第六章 恶意软件

- 6.1 恶意软件的类型
- 6.2 高级持续性威胁
- 6.3 传染-感染内容-病毒
- 6.4 传播-漏洞利用-蠕虫
- 6.5 传播-社会工程学-垃圾电子邮件、木马
- 6.6 载荷-系统损坏
- 6.7 载荷-攻击代理-僵尸程序（zombie、bot）
- 6.8 载荷-信息窃取-键盘记录器、网络钓鱼、间谍软件
- 6.9 载荷-隐蔽-后门、rootkit
- 6.10 对抗手段

第六章 恶意软件

NIST SP 800-83 将**恶意软件**定义为：

“一种被（往往是秘密地）植入系统中的，以损害受害者数据、应用程序或操作系统的可信性、完整性或可用性，亦或对用户实施骚扰或妨碍的程序”。

6.1 恶意软件的类型

- 6.1.1 恶意软件的粗略分类
- 6.1.2 攻击工具包

恶意软件的相关术语

名称	描述
高级持续性威胁(APT)	指向商业性和政治性目标、使用多种入侵技术和恶意软件并在很长一段时间内发起持续有效的攻击的网络犯罪，其元凶往往是由国家支持的组织。
广告软件（Adware）	集成在软件中的广告程序。它能够产生弹出式广告或将浏览器重定向到某个商业网站。
攻击工具包（Attack kit）	一套通过使用各种传播和载荷机制自动生成新恶意软件的工具。
Auto-rooter	用于远程入侵到新的机器上的恶意攻击工具。
后门（陷门）	能够绕过正常安全检查的任何机制；它可以允许未经授权访问某些功能。
下载器(downloaders)	在被攻击的机器上安装其他内容的程序。通常下载器是包含在恶意代码中，该恶意代码首先被安装在被感染系统中，而后下载大量的恶意软件。
路过式下载(drive-by-download)	一种利用受感染网站的攻击方式。当该网站被访问时，被植入其中的恶意代码可以利用浏览器中的漏洞攻击访问者所在的系统。
漏洞攻击程序（Exploits）	针对某个或多个漏洞进行攻击的代码。
洪泛攻击程序（DoS）	通过向联网的计算机系统发送大量数据包而实现拒绝服务攻击的程序。
键盘记录器（keylogger）	捕获受控系统中键盘输入的程序。

逻辑炸弹	被入侵者插入到正常软件中的程序。当预定义的条件满足时，逻辑炸弹被触发，开始执行非授权的操作；其他时间处于休眠状态。
宏病毒	一种使用宏或脚本语言编写的病毒，通常被植入到一个文档中，当该文档被浏览或编辑时被触发和运行，复制自身至其他的文档。
移动代码	能够不加修改地移植到不同类型的系统平台上、并按照完全相同的语义执行的软件（如，脚本、宏或其他可移植的指令）。
Rootkit	攻击者成功入侵计算机系统并获得 root 访问权限之后使用的一套攻击工具
垃圾邮件程序	发送大量垃圾邮件的程序
间谍软件（spyware）	通过监听键盘输入、显示器数据和/或网络流量，或通过搜寻系统中的文件获取敏感信息，并将收集到的信息发送给另一台计算机的软件
特洛伊木马	一种计算机程序，看上去具有有用的功能，但还具有隐蔽的、潜在的恶意功能，这些恶意功能可以用来避开安全机制的检查，有时是利用调用被感染系统的合法授权来实现的。
病毒	当其执行时，设法将自己复制到其他可执行代码中的恶意软件。如果复制成功，就称这个可执行代码被感染了。当被感染的可执行代码运行时，病毒也同时被执行。
蠕虫	能够独立执行并且可以将自己完整的可执行版本传播到网络中其他主机上的计算机程序，通常通过攻击目标系统中软件的漏洞或使用捕获的授权凭证实现。
僵尸（Zombie, bot）	在被感染的计算机中运行，激活后向其他计算机发动攻击的程序。

6.1.1 恶意软件的粗略分类

两大类：

- 基于其向目标传播和感染的方式进行分类
- 基于其工作方式或有效负载进行分类

分类依据：

- 依附于宿主程序（寄生代码，如病毒）
- 独立的、自成一体的程序（蠕虫、特洛伊木马和僵尸程序）
- 不自我复制的恶意软件（特洛伊木马和垃圾邮件）
- 能够自我复制的恶意软件（病毒和蠕虫）

6.1.1 恶意软件的粗略分类

传播机制包括：

1.病毒感染现有内容，随后传播到其他系统

2.蠕虫或路过式下载利用软件漏洞，允许恶意软件自我复制

3.借助社会工程学方法说服用户绕过安全机制来安装木马或响应网络钓鱼

恶意软件到达目标系统后执行的有效负载操作可能包括：

1.污染系统或数据文件

2.窃取服务使系统成为僵尸网络中的一个僵尸代理

3.窃取系统信息/密钥记录

4.潜伏/隐藏其在系统上的存在

6.1.2 攻击工具包

- **最初，恶意软件的开发和部署需要软件作者具备相当的技术技能**
  - 在20世纪90年代早期开发了病毒开发工具包，然后在2000年代开发了更多的通用攻击工具包，极大地帮助了恶意软件的开发和部署
- **工具包通常被称为“犯罪软件”(crimeware)**
  - 包括各种传播机制和负载模块，即使是新手也可以部署
  - 攻击者使用这些工具包生成的变体会给防御系统带来严重问题
- **例如：Zeus，Angler**

6.1.2 攻击工具包

- 近几十年来，恶意软件开发发生了一些变化，攻击者由个人的、炫耀技术为目的变化为更加有组织化和危害性更大的攻击源，例如：

出于政治动机的攻击者	罪犯	有组织犯罪	向公司，国家，政府的代理人出售服务的组织	国家政府机构

- 这大大改变了恶意软件兴起背后的可用资源和动机，并催生了出售攻击包的地下经济、获取受害主机的控制权和盗窃信息。

6.2 高级持续性威胁

- 高级持续性威胁(Advanced Persistent Threats，简称APTs)
- 具有充足资源的、持续性的应用大量入侵技术和恶意软件的应用程序，这个应用具有选择的目标，通常是一些政治或商业目标
- 通常归因于国家赞助的组织和犯罪企业
- 与其他类型的攻击不同，它们精心选择目标，并在较长时间内进行隐蔽入侵
- 备受瞩目的范例包括Aurora、RSA、APT1和Stuxnet

APT的特征

高级	持续性	威胁
• 使用多种的入侵技术和恶意软件，如果有需要的话还会开发定制的恶意软件 • 其中单一的组件在技术上也许不先进，但是每个组件都是针对目标精心选择的	• 攻击者在很长时间内确定针对攻击目标的攻击应用可以最大化攻击成功的几率 • 攻击手段的种类是逐渐递增的，通常是非常隐秘的，直到目标被攻陷	• 针对选定目标的威胁来自于有组织的、有能力的和良好经济支持的攻击者，他们试图攻陷这些目标 • 攻击者的积极参与极大程度提升了自动攻击工具的威胁等级，也增加了成功攻击的可能性

APT攻击

目标：

- 窃取知识产权或安全和基础设施相关数据，也包括基础设施的物理损坏

使用的技术：

- 社会工程学
- 网络钓鱼邮件
- 选取目标组织中的人员可能会访问的网站植入下载驱动

意图：

- 使用具有多种传播机制和有效载荷的复杂恶意软件感染目标
- 一旦他们获得了对目标组织中系统的初始访问权限，就会使用更多的攻击工具来维护和扩展他们的访问权限

6.3 传染-感染内容-病毒

- 6.3.1 病毒的性质
- 6.3.2 宏病毒和脚本病毒
- 6.3.3 病毒的分类

6.3.1 病毒的性质

- **感染程序的软件**
  - 修改它们以包含病毒的副本
  - 复制并继续感染其他内容
  - 易于在网络环境中传播
- **当附加到可执行程序时，病毒可以做程序允许做的任何事情**
  - 在运行主机程序时秘密执行
- **特定于操作系统和硬件**
  - 利用他们的细节和弱点

病毒的组成部分

感染机制	触发条件	有效载荷
• 病毒传播和进行自我复制的方法 • 感染机制也被称为**感染向量**（infection vector）	• 激活或交付病毒有效载荷的事件或条件 • 有时被称为**逻辑炸弹**(logic bomb)	• 病毒除传播之外的活动 • 可能包括破坏活动，也可能包括无破坏但值得注意的良性活动

病毒的生命周期

潜伏阶段

- 病毒处于休眠状态
- 病毒会被某些事件激活
- 并不是所有的病毒都有这个阶段

触发阶段

- 病毒被激活以执行其预期功能
- 可以由多种系统事件引起

传播阶段

- 病毒将自身的拷贝插入其他程序或硬盘上某个与系统相关的区域
- 也许会和原始版本不完全一样
- 每个被感染的程序都包含病毒的一个拷贝，而且这些拷贝也会自己进入传播阶段

执行阶段

- 执行病毒功能
- 有的功能是无害的，有些则是破坏性的

6.3.2 宏病毒和脚本病毒

- **NISTIR 7298将宏病毒定义为：**
  - “一种病毒，它附着在文档上，并使用文档应用程序的宏编程功能来执行和传播”
- **宏病毒感染用于支持各种用户文档类型中活动内容的脚本代码**
- **宏病毒的威胁性有如下几个原因：**
  - 独立的平台
  - 感染的是文档而不是可执行部分的代码
  - 很容易传播
  - 由于宏病毒感染用户文档而不是系统程序，传统文件系统的访问控制在防止其扩散方面的作用有限，因为用户需要修改它们
  - 比传统的可执行病毒更容易制造或修改

Melissa宏病毒的伪代码

```
Macro Document_Open
    disable Macro menu and some macro security features
    if called from a user document
        copy macro code into Normal template file
    else
        copy macro code into user docment being opened
    end if
    if registry key “Melissa” not present
        if Outlook is email client
            for first 50 addresses in address book
                send email to that address
                with currently infected document attached
            end for
        end if
        create registry key “Melissa”
    end if
    if minute in hour equals day of month
        insert text into document being opened
    end if
end macro
```

6.3.3 病毒的分类

依照目标进行分类

- **感染引导扇区病毒**
  - 感染主引导记录或引导记录，并在从包含病毒的磁盘引导系统时传播
- **感染可执行文件病毒**
  - 感染操作系统或shell中执行的文件
- **宏病毒**
  - 由应用程序解释的宏或脚本代码感染文件
- **多元复合型病毒**
  - 以多种方式感染文件

6.3.3 病毒的分类

依照病毒的隐藏方式进行分类

- **加密型病毒**
  - 病毒的一部分创建随机加密密钥，并加密病毒的其余部分
- **隐蔽型病毒**
  - 一种明确设计用于隐藏自身以躲避反病毒软件检测的病毒
- **多态病毒**
  - 每次感染都会变异的病毒
- **变性病毒**
  - 一种病毒，在每次迭代中完全变异和重写自身，并可能改变行为和外观

6.4 传播-漏洞利用-蠕虫

- 6.4.1 发现目标
- 6.4.2 蠕虫传播模型
- 6.4.3 Morris蠕虫
- 6.4.4 蠕虫攻击简史
- 6.4.5 蠕虫技术的现状
- 6.4.6 移动代码
- 6.4.7 手机蠕虫
- 6.4.8 客户端漏洞和路过式下载
- 6.4.9 点击劫持

6.4 传播-漏洞利用-蠕虫

- 主动寻找并感染其他机器的程序，而每台被感染机器又转而成为自动攻击其他机器的跳板
- 利用存在于客户端或服务器程序中的漏洞
- 可以使用网络连接在系统之间传播
- 通过共享媒体传播（USB驱动器、CD、DVD数据盘）
- 电子邮件蠕虫通过附带的文档或即时消息通信中的宏或脚本代码传播
- 激活后，蠕虫可能会再次复制和传播
- 通常携带某种形式的有效载荷
- 第一个著名的蠕虫程序是于20世纪80年代早期在Xerox Palo Alto实验室中实现的

蠕虫复制

电子邮件或即时通信工具	• 蠕虫通过邮件将自己的拷贝发送到其他系统中去，或者将自身当作即时通信服务的附件进行发送
文件共享	• 蠕虫可以在如USB设备等可插拔媒体上创建自己的拷贝，或像病毒那样感染此类媒体上适合的文件
远程执行能力	• 蠕虫在其他系统中执行自己的拷贝的能力
远程文件访问或传输能力	• 蠕虫利用远程文件访问或者传输服务向其他系统复制自身拷贝，该系统的用户此后便有可能执行它
远程登录能力	• 蠕虫以一个用户的身份登录到远程系统，然后使用命令将自己拷贝到将要被执行的另一个系统中

6.4.1 发现目标

- 网络蠕虫在传播阶段的首要功能是寻找其他系统进行感染，这个过程可以叫做扫描(scanning)或指纹采集（fingerprinting）。
- 对使用远程访问的网络服务来攻击软件漏洞的蠕虫而言，它必须先明确找出潜在的运行有易感染服务的系统，然后再进行感染。
- 已经安装在被感染机器上的蠕虫将重复相同的扫描过程，直到被感染机器建立成一个大型的分布式网络。

网络地址扫描方式

随机式探索

- 每个受损主机使用不同的种子探测IP地址空间中的随机地址
- 这会产生大量的互联网流量，甚至在发起实际攻击之前就可能造成普遍中断

黑名单

- 攻击者首先为潜在的易感染机器列出一个大名单
- 一旦名单编辑完成，攻击者即开始感染名单中的机器
- 每个被感染的机器会被分配名单中的一部分进行扫描
- 这导致扫描时间很短，这可能使检测感染发生变得困难

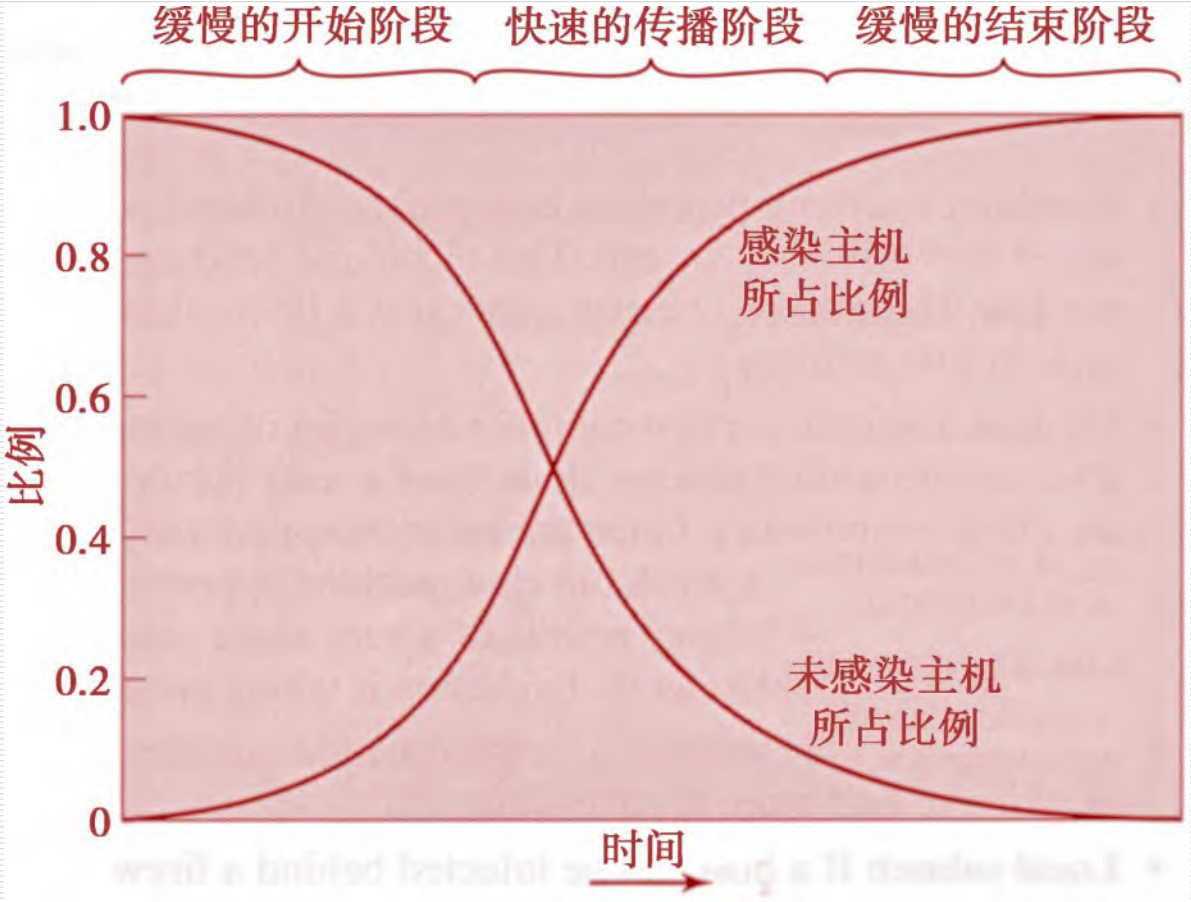
拓扑式探索

- 利用被感染机器中所包含的信息来寻找和扫描更多的主机

本地子网

- 如果防火墙后的一台主机可以被感染，则该主机会在其所在的本地网络中寻找目标。
- 利用子网地址结构，被感染的主机可以寻找到其他本应收到防火墙保护的主机。

6.4.2 蠕虫传播模型


缓慢的开始阶段
快速的传播阶段
缓慢的结束阶段
1.0
0.8
0.6
0.4
0.2
0
比例
感染主机
所占比例
未感染主机
所占比例
时间

6.4.3 Morris蠕虫

- 早期重要蠕虫感染
- 由Robert Morris于1988年发布
- 为了在UNIX系统中传播而设计的
  - 试图破解本地密码文件以使用合法用户的身份登录到其他系统
  - 利用finger协议中的漏洞报告远程用户的行踪
  - 利用负责收发邮件的远程进程的调试选项中的一个陷门
- 成功的攻击实现了与操作系统命令解释器的通信
  - 向解释器发送引导程序以复制蠕虫

6.4.4 蠕虫攻击简史

Melissa	1998	电子邮件蠕虫 首先将病毒、蠕虫和特洛伊木马包含在一个软件包中
Code Red	2001.07	利用Microsoft IIS漏洞 探测随机IP地址 活动时会消耗大量网络容量
Code Red II	2001.08	还针对Microsoft IIS 安装后门以进行访问
Nimda	2001.09	具有蠕虫、病毒和移动代码特征 使用电子邮件、Windows共享、Web服务器、Web客户端、后门进行传播
SQL Slammer	2003早期	利用SQL server中的缓冲区溢出漏洞进行攻击 简短且迅速扩展
Sobig.F	2003后期	利用开放式代理服务器将受感染的机器变成垃圾邮件引擎
Mydoom	2004	群发邮件蠕虫 在受感染的计算机中安装后门
Warezov	2006	在系统目录中创建可执行文件 将自身作为电子邮件附件发送 可以禁用安全相关产品
Conficker (Downadup)	November 2008	利用Windows缓冲区溢出漏洞 自SQL Slammer以来最广泛的感染
Stuxnet	2010	限制传播速度以减少检测机会 目标工业控制系统

6.4.4 蠕虫攻击简史

- 2017年5月，WannaCry勒索软件攻击蔓延速度非常快，数小时至数天，感染了150多个国家的公共和私人组织的数十万个系统。
- 它通过积极扫描本地和随机远程网络来传播蠕虫，试图利用未修补的Windows系统上的SMB文件共享服务中的漏洞。
- 这种快速传播只是由英国安全研究人员意外激活“致命转换”命令而放缓，该研究人员在该恶意软件的初始版本中检查了其存在。
- 一旦将其安装在被感染的系统，它同样会加密文件，并索要赎金来恢复它们。

6.4.5 蠕虫技术的现状

多平台（multiplatform）：可以攻击多种平台

多种攻击手段（multiexploit）：新的蠕虫会使用多种方法对系统进行渗透

超快速传播（ultrafast spreading）：使用多种技术手段优化蠕虫的传播速率，尽可能在短时间内感染尽可能多的机器。

多态（polymorphic）：每个蠕虫的拷贝都能够利用在功能上等价的指令和加密技术来生成新的代码。

变形（metamorphic）：除了改变自身形态外，变形蠕虫还根据其行为模式库在传播的不同阶段表现出不同的行为。

传输载体（transport vehicle）：因为蠕虫能够迅速地感染大量系统，因此它们是传播其他分布式攻击程序的理想载体。

0-day攻击（zero-day exploit）：为获得最大的震动和扩散范围，蠕虫会利用未为人知的漏洞。这种漏洞只有在蠕虫发起攻击时，才会被网络公众所发现。

6.4.6 移动代码

- NIST SP 800-28将移动代码定义为
  - “可以不加修改就能够在不同系统平台上运行、并且能够实现相同功能的程序”
- 从远程系统传输到本地系统，然后在本地系统上执行
- 通常作为病毒、蠕虫或特洛伊木马的机制
- 利用漏洞执行自己的漏洞攻击
- 受欢迎的载体包括：Java applet、ActiveX、JavaScript、VBScript
- 在本地系统上使用移动代码进行恶意操作的最常见方式有：
  - 跨站点脚本
  - 交互式和动态网站
  - 电子邮件附件
  - 从不可信网站下载程序或者下载不可信软件

6.4.7 手机蠕虫

- 第一个发现是2004年的卡比尔蠕虫
- 2005年，Lasco和CommWarrior
- 通过蓝牙无线连接或彩信进行通信
- 目标是智能手机
- 可以使手机完全瘫痪、删除手机数据、或者向收取额外费用的号码发送信息
- CommWarrior蠕虫利用蓝牙技术向接受区域内的其他手机传播。它也以彩信的方式向手机通讯录中的号码发送自己的拷贝，而且会自动回复收到的短信和彩信。

6.4.8 客户端漏洞和路过式下载

- 另一种攻击软件漏洞的方式是利用应用程序中的缺陷(bug)来安装恶意软件，最普通的一种技术是路过式下载。
- **路过式下载：利用了浏览器的漏洞，使得当用户浏览一个受攻击者控制的Web页面时，该页面包含的代码会攻击该浏览器的缺陷并在用户不知情或未允许的情况下向系统安装恶意软件。**
- 在大多数情况下，恶意软件不会像蠕虫那样主动传播
- 当用户访问恶意网页时传播

路过式下载变种——水坑式攻击

水坑式攻击(watering-hole attack)

- 一种用于高针对性攻击的下载驱动变体
- 攻击者研究他们的目标受害者，以确定他们可能访问的网站，然后扫描这些站点找出那些含有能让他们植入路过式下载的漏洞
- 然后，他们等待其中一名受害者访问其中一个有害的站点
- 他们的攻击代码甚至可以被设定为只感染属于目标组织的系统，而对其他浏览该站点的访问者没有影响
- 极大增加了受控制站点无法被检测出来的可能性

恶意广告(Malvertising)

- 通过web站点部署恶意软件的技术，该技术不会真正损害web站点
- 攻击者在他们目标网站付钱植入包含有恶意代码的广告。
- 利用这些恶意植入代码，攻击者通过向访问者展示广告来令其感染
- 这些恶意代码是动态生成的，同样可以减少被侦测到的机会，或者只感染特殊的系统
- 近年来，恶意广告发展迅速，因为它们很容易被放置在期望的网站上，而且几乎没有问题，也很难追踪
- 攻击者在预料到他们的受害者可能会浏览目标网站后仅仅将这些恶意广告放置几小时，以此来大幅度降低恶意广告的可见度

6.4.9 点击劫持

也称为用户界面（UI）伪装攻击

使用类似的技术，键盘输入也可以被劫持

- 用户会被误导而以为他们在为电邮或银行账户输入口令，而实际上他们将口令输入到了攻击者控制的一个无形的框架内

攻击者收集被感染用户鼠标点击信息的攻击

- 攻击者可以强迫用户做一系列的事情，从调整计算机的设置到在用户不知情的情况下让用户访问可能含有恶意代码的网站
- 利用Adobe Flash和JavaScript，攻击者甚至可能在一个合法按钮的上面或者下面部署一个按钮，并将其制作成难以被用户察觉的样子
- 典型例子是利用多重透明或模糊的页面层次来欺骗用户在试图点击最上层页面时，却实际上点击了另一个按钮或链接到另一个页面
- 攻击者实施点击劫持的意图是将一个页面链接至属于其他应用、域名（也许两者都是）的页面

6.5 传播-社会工程学-垃圾电子邮件、木马

围绕社会工程学讨论，其“欺骗”用户协助损害他们自己的系统或个人信息

垃圾电子邮件

- 大量的不请自来的电子邮件
- 恶意软件的重要载体
- 用于网络钓鱼攻击

特洛伊木马

- 一个有用的或者表面上看起来有用的程序
- 用于完成攻击者无法直接完成的功能

手机木马

- 首次被发现是在2004年
- 其目标是智能手机

6.6 载荷-系统损坏

- 6.6.1 数据损坏和勒索软件
- 6.6.2 物理损害和逻辑炸弹

6.6.1 数据损坏和勒索软件

CIH(Chernobyl)病毒

- 首次发现于1998年，是一个有破坏性的，寄生性的，内存驻留性质的病毒，运行于Windows 95和98系统上
- 它会在可执行文件被打开时感染他们。当触发日期一到，它会通过重写硬盘从0开始的第一个兆字节的数据以删除被感染系统中的数据，导致整个文件系统的大面积损坏

6.6.1 数据损坏和勒索软件

求职信(Klez)蠕虫

- 感染波及从Windows 95到Windows XP的一系列操作系统，首次发现于2001年10月，通过电子邮件，向用户地址薄中的邮箱地址和系统中的文件传播自身的拷贝
- 它可以暂停和删除一些运行在系统中的反病毒程序
- 在发作日期，会清空本地硬盘下的文件。

6.6.1 数据损坏和勒索软件

勒索软件(ransomware)

- 加密用户数据，然后向用户索要赎金才可以恢复数据
- 1989年发现的Cyborg木马
- 到了2006年年中，涌现出一批使用公钥加密算法和越来越长的密钥对数据进行加密的蠕虫和木马（如Gpcode木马）
- 用户必须支付赎金，或在指定网站进行支付才可以拿到解密的密钥

WannaCry

- 2017年5月在许多国家感染了大量系统
- 当安装在受感染的系统上时，它会加密大量满足列表中文件类型要求的文件，而后索要比特币赎金来恢复它们
- 通常，只有当组织有良好的备份和适当的事件响应和灾难恢复计划时，才可能恢复此信息
- 目标从个人计算机系统扩展到移动设备和Linux服务器
- 有时会威胁发布敏感个人信息或在短时间后永久销毁加密密钥等策略来增加受害者的支付压力

6.6.2 物理损害和逻辑炸弹

- **真实世界的损害**
  - 对物理设备造成损坏
    - Chernoby1病毒重写BIOS代码
  - Stuxnet 蠕虫
    - 针对特定的工业控制系统软件
  - 人们担心使用复杂的有针对性的恶意软件进行工业破坏
- **逻辑炸弹**
  - 嵌入在恶意软件中的代码，在特定条件满足时便会 “爆炸”

6.7 载荷-攻击代理-zombie、bot

- 6.7.1 bot的用途
- 6.7.2 远程控制功能

6.7.1 bot的用途

- 秘密地控制一台连接Internet的计算机，并利用所控制的计算机发动攻击
- 僵尸网络-能够协调行动的僵尸机的集合
- 使用：
  - 分布式拒绝服务（DDoS）攻击
  - 发送垃圾邮件
  - 嗅探通信流量
  - 记录键盘
  - 传播新的恶意软件
  - 安装广告插件和浏览器辅助插件
  - 攻击IRC聊天网络
  - 操纵在线投票/游戏

6.7.2 远程控制功能

- bot与蠕虫的区别所在
  - 蠕虫会自我复制并自我激活
  - bot是由某种形式的指挥控制(C&C)服务器网络控制
- 早期实现远程控制的工具是IRC服务器
  - 所有的bot都会加入这个服务器的一个特定通道中，并把通道中收到的消息当作命令处理
  - 近来越来越多的僵尸网络通过HTTP等协议使用隐蔽通信通道
  - 分布式控制机制使用对等协议来避免单点故障

6.8 载荷-信息窃取-键盘记录器、网络钓鱼、间谍软件

- 6.8.1 凭借盗窃、键盘记录器和间谍软件
- 6.8.2 网络钓鱼和身份盗窃
- 6.8.3 侦察、间谍和数据渗漏

6.8.1 凭借盗窃、键盘记录器和间谍软件

键盘记录器

- 抓取被感染机器中的键击信息，从而允许攻击者监视那些敏感信息
- 通常使用一些过滤机制以便只记录与攻击者想要的关键字相近的信息

间谍软件

- 监听受害系统中更多种类的活动
  - 监视浏览活动的历史和内容
  - 将某些网页请求重定向到虚假网站
  - 动态修改浏览器和网站的交换数据

6.8.2 网络钓鱼和身份盗窃

利用社会工程学，伪装成可信来源的通信取得用户的信任

- 在垃圾邮件中包含URL，该URL链接到模仿银行、游戏或类似网站登录页面的虚假网站
- 提示用户需要紧急验证他们的账户以免被锁定的消息
- 攻击者使用捕获的凭据利用该帐户进行攻击

鱼叉式网络钓鱼(spear-phishing)

- 攻击者会仔细研究收件人
- 电子邮件是专门为适合其收件人而设计的，通常引用一系列信息以使他们相信其真实性

6.9 载荷-隐蔽-后门、rootkit

- 6.9.1 后门
- 6.9.2 rootkit
- 6.9.3 内核模式下的rootkit

6.9.1 后门

- 也称为陷门（trapdoor）
- 程序的秘密入口点，使得知情者不经过通常的安全访问程序而获取访问权限
- 维护挂钩（maintenance hook）是程序员用来调试和测试程序的后门
- 难以在应用程序中为后门实现操作系统控制

6.9.2 rootkit

- 安装在系统上的一组隐藏程序，用于维护对该系统的秘密访问
- 通过破坏对计算机上的进程、文件和注册表进行监视和报告的机制来隐藏
- 向攻击者授予管理员（或root）权限
- 可以添加或更改程序和文件，监视进程，发送和接收网络流量，并根据需要获得后门访问

Rootkit分类

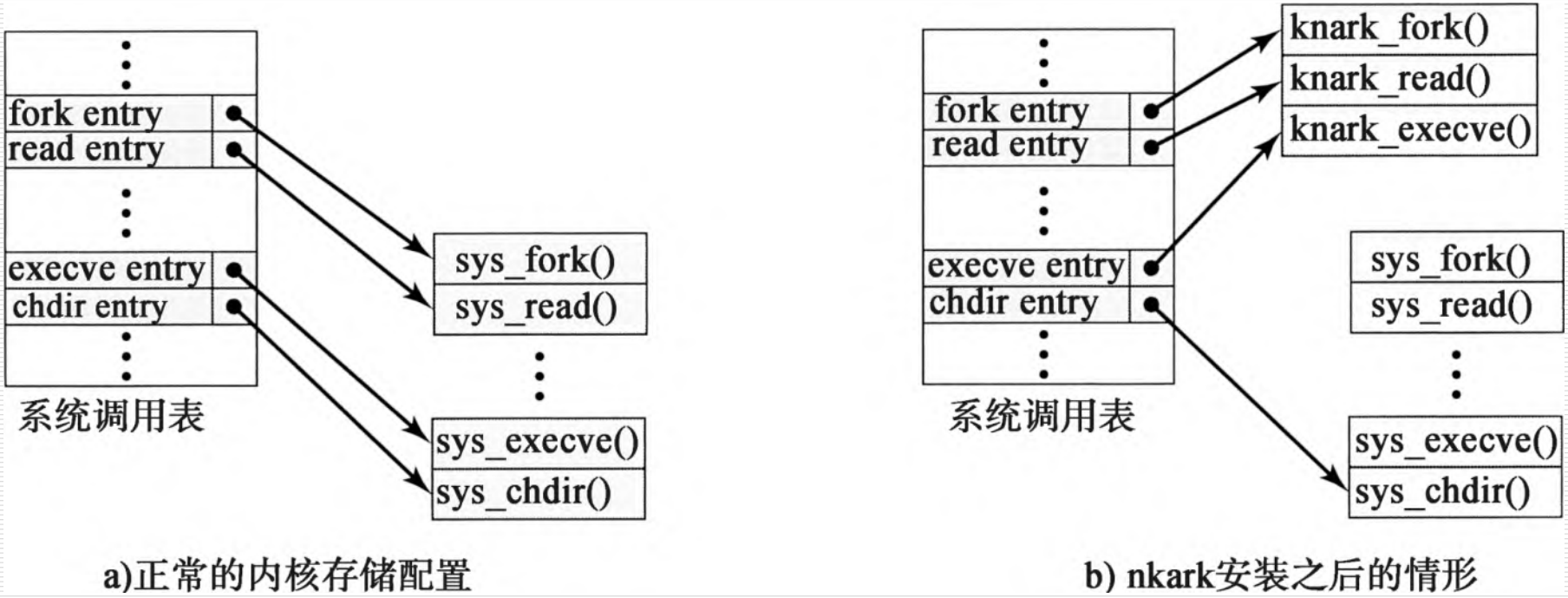
- 持续的（persistent）：系统每一次启动都会被激活。Rootkit必须把它的代码存储在持续性存储如注册表或文件系统中，并配置一种方式使它不需要用户干预就可以自己执行。这意味着更容易检测，因为持久存储中的副本可能会被扫描。
- 基于内存的（memory based）：没有持续性，重起后rootkit就会失效。但是，因为它只存在于内存中，所以很难发现。
- 用户模式（user mode）：截获API（应用程序接口）调用，并修改返回值。

Rootkit分类

- 内核模式（kernel mode）：能够截获对本地内核模式的API的调用。Rootkit还能够通过删除内核的活动进程列表中的恶意软件进程来隐藏自己。
- 基于虚拟机的（virtual machine based）：此类rootkit会首先安装一个轻量级虚拟机监视器，然后在监视器之上的虚拟机中运行操作系统。这样一来，rootkit就能够对已经虚拟化的系统中所发生的状态和事件进行透明地截获和修改。
- 外部模式(external mode)：将恶意软件植入目标系统的正常运行模式之外，如机器的BIOS中或系统管理模式中等，这样它就可以直接获得硬件的访问权限。

6.9.3 内核模式下的rootkit

下图为rootkit对系统调用表的修改



修改系统调用的技术

修改系统调用表（modify the system call table）	• 黑客修改存储在系统调用表中的选定的系统调用的地址。这样rootkit就把系统调用从原来合法的例程指向了rootkit所指定的程序上。
修改系统调用表的目标对象（modify system call table targets）	• 黑客用恶意代码覆盖了所选定的正常系统调用例程，而系统调用表没有被修改。
重定向系统调用表（redirect the system call table）	• 黑客把对整个系统调用表的引用重定向到新的内核存储单元中的一个新表上。

6.10 对抗手段

- 6.10.1 针对恶意软件的对抗措施
- 6.10.2 基于主机的扫描器和基于签名的反病毒软件
- 6.10.3 边界扫描处理
- 6.10.4 分布式情报收集处理

6.10.1 针对恶意软件的对抗措施

- 理想的应对恶意软件威胁的方法是预防：首当其冲的是阻止恶意软件进入计算机系统，然后是阻止其修改计算机系统。
- 采取适当的措施以强化系统和用户防止恶意软件感染的能力的确可以极大地减少其攻击的成功率。
- **NIST SP 800-83**提出了恶意软件预防措施的4个主要元素：**规则，警惕性，弥补弱点和缓解威胁。**
- 拥有一个解决恶意软件防御的合适的策略为采取适当的预防措施提供依据。

6.10.1 针对恶意软件的对抗措施

如果预防措施失败，针对恶意软件威胁还存在以下由各种技术性手段所支持的缓解措施：

- 检测（detection）：一旦被感染，就马上确定恶意软件的存在并对其定位。
- 识别（identification）：一旦检测到恶意软件，立即识别出是何种恶意软件感染了系统。
- 清除（removal）：一旦识别出恶意软件类型，立刻清除恶意代码在被感染的系统中的所有痕迹，以阻止其继续扩散。

6.10.2 基于主机的扫描器和基于签名的反病毒软件

反病毒软件的发展，如下表：

第一代：简单的扫描器	第二代：启发式扫描器	第三代：活动陷阱	第四代：全面的保护
需要病毒特征码来识别病毒 局限于检测已知病毒	通过启发式规则来检测可能存在的病毒感染 另一种方法是完整性检查	通过病毒行为来识别病毒而不是通过被感染文件的内部结构特征	综合运用各种反病毒技术的软件包 包括扫描和活动陷阱组件，同时还加入了访问控制功能

6.10.2 基于主机的扫描器和基于签名的反病毒软件

沙箱分析

- 在模拟沙箱或虚拟机上运行潜在恶意代码
- 允许代码在受控环境中执行，在该环境中可以密切监视其行为，而不会威胁真实系统的安全
- 在此类环境中运行潜在的恶意软件可以让检测系统对恶意软件复杂加密、多态或变质进行检测
- 沙箱分析设计最困难的部分是确定执行每次解释执行的时间

6.10.2 基于主机的扫描器和基于签名的反病毒软件

基于主机的动态恶意软件分析

- **与主机的操作系统集成，实时监视程序行为以防恶意操作**
  - 在程序的恶意行为影响计算机之前将其阻断
  - 能及时阻断可疑软件的执行，它比起现有的反病毒检测技术（如特征码技术和启发式技术）具有很大的优势
- **局限性**
  - 因为在恶意程序的所有行为被识别出来之前，该程序已经在目标机器上执行了，所以在它被检测并阻止之前就可能已经对系统造成了损害。

6.10.3 边界扫描处理

- 反病毒软件通常包含在运行这些系统上的电子邮件和Web代理服务中。
- 也可能包含在入侵检测系统的流量分析组件中。
- 也可能包含有预防入侵的措施，有能力屏蔽任何可疑的网络流量。
- 该方法仅限于扫描恶意软件内容。

监控软件类型

入口监控软件

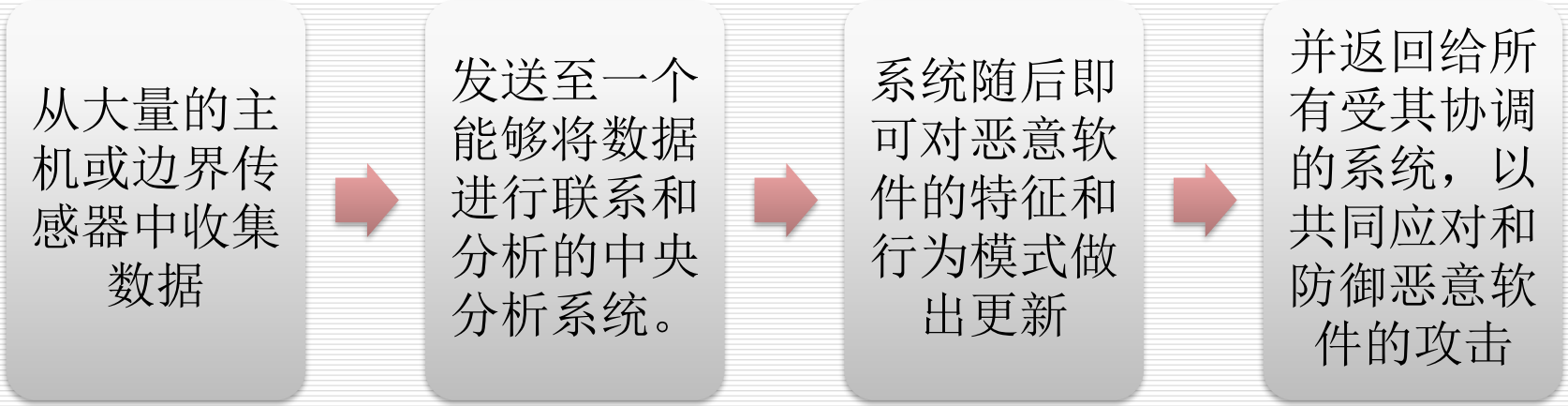
- 被安装在企业内部网络和Internet之间
- 一个例子是监测那些针对未被使用的本地IP地址的输入通信流量

出口监控软件

- 被安装在企业内部网络中的各个独立局域网的出口点上，也可以被安装在企业内部网与Internet之间
- 监控输出通信中是否存在扫描或者其他可疑行为来捕获恶意软件攻击的源头

6.10.4 分布式情报收集处理

- 使用反病毒软件的最后一种方式是对其进行分布式配置。



- 实际上是分布式入侵防御系统的一种特殊范例。

总结

- **恶意软件（malware）的类型**
  - 恶意软件粗略的分类
  - 攻击工具包
- **高级持续性威胁（APT）**
- **传播-感染的内容-病毒**
  - 病毒的性质
  - 宏病毒和脚本病毒
  - 病毒的分类
- **传播-漏洞利用-蠕虫**
  - 发现目标
  - 蠕虫传播模型
  - Morris蠕虫
  - 蠕虫攻击简史
  - 蠕虫技术现状
  - 移动代码
  - 手机蠕虫
  - 客户端漏洞和路过式下载
  - 点击劫持
- **传播-社会工程学-垃圾电子邮件，木马**
  - 垃圾(大量不请自来的)电子邮件
  - 特洛伊木马
  - 手机木马
- **载荷-系统损坏**
  - 数据损坏和勒索软件
  - 物理损害和逻辑炸弹
- **载荷-攻击代理-僵尸程序（Zombie，bots）**
  - bot的使用
  - 远程控制功能
- **载荷-信息窃取-键盘记录器，网络钓鱼，间谍软件**
  - 凭证盗窃，键盘记录器和间谍软件
  - 网络钓鱼和身份盗窃
  - 侦查、间谍活动和数据渗漏
- **载荷-隐蔽-后门，Rootkits**
  - 后门
  - Rootkit
  - 内核模式下的rootkit
  - 虚拟机和其他外部rootkit
- **对抗手段**
  - 恶意软件的对抗措施
  - 基于主机的扫描器和基于签名的反病毒软件
  - 边界扫描方法
  - 分布式情报收集方法

谢谢各位！