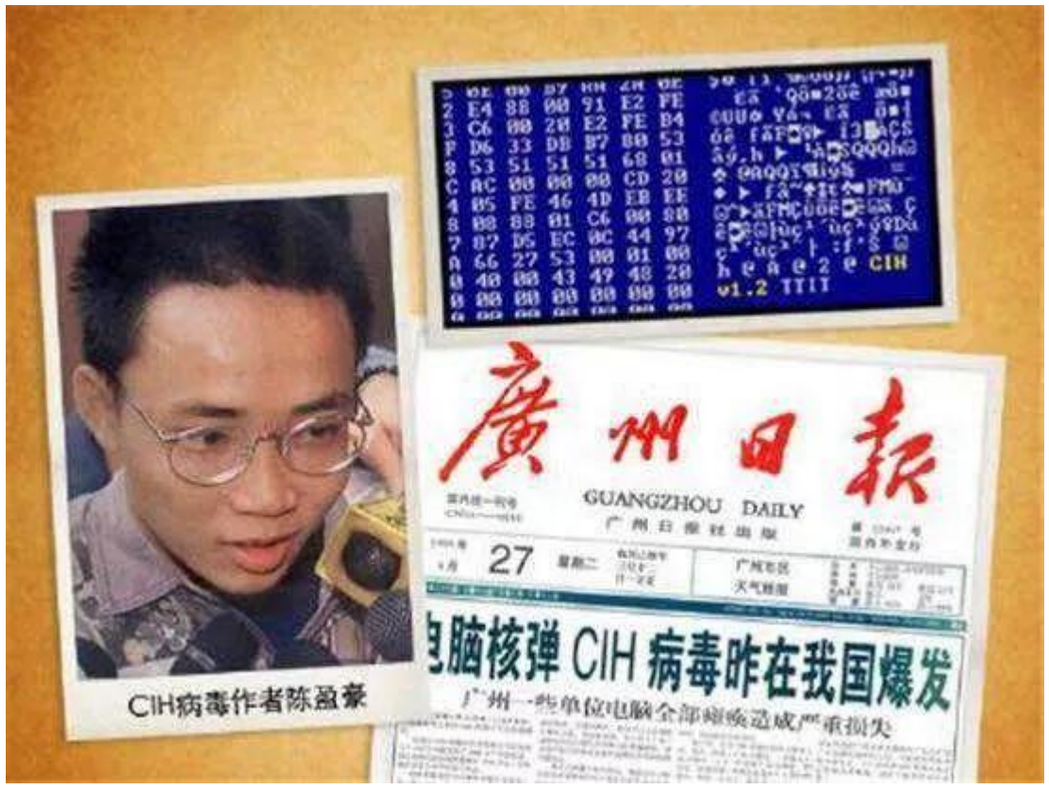
互联网爆发阶段的恶意代码

- 主要特征
  - 类型多样化，不再局限于可执行文件；
  - 更多以互联网作为传播渠道；
  - 攻击目标突破软件限制；
  - 变种多，更新快，隐蔽性更强，破坏性更大。

恶意代码发展趋势

- 更加复杂
- 更加智能
- 平台化

计算机病毒



计算机病毒定义

计算机病毒是一种计算机程序，它递归地、明确地复制自已或其演化体 。

——美.Cohen

编制或者在计算机程序中插入的破坏计算机功能或者破坏数据，影响计算机使用并且能够自我复制的一组计算机指令或者程序代码。

–《计算机信息系统安全保护条例》

计算机病毒特点

- 可执行性
- 非授权性
- 自我复制性
- 破坏性

病毒相关概念

- 计算机病毒与生物病毒有相似性
  - 自身病毒体
  - 宿主(host)

- 病毒感染
  - 将病毒代码嵌入到宿主程序中
  - 宿主为病毒提供执行环境。

可执行程序： **exe**, **dll**, sys等文件
Office文件
….

病毒分类(按感染文件类型)

- **PE病毒**
  - 感染Windows**平台exe、.dll等**PE文件格式的病毒，数量多、破坏性大。
- **宏病毒**
  - 恶意代码以宏指令形式潜伏在Office文档中，并随Office软件应用而传播广泛。

病毒分类(按运行平台)

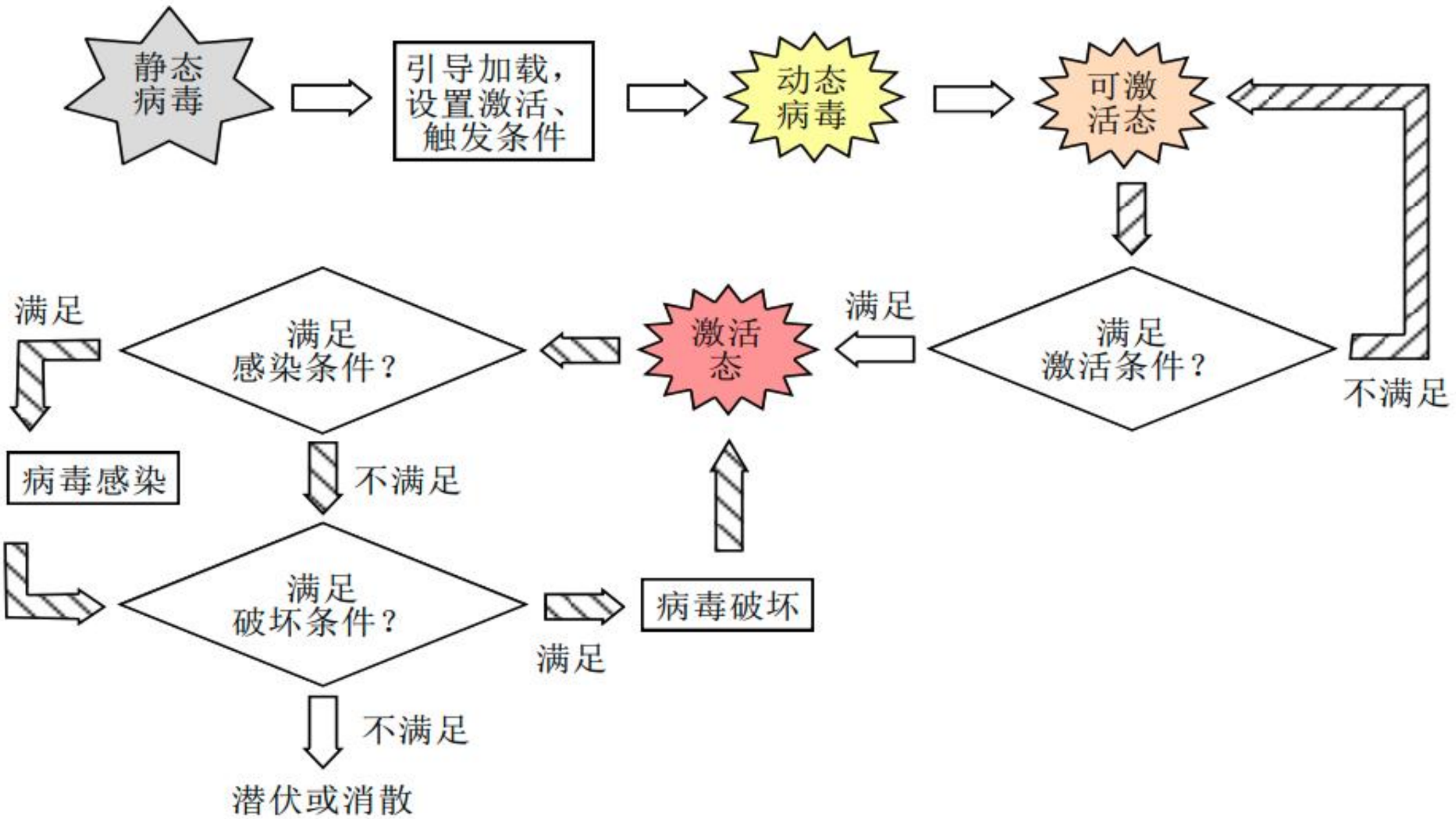
- DOS病毒
- **Windows病毒**
- Linux病毒
- Mac(OSX, iOS)病毒
- 安卓病毒

病毒危害

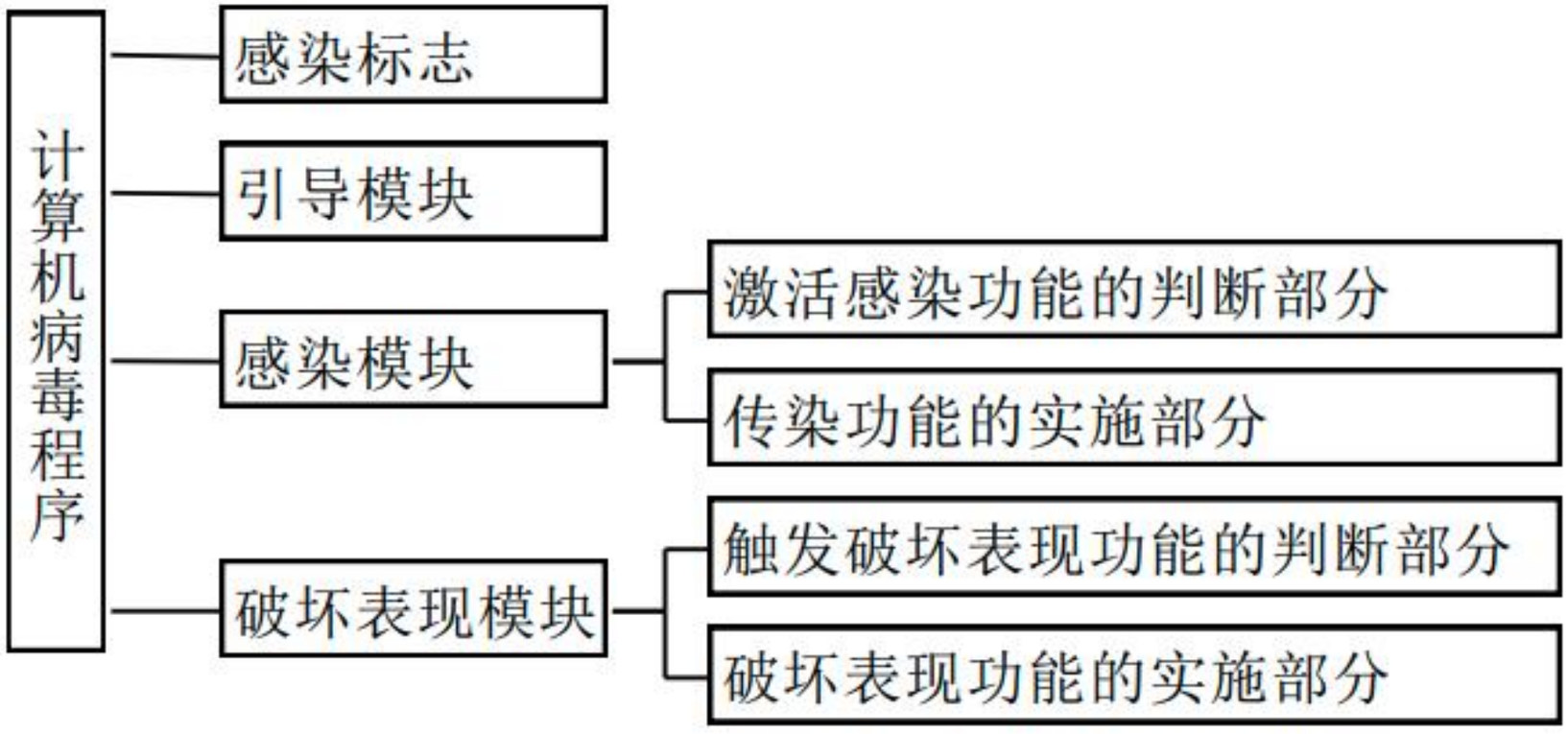
- **破坏磁盘引导主扇区、FAT表**
- 破坏磁盘文件
- 使系统运行速度变慢

病毒工作主要流程

病毒工作详细流程



病毒功能结构



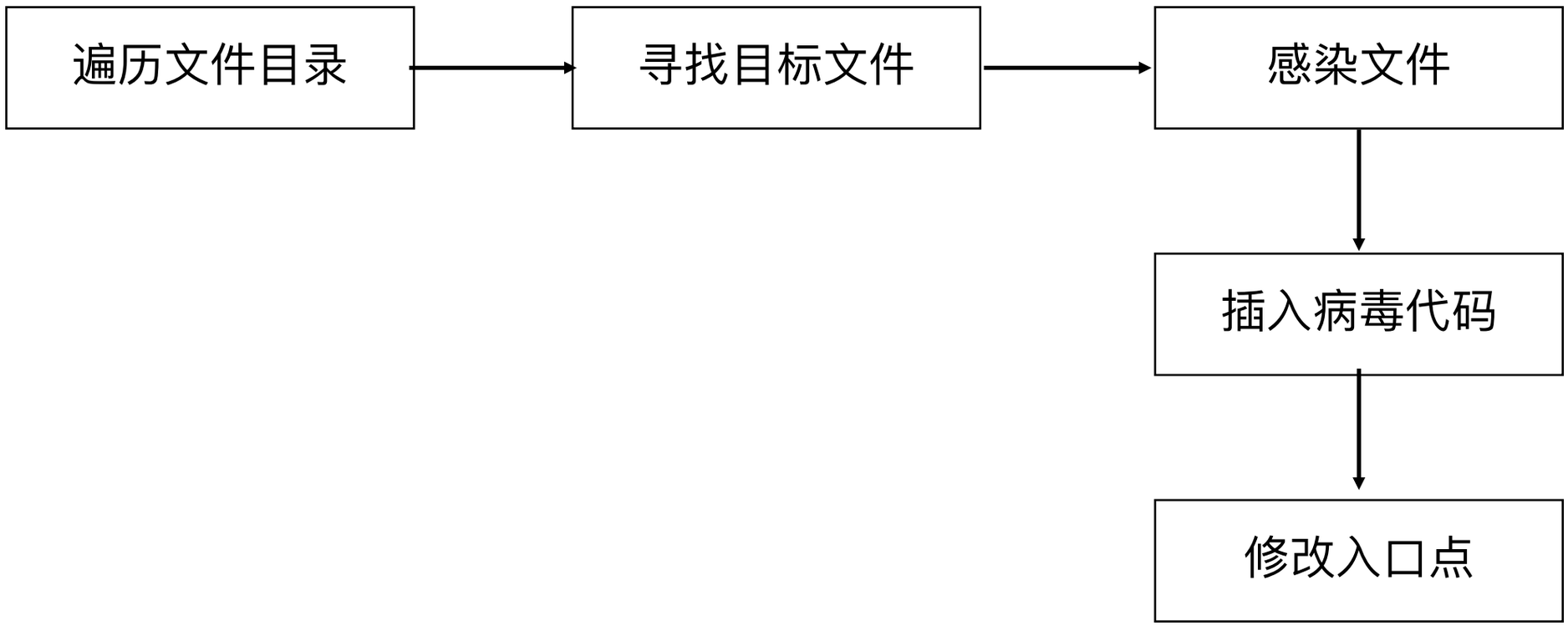
引导模块

- 检查运行环境，如操作系统、内存等环境信息
- 将病毒加载到内存，使病毒处于活跃状态

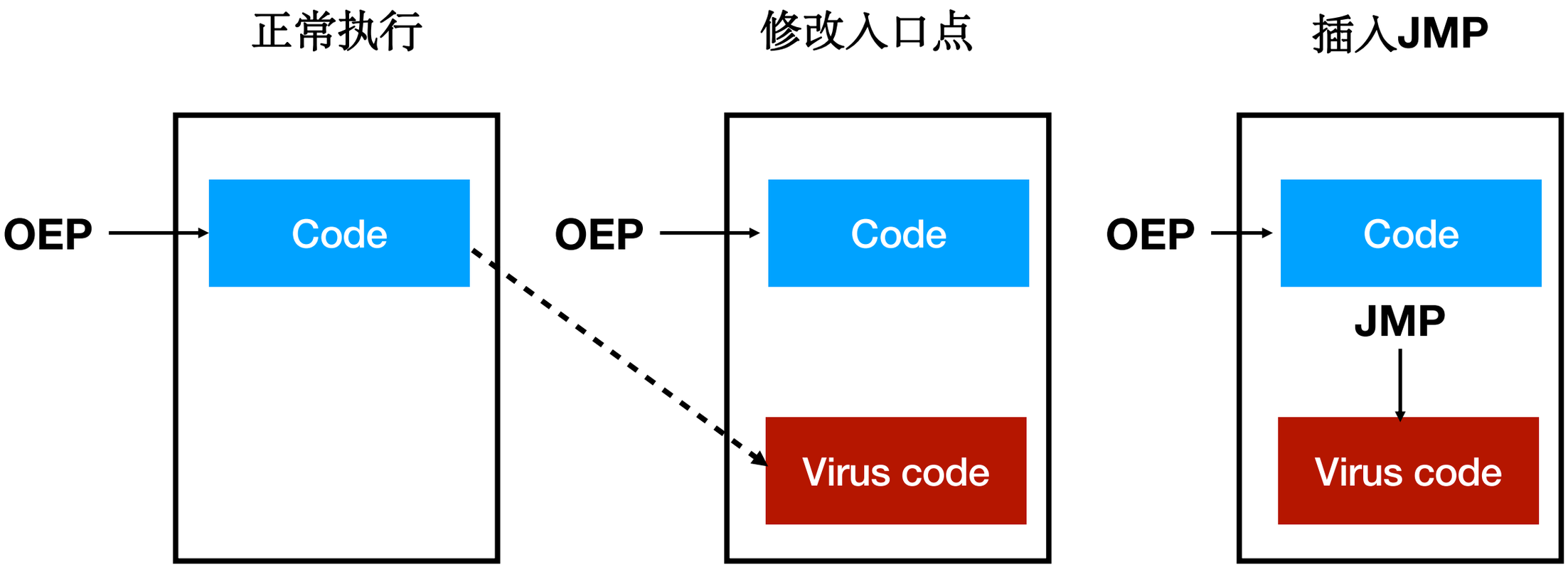
感染模块

- 寻找感染目标
- 检查目标是否满足感染条件
  - 文件类型、文件名、后缀等
- 如果满足感染，则将病毒插入目标文件

本地文件感染过程

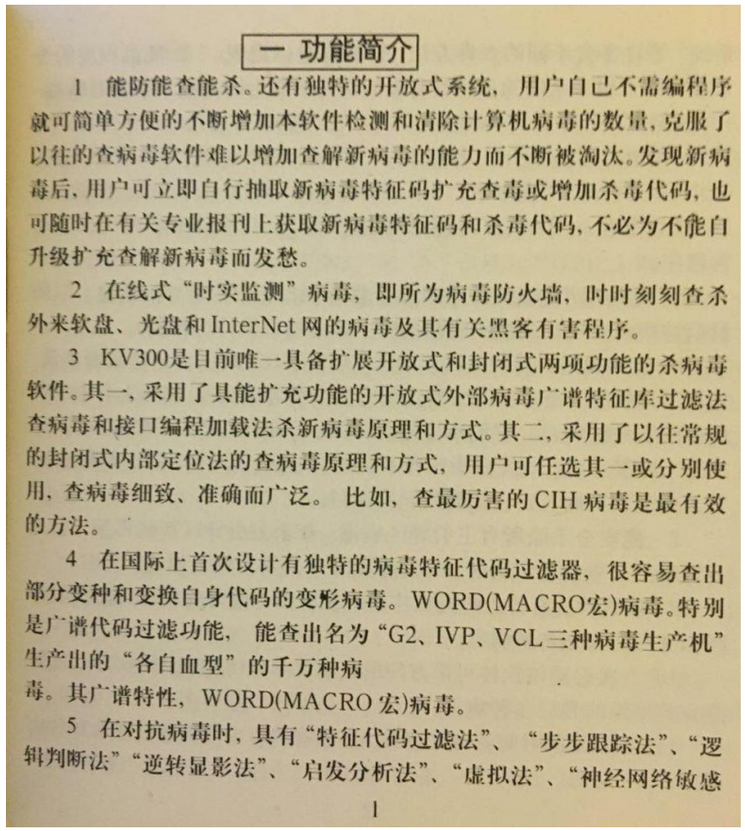


PE文件感染方式



病毒检测

- 主要采用**特征匹配(signature match)**和**动态分析技术**



一 功能简介

1 能防能查能杀。还有独特的开放式系统，用户自己不需编程序就可简单方便的不断增加本软件检测和清除计算机病毒的数量，克服了以往的查病毒软件难以增加查解新病毒的能力而不断被淘汰。发现新病毒后，用户可立即自行抽取新病毒特征码扩充查毒或增加杀毒代码，也可随时在有关专业报刊上获取新病毒特征码和杀毒代码，不必为不能自升级扩充查解新病毒而发愁。

2 在线式“时实监测”病毒，即所为病毒防火墙，时时刻刻查杀外来软盘、光盘和InterNet网的病毒及其有关黑客有害程序。

3 KV300是目前唯一具备扩展开放式和封闭式两项功能的杀病毒软件。其一，采用了具能扩充功能的开放式外部病毒广谱特征库过滤法查病毒和接口编程加载法杀新病毒原理和方式。其二，采用了以往常规的封闭式内部定位法的查病毒原理和方式，用户可任选其一或分别使用，查病毒细致、准确而广泛。比如，查最厉害的CIH病毒是最有效的方法。

4 在国际上首次设计有独特的病毒特征代码过滤器，很容易查出部分变种和变换自身代码的变形病毒。WORD(MACRO宏)病毒。特别是广谱代码过滤功能，能查出名为“G2、IVP、VCL三种病毒生产机”生产出的“各自血型”的千万种病
毒。其广谱特性，WORD(MACRO宏)病毒。

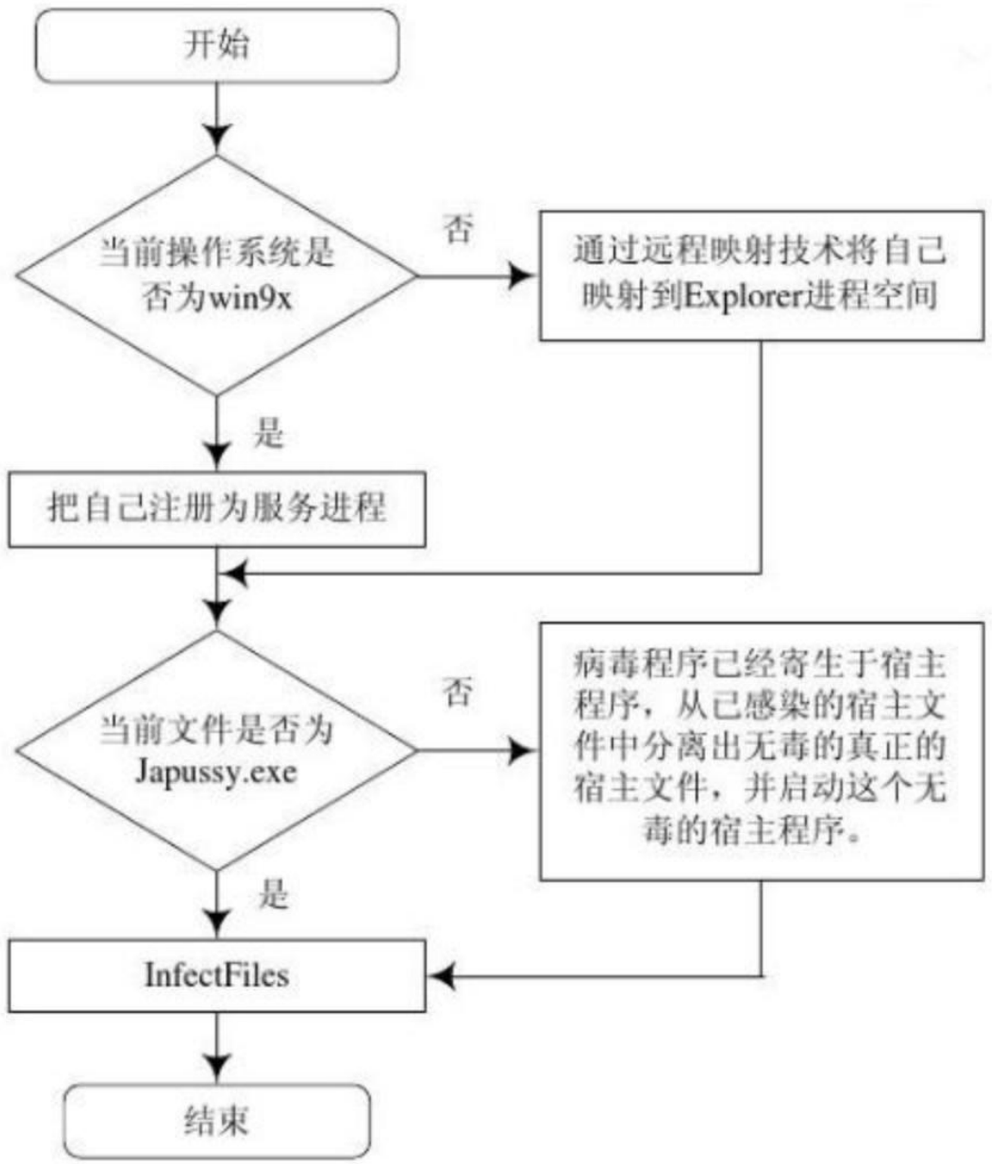
5 在对抗病毒时，具有“特征代码过滤法”、“步步跟踪法”、“逻辑判断法”“逆转显影法”、“启发分析法”、“虚拟法”、“神经网络敏感

1

病毒清除

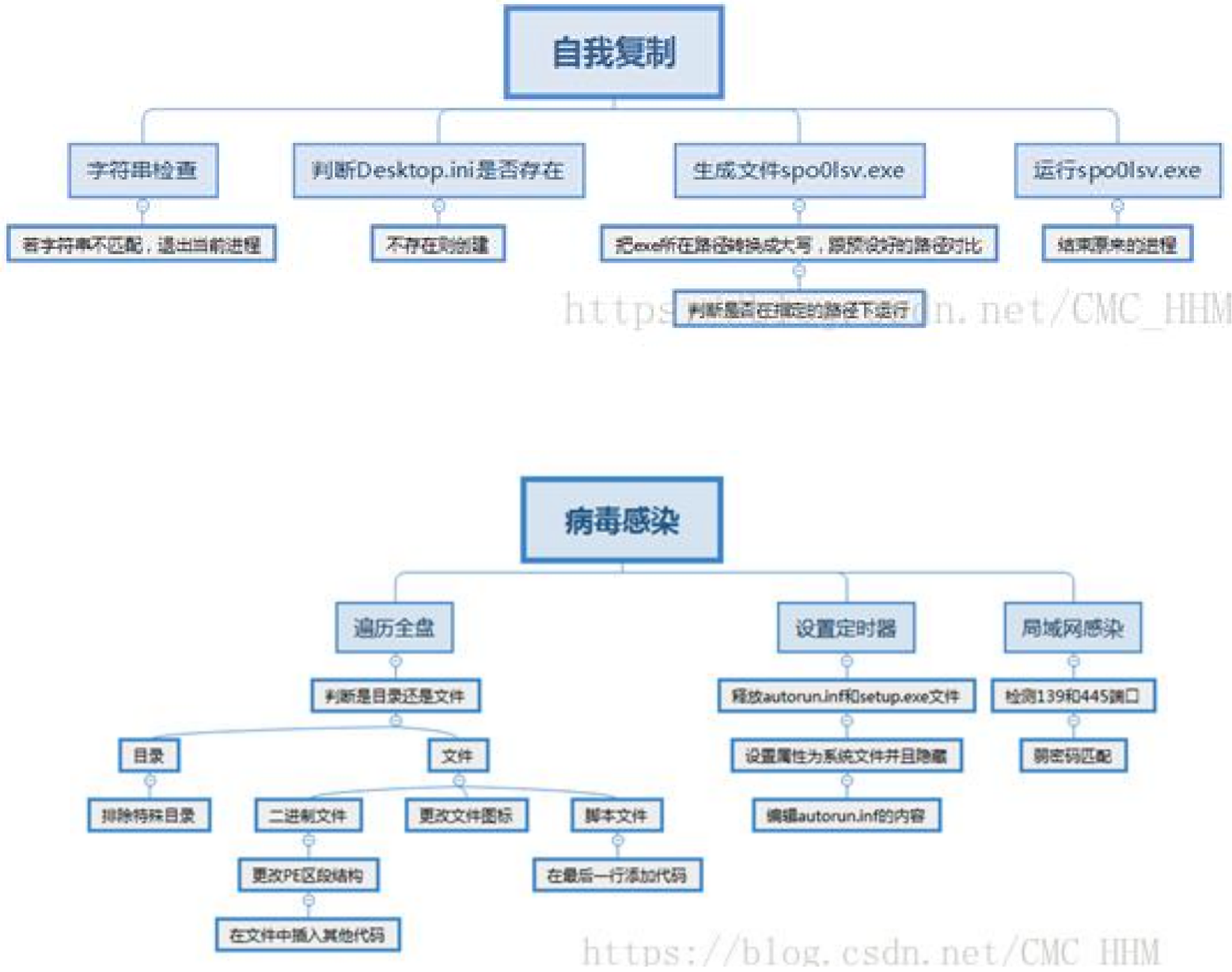
- 如果已感染文件，很难彻底清除
- 扇区修复十分困难

熊猫烧香病毒分析(1)

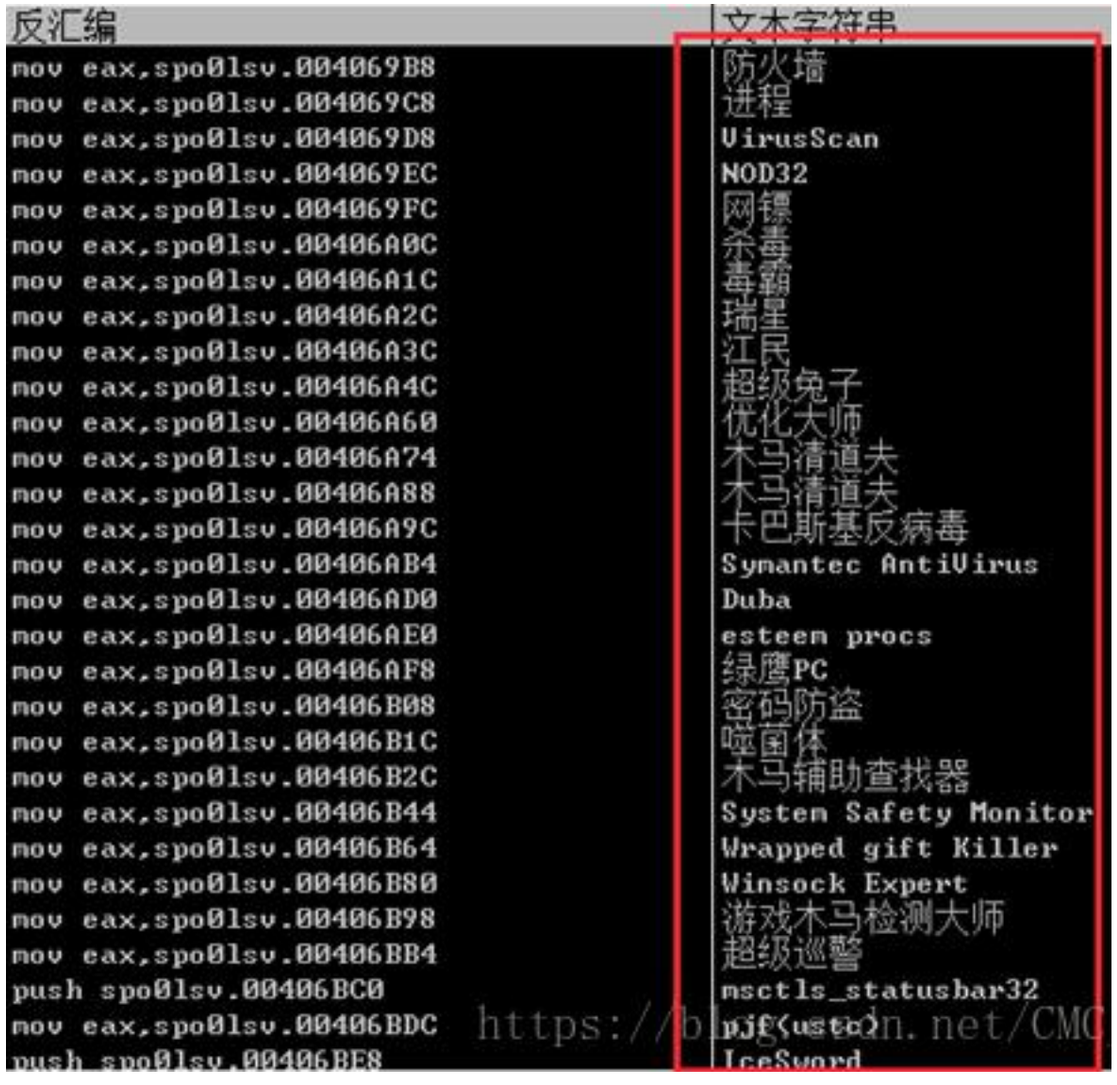


含有病毒体的文件被运行后，病毒将自身拷贝至系统目录，同时修改注册表，将自身设置为开机启动项，并遍历各个驱动器，将自身写入磁盘根目录，增加一个autorun.inf 文件，使得用户打开该盘符时激活病毒体。随后病毒体开一个线程进行本地染，同时开另外一个线程连接网站下载 DDoS 程序发起恶意攻击。

熊猫烧香病毒分析(2)



熊猫烧香病毒分析(3)



检测杀毒软件

本节小结

- 恶意代码定义、分类与命名规则
- 不同恶意代码特点
- 计算机病毒原理，重点掌握感染机制

第三章 恶意代码

第二节 木马、蠕虫与勒索软件

本节内容与目标

- 理解木马工作原理
- 了解蠕虫工作机制
- **掌握勒索软件工作流程**

木马

- “木马”来源于《伊利亚特》中的战争手段
- 木马定义
  - 指一种与远程计算机之间建立连接，使远程计算机能够通过网络控制目标系统，并可能造成信息损失、系统破坏的恶意程序。

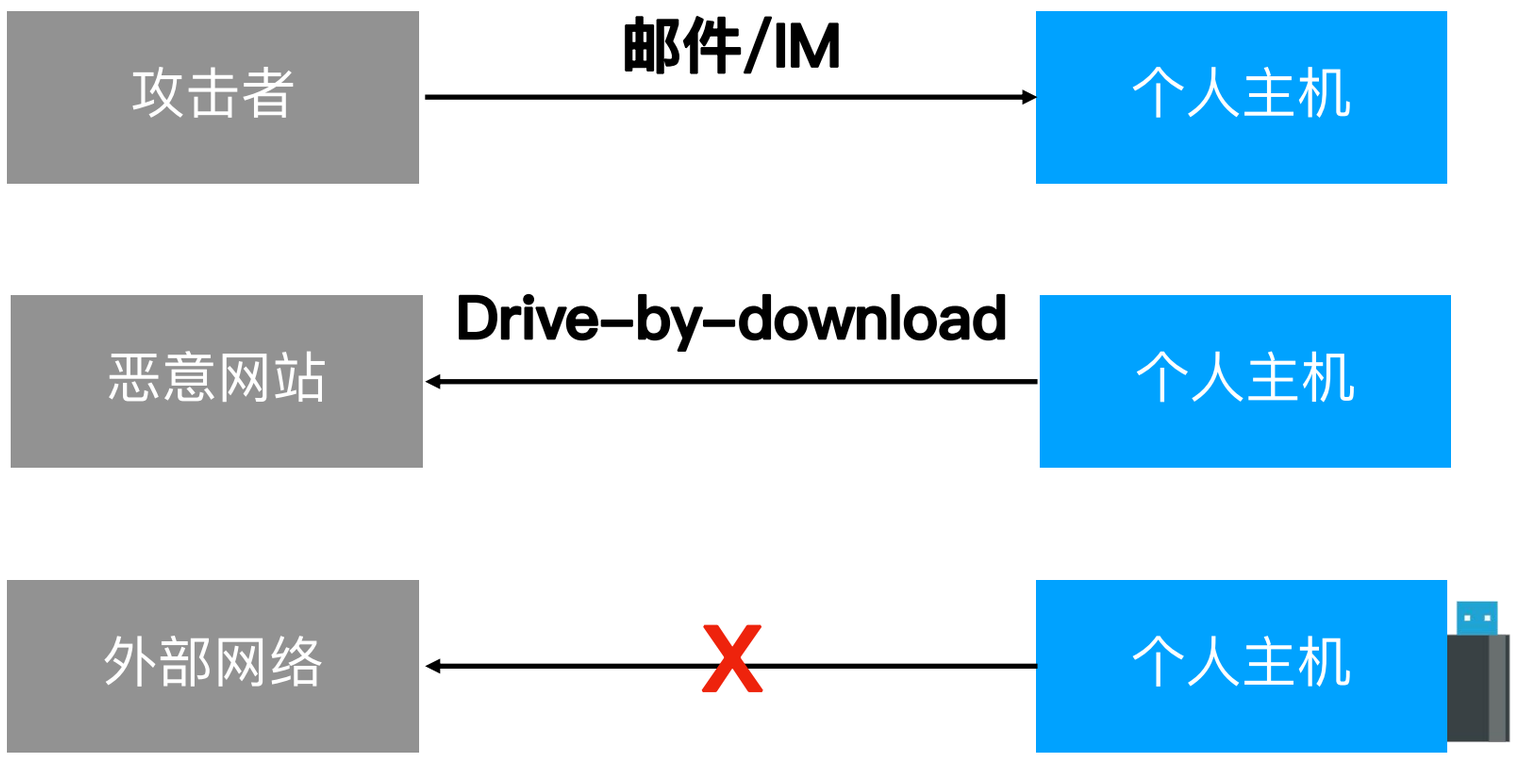
木马特点

- 木马特点
  - 欺骗性
  - 隐蔽性
  - 自动运行性
- 木马与病毒主要区别
  - 木马具有隐藏性特点，而病毒主要是起破坏作用
  - 木马感染性不强，病毒具有很强感染性

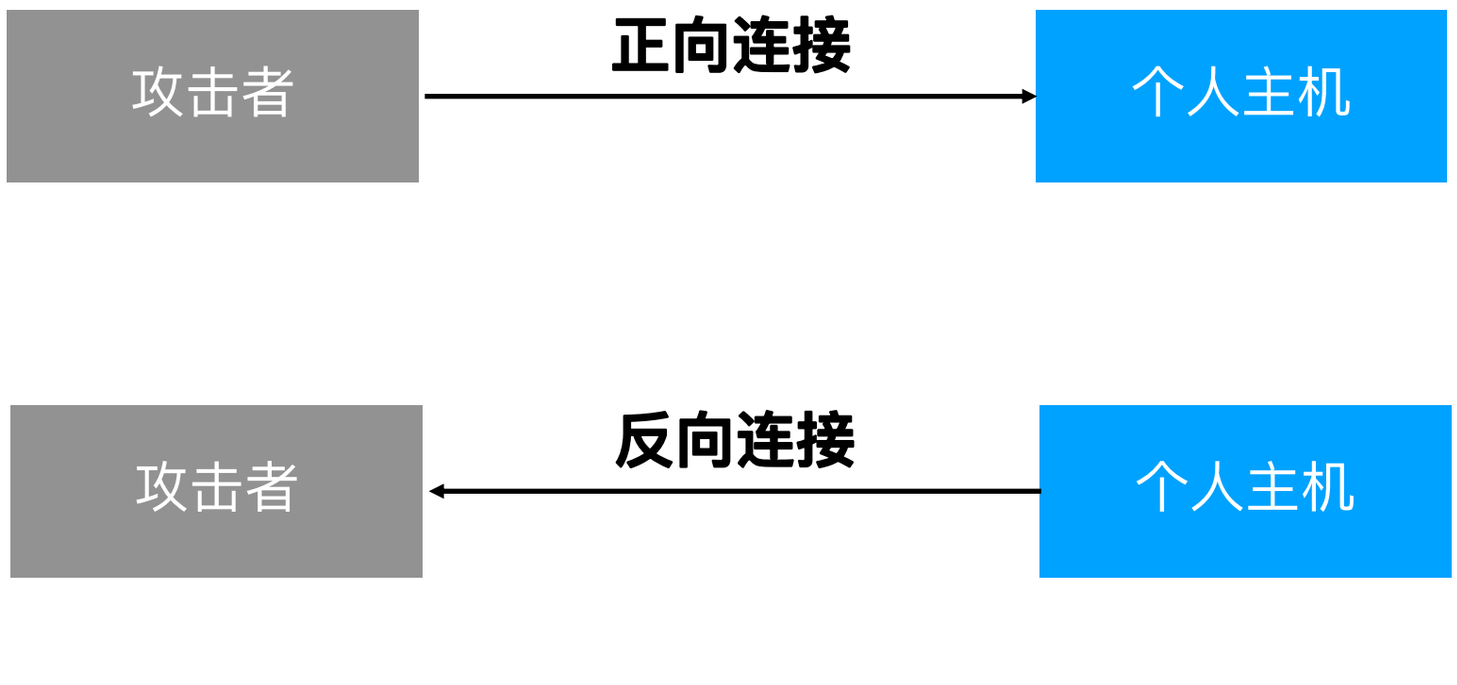
木马基本原理

谁是客户端，谁是服务端?

木马植入



控制技术



木马典型行为

- 自动运行
  - 注册表、Auto.inf、文件关联
- 隐藏技术
  - 进程隐藏、文件隐藏、网络隐藏
- 数据传输

木马技术发展

- 复杂性
- 更隐蔽(主机端和网络端)
- 专业化与平台化
- 面向新型系统与设备(如IoT)

木马检测

- 特征码
  - 静态规则匹配，被IDS、防火墙等安全系统广泛使用
- 动态行为分析
  - 根据木马行为进行检测

蠕虫

- 蠕虫定义

“计算机蠕虫（computer worm）与计算机病毒相似，是一种能够**自我复制**的计算机程序。与计算机病毒不同的是，计算机蠕虫不需要附在别的程序内，可能不用用户介入操作也能自我复制或运行。计算机蠕虫未必会直接破坏被感染的系统，却几乎都对网络有害。”

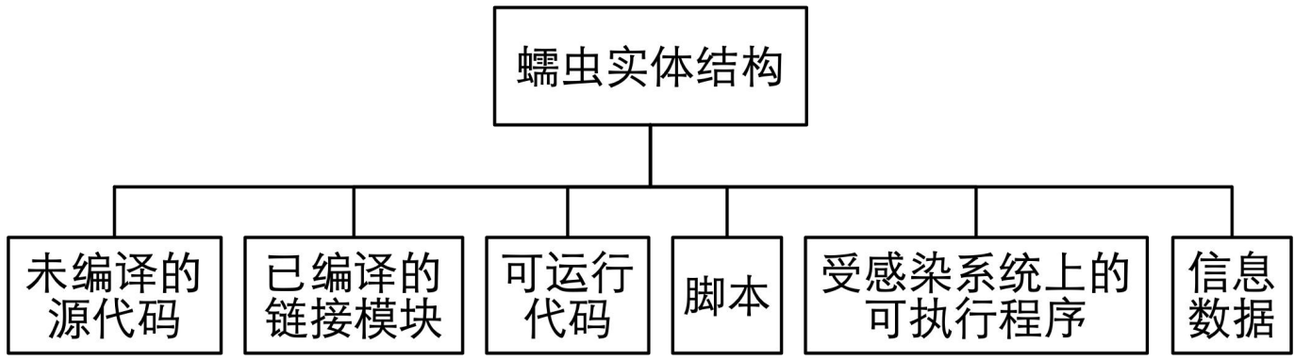
蠕虫典型特点

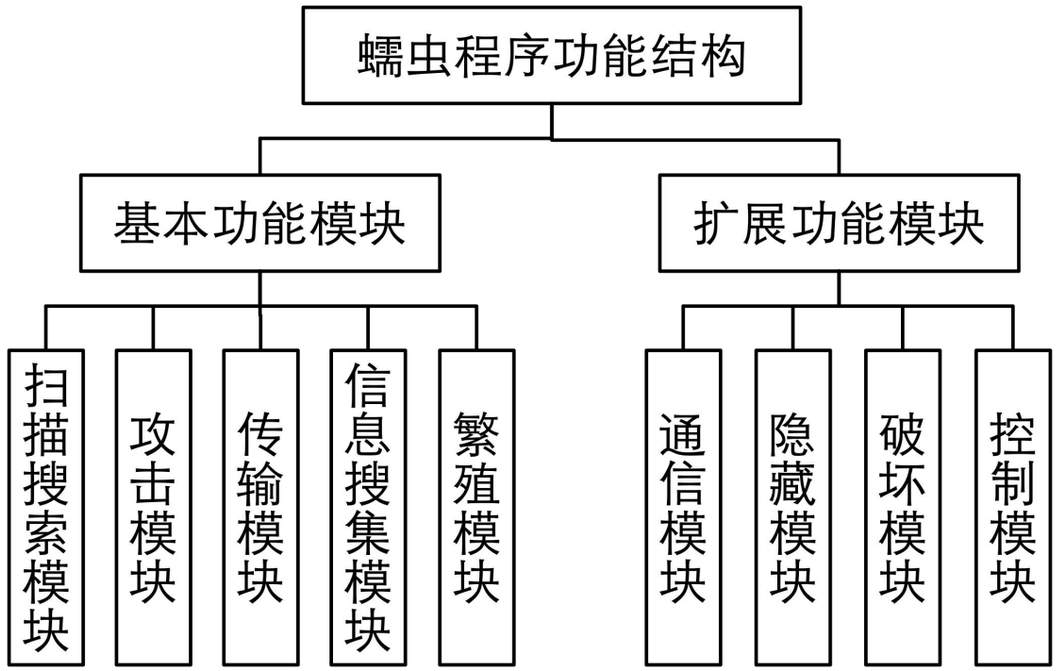
- 主要利用漏洞进行主动攻击
- 可不利用文件寄生(fileless)，也可为独立运行程序
- 传播方式多样
  - 主要通过网络进行传播，有较强的传播性/自我复制性
- 传播速度快
- 破坏性强

病毒和蠕虫区别

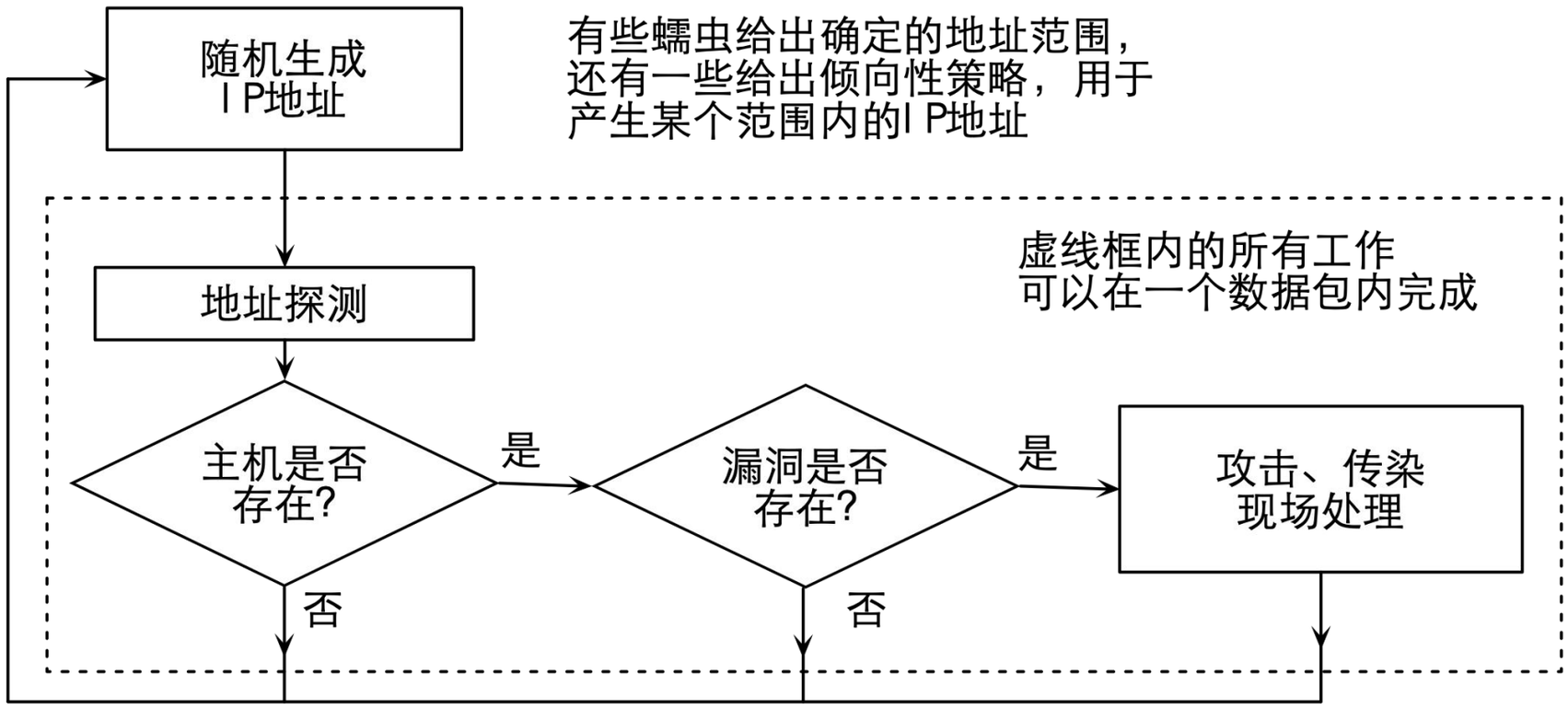
	病毒	蠕虫
存在形式	寄生	独立个体/寄生
复制机制	插入到宿主程序中	自身拷贝
传染机制	宿主程序运行	系统存在漏洞
搜索机制	主要针对本地文件	主要针对网络中其它计算机
影响重点	文件系统	网络性能、系统性能

蠕虫功能结构





蠕虫工作主要原理



蠕虫扫描策略

- 扫描目标
  - 尽量减少重复的扫描，使扫描发送的数据包总量减少到最小
  - 保证扫描覆盖到尽量大的范围
- 基础扫描策略
  - 随机选取某一段IP地址，然后对这一地址段上主机进行扫描
  - 大量扫描引起严重网络拥塞

其它常用扫描策略

- 选择性随机扫描(包括本地优先扫描)
- 可路由地址扫描(Routable Scan)
- 地址分组扫描(Divide–Conquer Scan)

蠕虫行为特征

- 主动扫描
- 网络拥塞
- 利用系统、网络应用服务器漏洞
- 消耗系统资源，降低系统性能

蠕虫技术发展

- 隐蔽性更强
  - 不被IDS/防火墙发现
  - 漏洞利用
- 功能结构更加复杂
  - 自动升级
  - 扫描策略更多样化
  - 功能更复杂

蠕虫检测

- 针对蠕虫网络行为特性构建检测模型
  - 扫描检测
  - 网络流量异常检测
- 特征码生成
  - 针对**多态**蠕虫的特征码生成

勒索软件

- 勒索软件是一种通过屏幕锁定或加密用户文件，并要求支付赎金来解密文件的恶意软件。

最早已知的索软件

- 1989年出现的”AIDS”是最早已知的勒索软件。
- 该病毒的实体数据会宣称受害者的某个软件已经结束了授权使用，并且加密磁盘上的文件，要求缴出189美元的费用给PC Cyborg Corporation以解除锁定。开发者Popp在法庭上以精神障碍（无行为能力）为自己辩护，但他仍承诺将获得的非法款项用于资助艾滋病的研究。

1989 AIDS/PC-CYBORO

CyberLocker

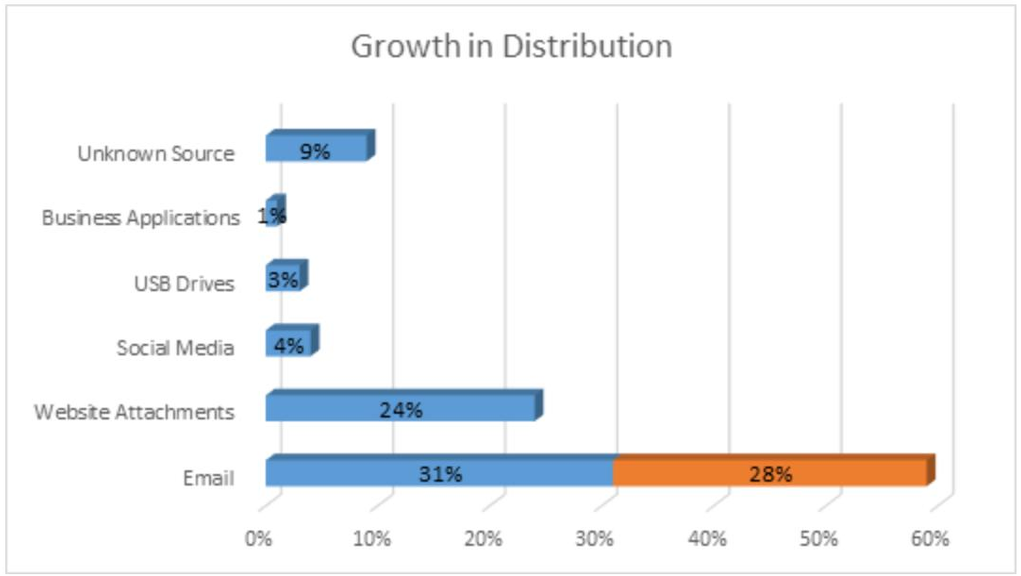
- 2013年出现
- 钓鱼邮件传播
- 要求72小时支付比特币
- 感染50万台主机
- 勒索金额约$27M

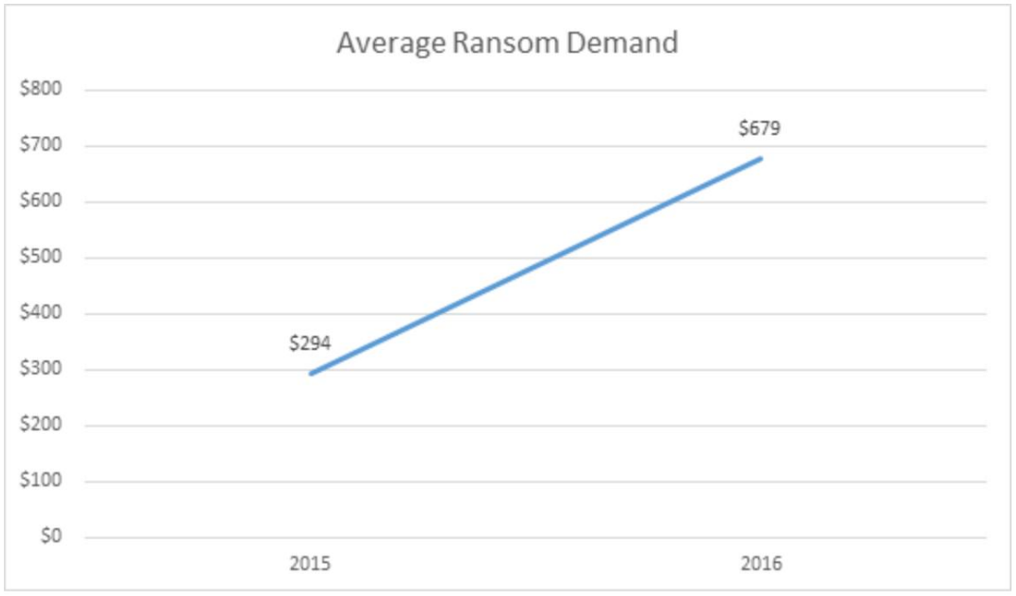
WannaCry

- 数百万台主机中招
- 全球损失约90亿美元
- 主要利用永恒之蓝漏洞

勒索软件数据

- 勒索金额逐年递增
- 邮件是主要传播渠道





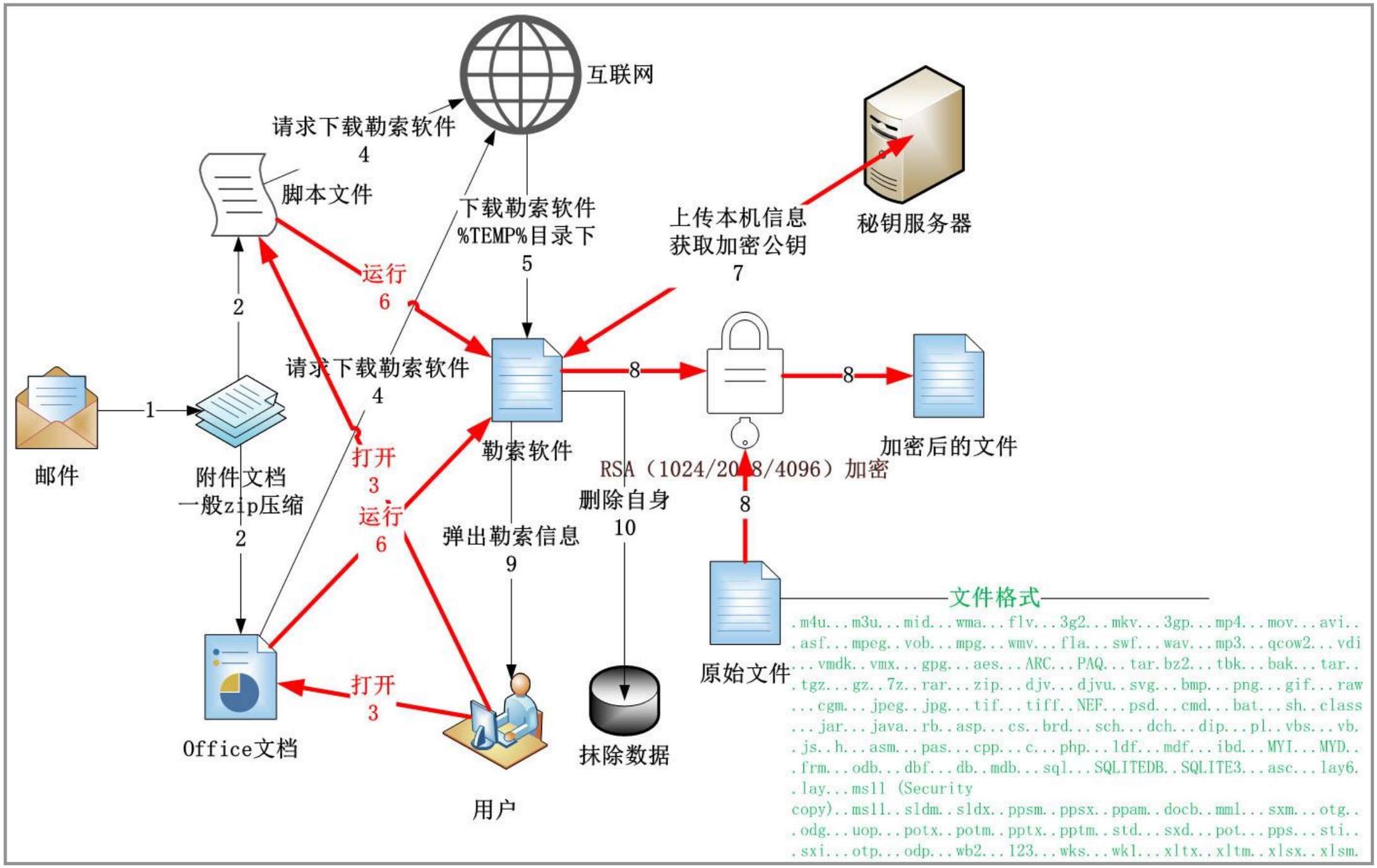
https://blogs.systweak.com/ransomware-statistics-growth-of-ransomware-in-2016/

勒索软件技术特点

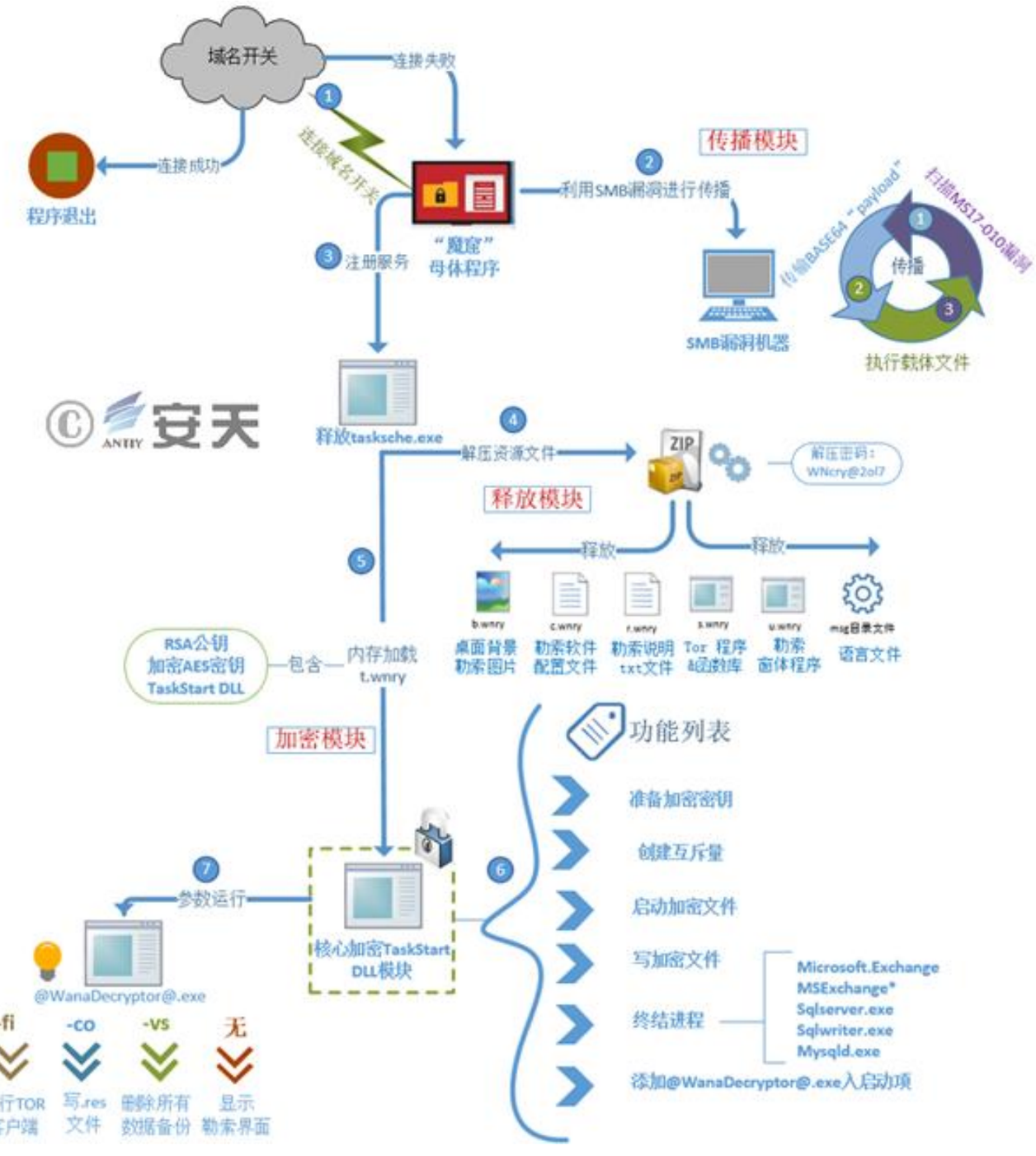
- 以加密(本地)数据为主要特点
- 加密程度不断提高
- 具有躲避安全检测的能力
- 电子支付追踪手段困难

勒索软件工作流程

勒索软件详细工作流程



案例分析 WannaCry



本节小结

- 木马、蠕虫与勒索软件原理
- 蠕虫与病毒的区别
- 勒索软件工作流程

第三章 恶意代码

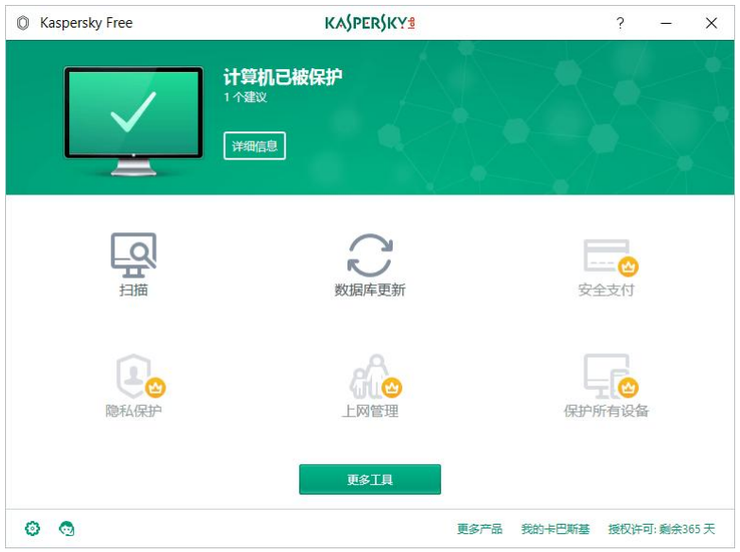
第三节 静态分析与检测技术

本节内容与目标

- 理解静态分析定义和使用场景
- 掌握杀毒软件工作流程
- 熟悉特征码提取流程

思考

- 杀毒软件工作原理是什么?
- 为什么需要频繁升级?
- 有哪些关键技术?
- 如何开发杀毒软件?



静态分析

- 静态分析定义
  - 在不执行程序的情况下对程序的文件结构、字符串、函数等静态信息进行分析，逆向分析恶意代码模块构成，包括内部数据结构和关键控制流程等。
- 使用场景
  - **理解恶意代码机理**
  - **分析恶意代码文件结构**

静态分析方法与比较

分析方法	目的	使用工具	难度
恶意代码扫描	标识已知恶意代码	反病毒引擎, VirusTotal	低
文件格式识别	确定攻击平台和类型	file, peid, FileAnalyzer	低
字符串提取	寻找恶意代码分析线索	strings	低
二进制结构分析	初步了解二进制文件结构	binutils (nm, objdump)	中
反汇编	二进制代码->汇编代码	IDA Pro, GDB, VC, …	中高
反编译	汇编代码->高级语言	REC, DCC, JAD, …	中高
代码结构与逻辑分析	分析二进制代码组，理解二进制代码逻辑成结构	IDA Pro, Ollydbg, …	高
加壳识别和代码脱壳	识别是否加壳及类型；对抗代码混淆恢复原始代码	UPX,VMUnpacker,手工	高

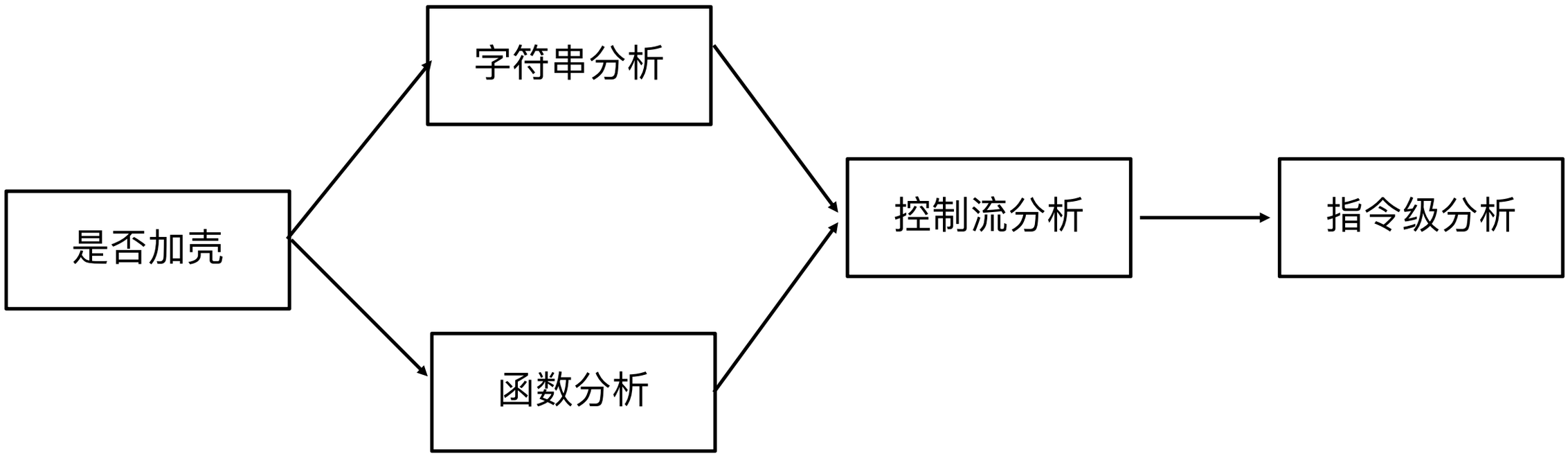
基础知识复习

- 程序编译过程
- **PE文件结构**

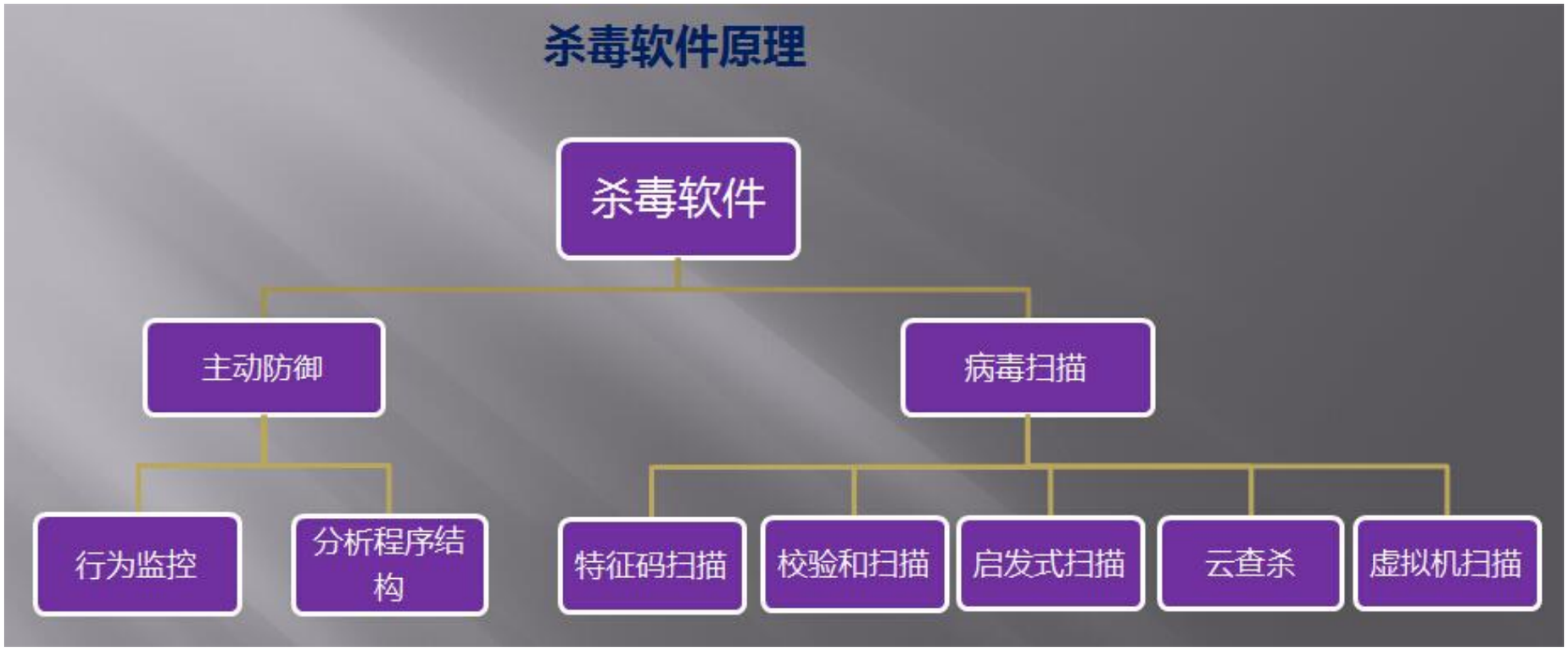
程序加壳

- 定义
  - “加壳的一种常用的方式是在二进制的程序中植入一段代码，在运行的时候优先取得程序的控制权，之后再把控制权交还给原始代码，这样做的目的是隐藏程序真正的OEP（入口点，防止被破解）”– 百度百科
- 使用场景
  - 保护应用程序版权，加大破解难度
  - 减小应用程序大小规模
  - 恶意代码绕过特征码检测

静态分析流程

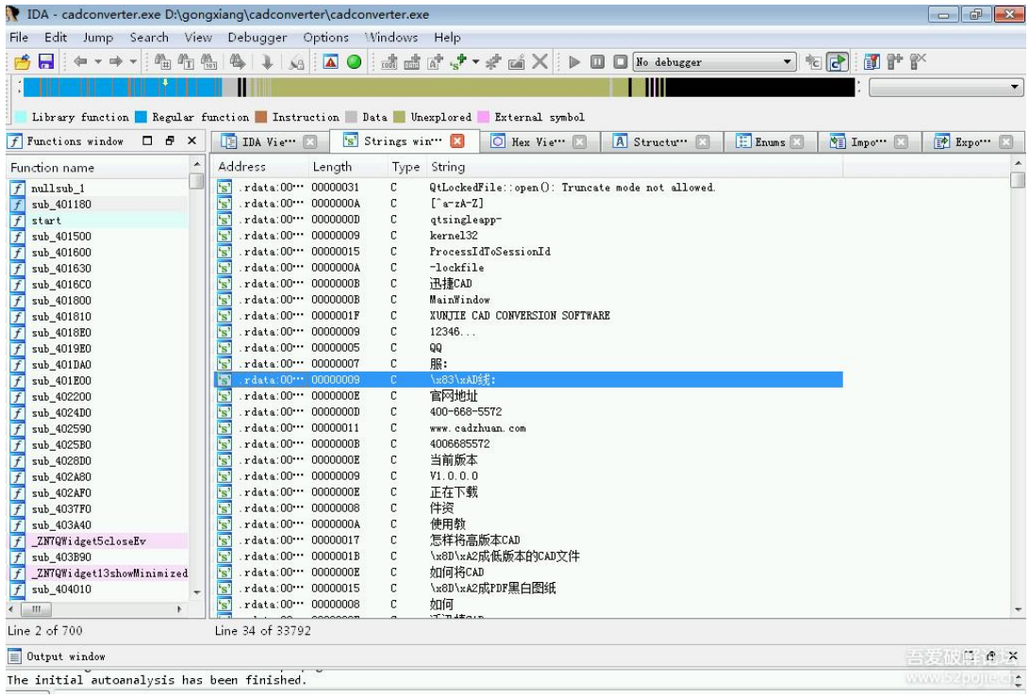


杀毒软件工作原理



字符串分析

- 字符串分析主要工具
  - Sysinternals
  - IDA Pro
- 字符串可能包含信息
  - 函数名
  - IP地址
  - 域名信息
  - 功能信息



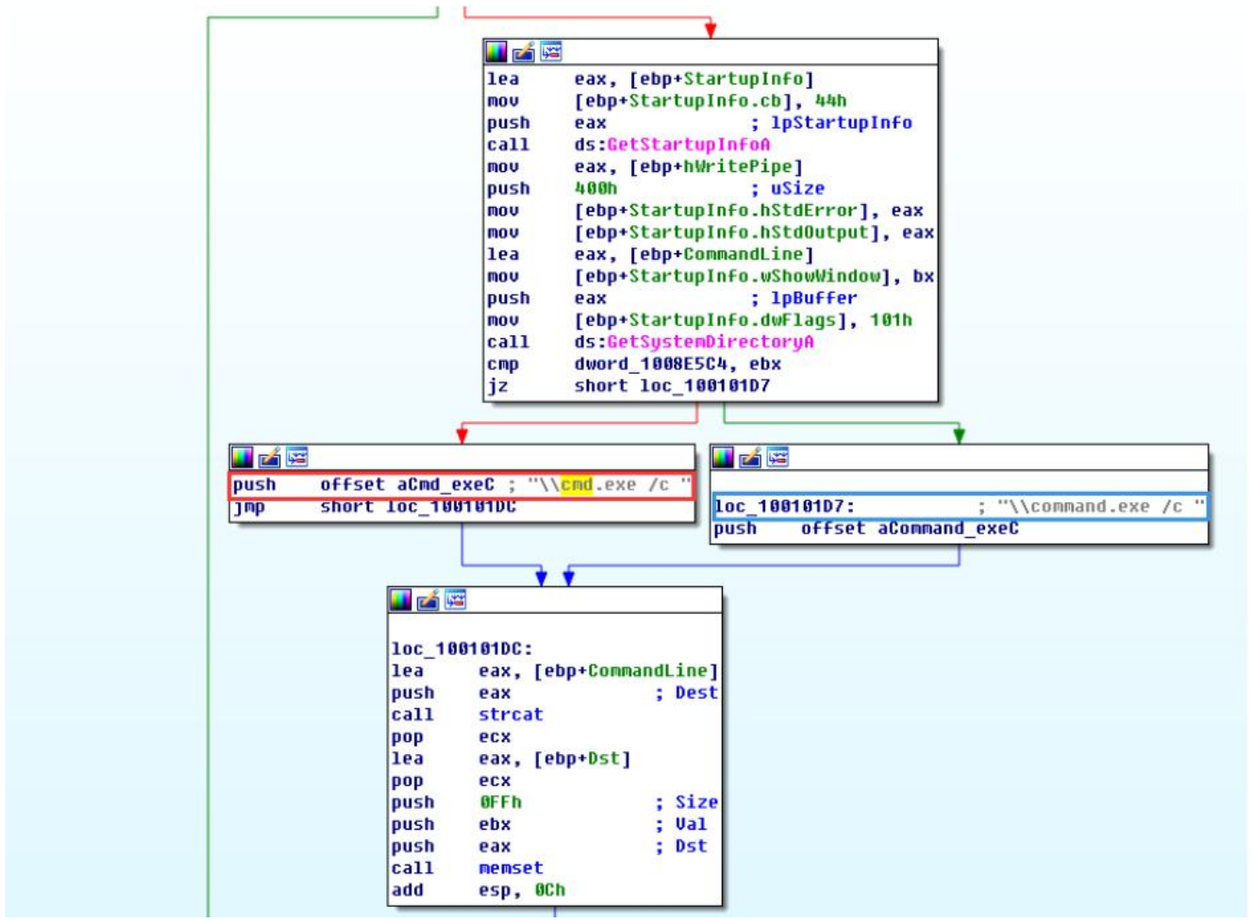

```
xdoors_d:10095B20 ; char aCommand_exeC[]
xdoors_d:10095B20 aCommand_exeC   db '\command.exe /c ',0 ; DATA XREF: sub_1000FF58:loc_100101D7↑o
xdoors_d:10095B31                 align 4
xdoors_d:10095B34 aCmd_exeC       db 'cmd.exe /c ',0      ; DATA XREF: sub_1000FF58+278↑o
xdoors_d:10095B41                 align 4
xdoors_d:10095B44 ; char aHiMasterDDDDDD[]
xdoors_d:10095B44 aHiMasterDDDDDD db 'Hi,Master [%d/%d/%d %d:%d:%d]',0Dh,0Ah
xdoors_d:10095B44                                         ; DATA XREF: sub_1000FF58+145↑o
xdoors_d:10095B44                 db 'WelCome Back...Are You Enjoying Today?',0Dh,0Ah
xdoors_d:10095B44                 db 0Dh,0Ah
xdoors_d:10095B44                 db 'Machine UpTime  [%-.2d Days %-.2d Hours %-.2d Minutes %-.2d Secon'
xdoors_d:10095B44                 db 'ds]',0Dh,0Ah
xdoors_d:10095B44                 db 'Machine IdleTime [%-.2d Days %-.2d Hours %-.2d Minutes %-.2d Seco'
xdoors_d:10095B44                 db 'nds]',0Dh,0Ah
xdoors_d:10095B44                 db 0Dh,0Ah
xdoors_d:10095B44                 db 'Encrypt Magic Number For This Remote Shell Session [0x%02x]',0Dh,0Ah
xdoors_d:10095B44                 db 0Dh,0Ah,0
xdoors_d:10095C5C ; char asc_10095C5C[]
```

(导入)函数分析

- 函数分析主要工具
  - IDA Pro
- 函数分析作用
  - 分析该程序部分功能

控制流分析

- 了解整体执行流程
- 分析**函数调用关系**



恶意代码静态检测

- 定义
  - 在不执行程序的情况下**检测**恶意代码
- 主要技术
  - 特征匹配(**包括hash**)
- 静态检测主要系统
  - AV(杀毒软件)，IDS(入侵检测系统)

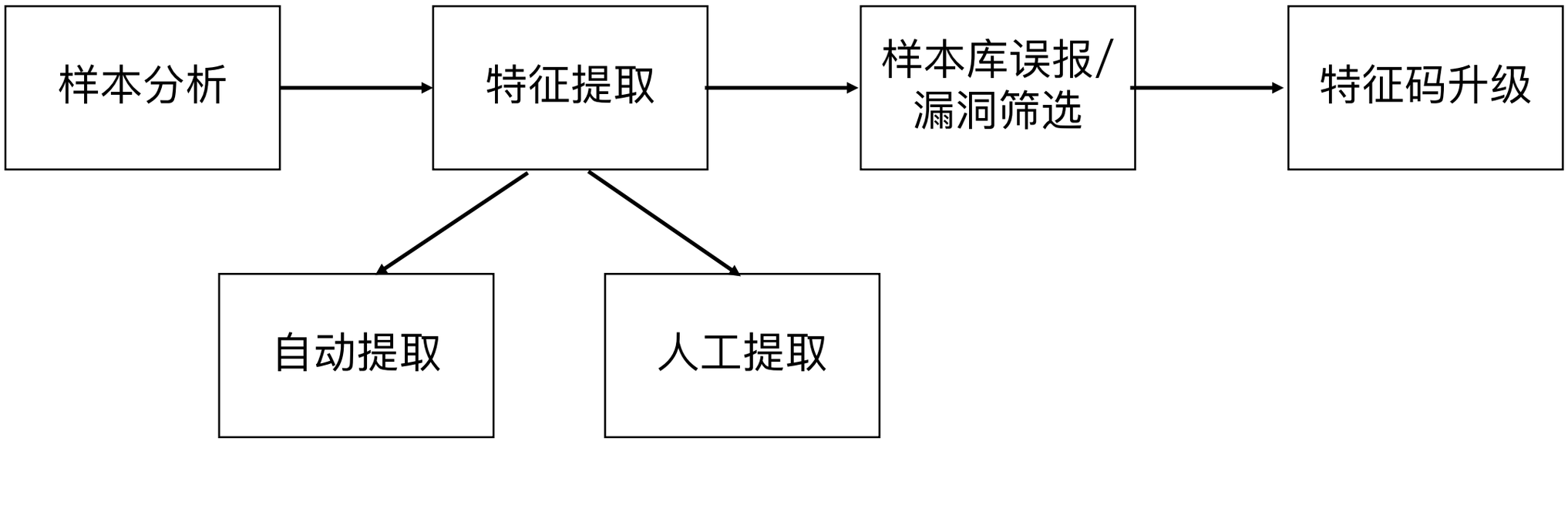
特征码

- 特征码定义
- 特征码提取方法
- 特征码匹配算法
- 特征码升级

特征码定义

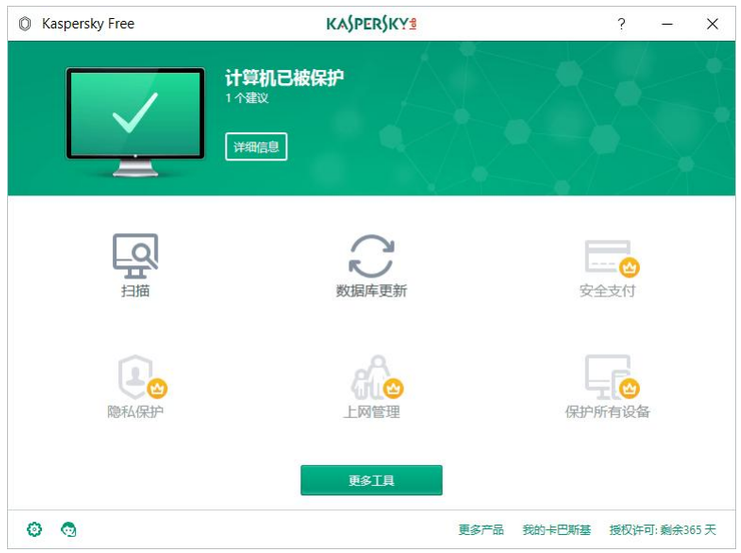
- 一段可**(唯一)**标识恶意代码的二进制序列
- 例如：02 39 45 9a 6f 10 40 ae f0 15 97
- 怎样提取特征码?
- 多少字节合适?

特征码提取流程



特征库升级

- 杀毒软件和IDS维护庞大的特征库
- 需要频繁升级(为什么？)



特征码匹配算法

- 典型字符串匹配算法
  - KMP, BM等模式匹配算法
- 实际环境
  - 如何高效匹配大量特征码与文件？例如1千万个特征码?
  - 软件实现效率能否达到要求?

ClamAV介绍

- “Clam AntiVirus（ClamAV）是免费而且开放源代码的杀毒软件，软件与病毒码的更新皆由社区免费发布。目前ClamAV主要是使用在由Linux、FreeBSD等Unix-like系统架设的邮件服务器上，提供电子邮件的病毒扫描服务。ClamAV本身是在文字接口下运作，但也有许多图形接口的前端工具（GUI front-end）可用，另外由于其开放源代码的特性，在Windows与Mac OS X平台都有其移植版。”

本节小结

- 静态分析方法
- 杀毒软件工作原理
- 特征码提取方法、匹配算法

第三章 恶意代码

第四节 动态分析与检测技术

本节内容与目标

- 免查杀技术原理
- 恶意代码常见隐蔽技术
- DKOM(内核数据修改)技术
- 沙箱技术

恶意代码隐蔽技术

- 提高恶意代码在目标系统生存能力，是恶意代码核心能力，常见隐蔽技术包括：
  - **免查杀**
  - **文件隐蔽**
  - **进程隐蔽**
  - **通信隐蔽**
  - **启动方式隐蔽**

免查杀

- 特征码通过二进制字节匹配检测恶意代码，“免查杀”目标为绕过(静态检测)。
- 免查杀主要通过**代码混淆**实现，包括
  - **加壳**
  - **花指令**
  - ...

花指令–插入垃圾代码

- 真实代码插入**垃圾代码**，保证原有程序正确执行，以此绕过静态检测。
- 例子：

```
add eax, ebx
mul ecx
```

插入垃圾代码前

```
xor esi, 011223344h ; garbage
add esi, eax ; garbage
add eax, ebx
mov edx, eax ; garbage
shl edx, 4 ; garbage
mul ecx
xor esi, ecx ; garbage
```

插入垃圾代码后

花指令–插入指令替换

- 替换与原有指令等价指令。
- 例子：

```
mov op1, op2可以替换为 push op2/pop op1这两条指令。
jmp label可以替换为 push label/ret这两条指令，IDA将不会显示被引用的label。
call label可以替换为push label_after_call_instruction/push label/ref这三条指令。
push op可以替换为 sub esp, 4(或者8)/mov [esp], op这两条指令。
```

动态分析

- 定义
  - 通过运行程序，观察程序产生的行为等动态信息，用以分析和检测恶意代码。
- 主要系统
  - 杀毒软件、沙箱

文件隐藏

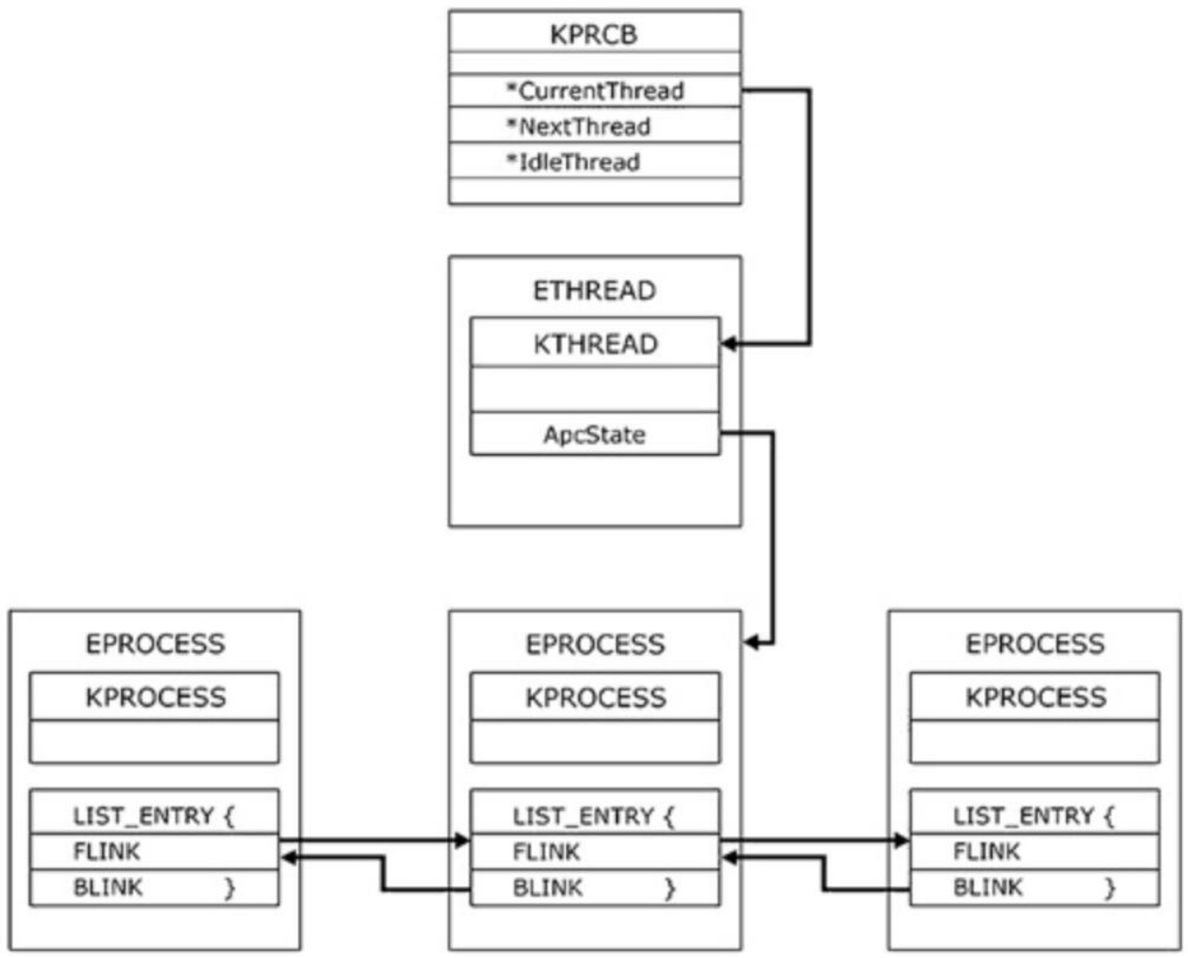
- 让用户无法觉察和搜索到恶意代码文件。
- 常见实现手段
  - 伪装为系统文件
  - 放在系统目录(如Windows目录)
  - 设置文件为隐藏
  - **利用驱动隐藏文件(ring0实现)**

进程隐藏

- 恶意代码在目标系统以**进程、线程、驱动**等形式运行，安全检测软件对系统运行程序进行检测。
- 常见实现手段
  - 进程名伪装(如svchost)
  - **修改进程信息，使进程不可见(DKOM技术，直接修改内核对象)**
  - **远程线程注入**
  - **利用DLL实现隐藏**
  - **实现为驱动或系统服务**

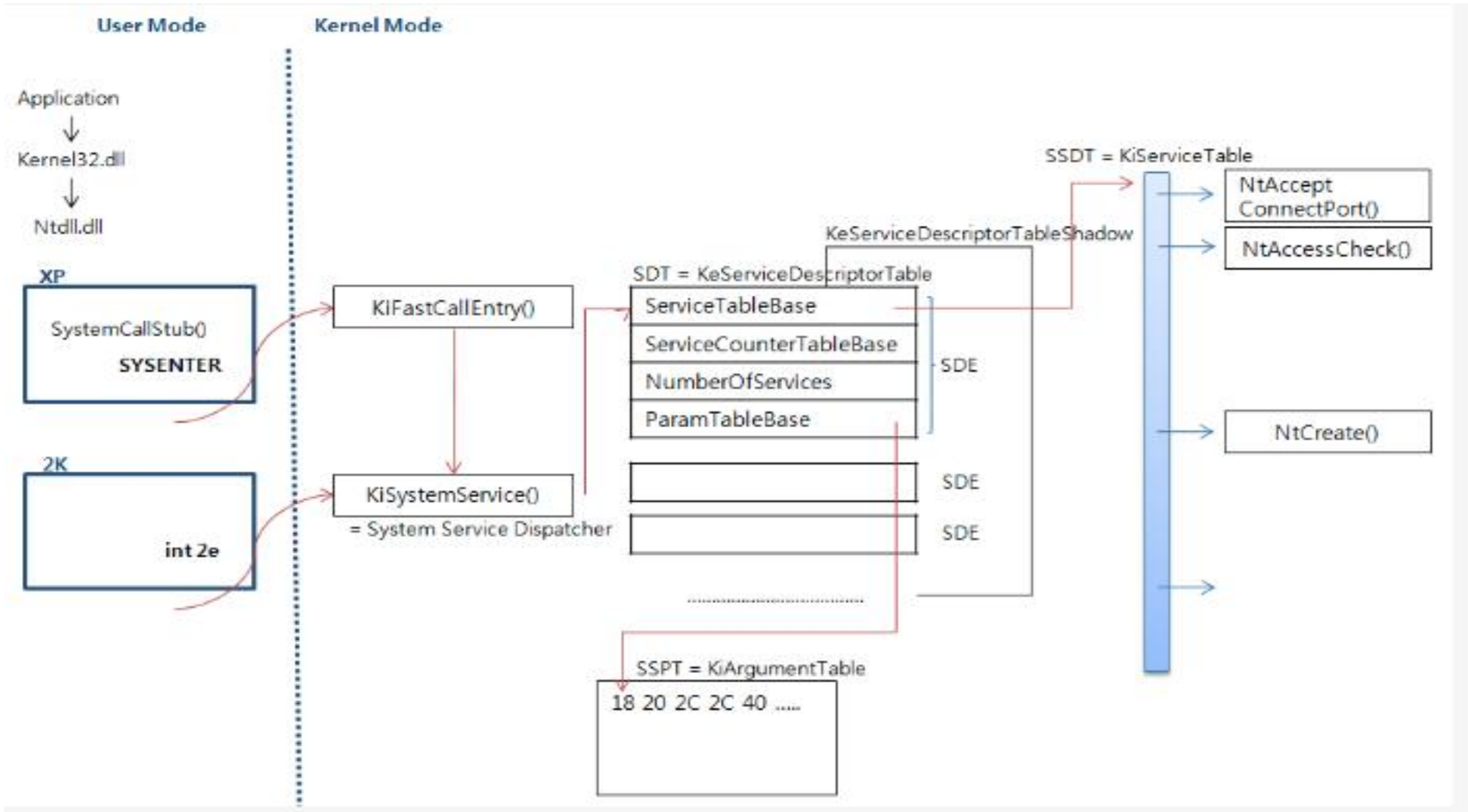
DKOM 修改内核对象

- **EPROCESS**是Windows进程内核对象，系统使用**ZwQuerySystemInformation**( )获取进程信息。
- 通过**修改EPROCESS**链表实现进程隐藏。



SSDT表

- 内核API地址表
- SSDT未公开但可导入
- 通过**hook(钩子)**替换为分析函数



通信隐藏

- 主要通过**隧道技术**实现，将数据封装在常见协议数据包中
  - DNS
  - HTTP(s)
  - ICMP

动态分析内容

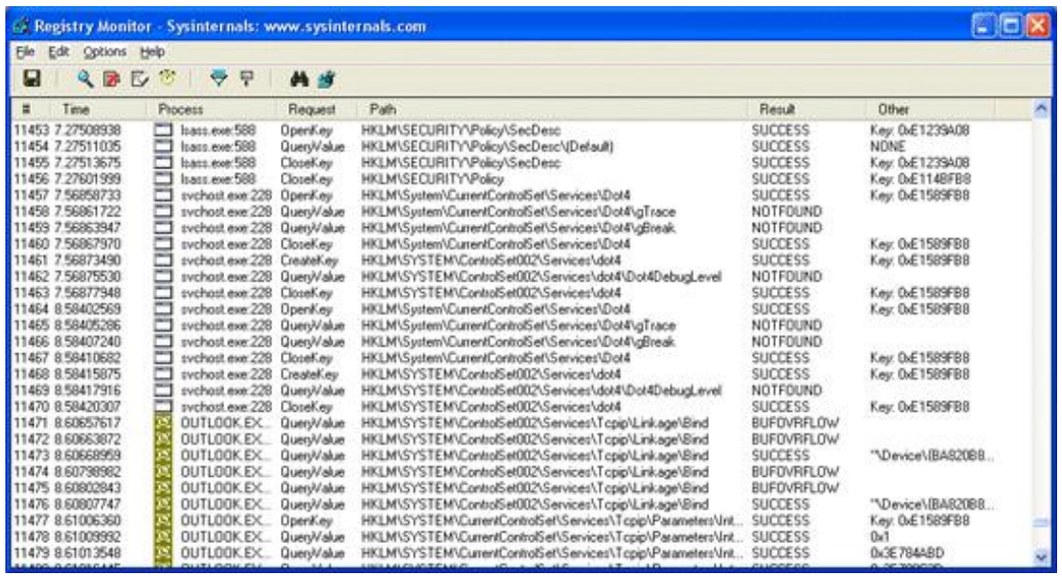
- 进程
  - 创建进程、关闭进程、线程注入
- 注册表
  - 关键位置注册表信息修改
- 文件读写
  - 系统文件读写、文件释放
- 网络
  - 网络请求(HTTP, DNS访问等)
- 服务相关
  - 加入启动项等

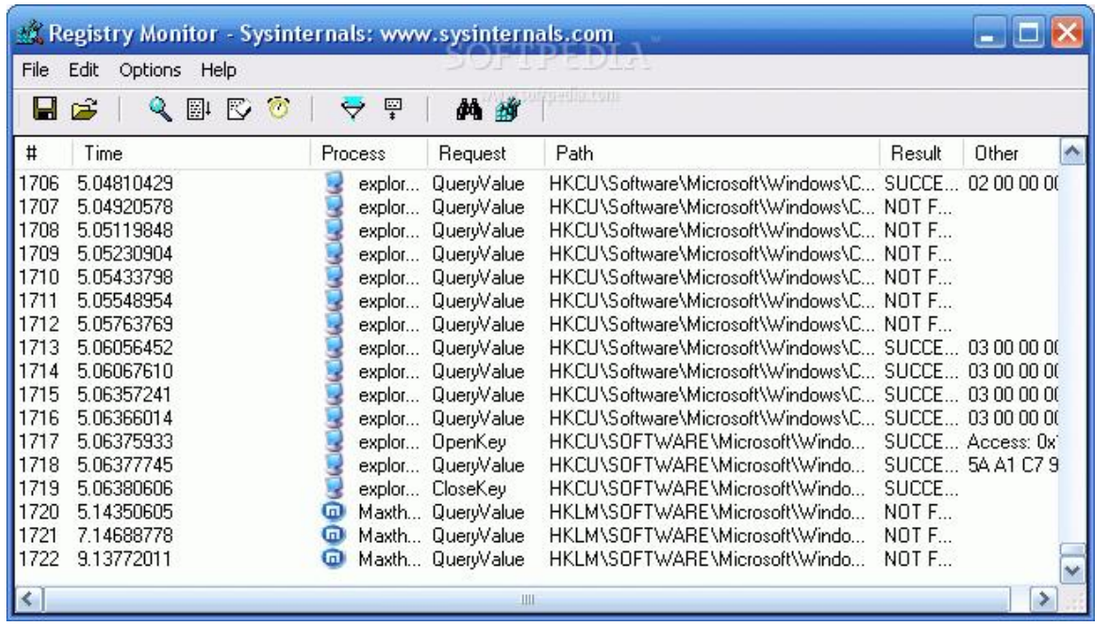
动态分析常用工具

- ProcessMonitor(前身为FileMon和Regmon)
- 沙箱分析

文件分析工具Filemon

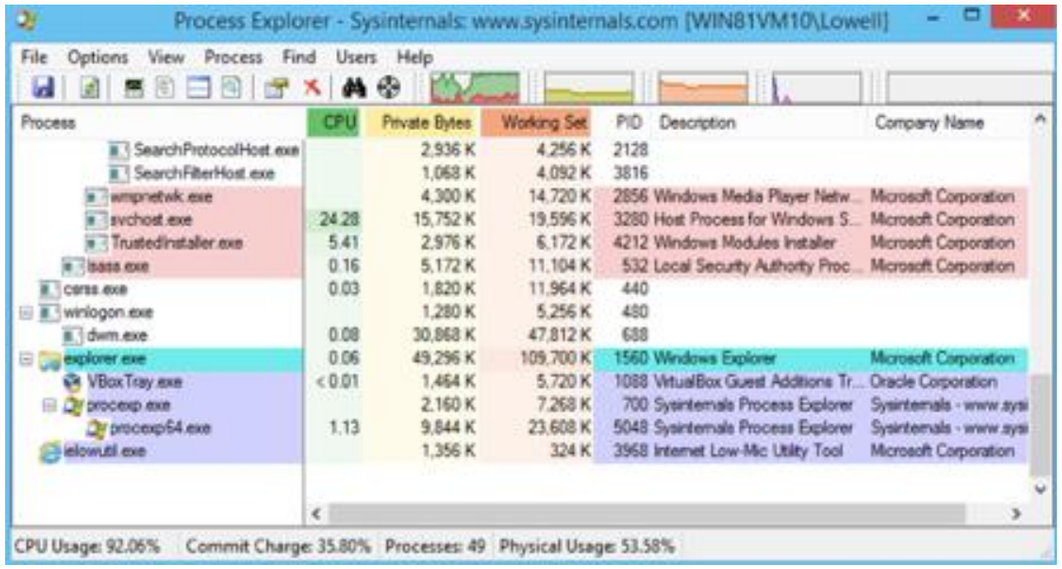
- “Filemon和Regmon是著名的文件监视和注册表监视工具，于1996年在Sysinternals网站上公开下载，早期的版本还提供了程序源代码。这两个系统监视工具是Windows NT系列操作系统上最早的驱动级文件和注册表监视工具，至今仍是最强大也是最稳定的文件和注册表监视工具之一，为后来的驱动开发者奠定了基础。”





ProcessMonitor简介

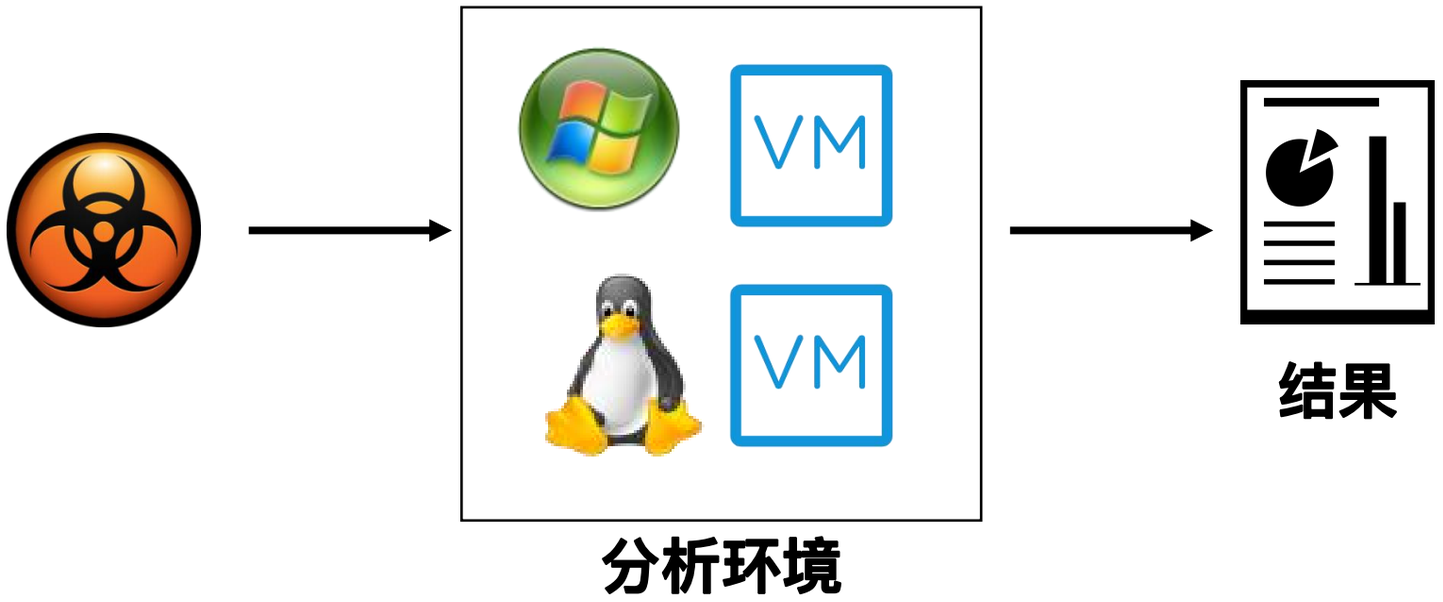
- Windows动态行为分析软件
- 前身为Regmon和Filemon
- 恶意软件行为分析利器



ProcessMonitor功能与特性

- 注册表
  - 监控注册表的创建、读取、删除或查询操作。
- 文件系统
  - 监控本地磁盘或网络驱动器中文件的创建、写入、删除等操作。
- 网络
  - 监控进程的 TCP/UDP 源和目标及流量。
- 进程
  - 可以被动监控进程和线程的活动，包括线程的启动或退出等。
- 性能分析
  - 捕获进程的 CPU 时间和内存使用。
- 稳定性和兼容性非常好
  - 支持多个Windows版本(7以上)

动态分析原理



内核API HOOK分析方案

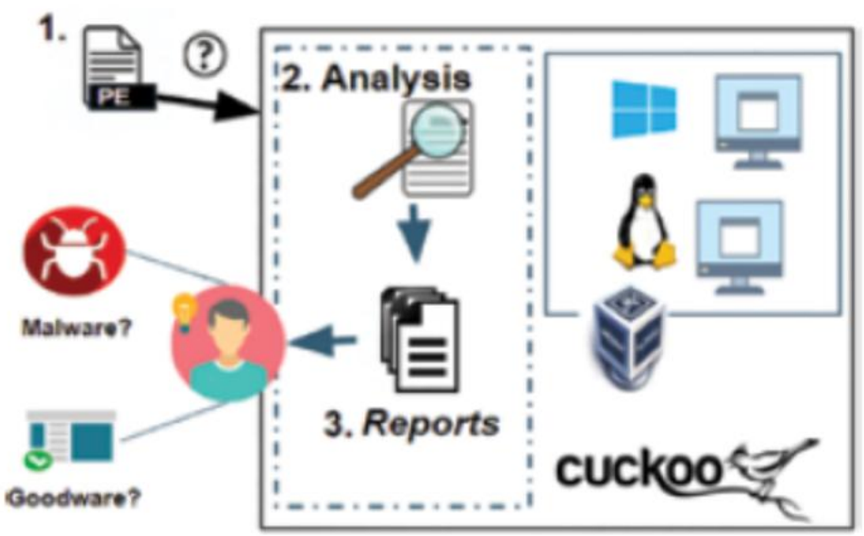
- 进程
  - NtOpenProcess, NtTerminateProcess
- 注册表
  - NtOpenRegistry, NtWriteRegistry
- 文件
  - NtOpenFile, NtWriteFile

沙箱分析

- 如何安全的分析恶意代码?
- 将恶意代码放入一个**安全的执行环境中**(**沙箱**)执行
- 业界常见(开源)沙箱
  - **Cuckoo**

Cuckoo沙箱

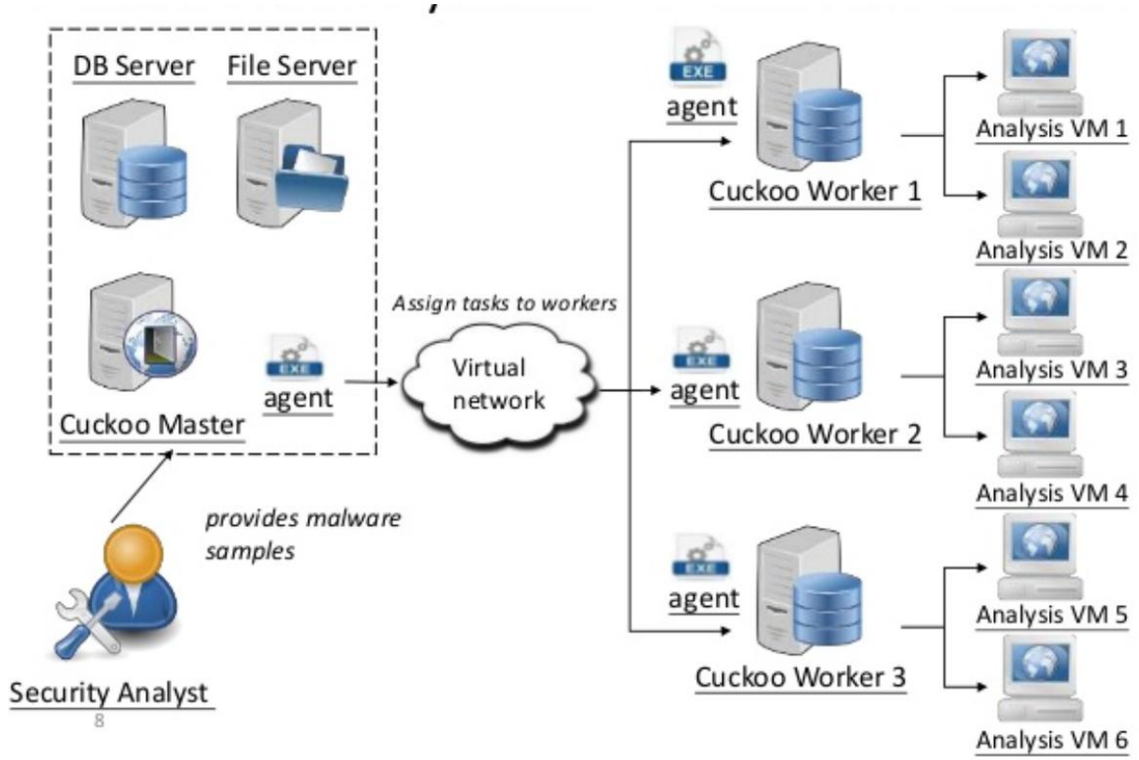
- 非常流行的开源沙箱系统
- 运行环境支持VMware, VirtualBox、KVM和docker
- 分析系统支持Windows、Linux、安卓等



- 主要特性
  - 文件、注册表、进程与内存dump分析
  - 网络数据包捕获
  - 规则支持，如Yara

分布式Cuckoo沙箱

- 大量样本分析需要更多算力
- 构建分布式沙箱集群



本节小结

- 免查杀技术
- 恶意代码隐藏技术，重点是内核数据结构修改