3-2 恶意软件的原理与防护

2019年4月10日 23:34

5 木马

5.1 木马的基本概念

- 通过欺骗或诱骗的方式安装，并在用户的计算机中隐藏以实现控制用户计算机的目的。
  - 具有远程控制、信息窃取、破坏等功能的恶意代码；
- 特点
  - 欺骗性、隐藏性、非授权性、交互性

5.2 木马的分类

- 行为视角（粒度细，如卡巴斯基 SafeStream病毒库的分类标准库的分类标准）
- 功能视角
  - 远程控制型木马

        - 远程控制
        - 交互性：双向（攻击者 <->被控制端）
        - 典型案例： 卡巴斯基分类标准下的木马子类： Backdoor
    - 信息获取型木马
        - 功能：信息获取:键盘输入，内存，文件数据等
        - 交互性：单向交互（攻击者 <-被控制端）
        - 发送至三方空间，文件服务器、指定邮箱等，或者直接开启 FTP服务程序等
        - 典型案例： 卡巴斯基分类标准下的 Trojan-Bank、 Trojan-GameThief、 Trojan-IM、 Trojan-Spy、 Trojan-PSW、 Trojan-Mailfinder等。
    - 破坏型木马
        - 功能：对本地或远程主机系统进行数据破坏、资源消耗等。
        - 交互性：单向（攻击者 ->被控制端），或无交互
        - 典型案例： 卡巴斯基分类标准下的 Trojan-DDoS、 Trojan-Ransom、 Trojan-ArcBomb、 Trojan-Downloader、 Trojan-Dropper等。

5.3 木马的植入方式

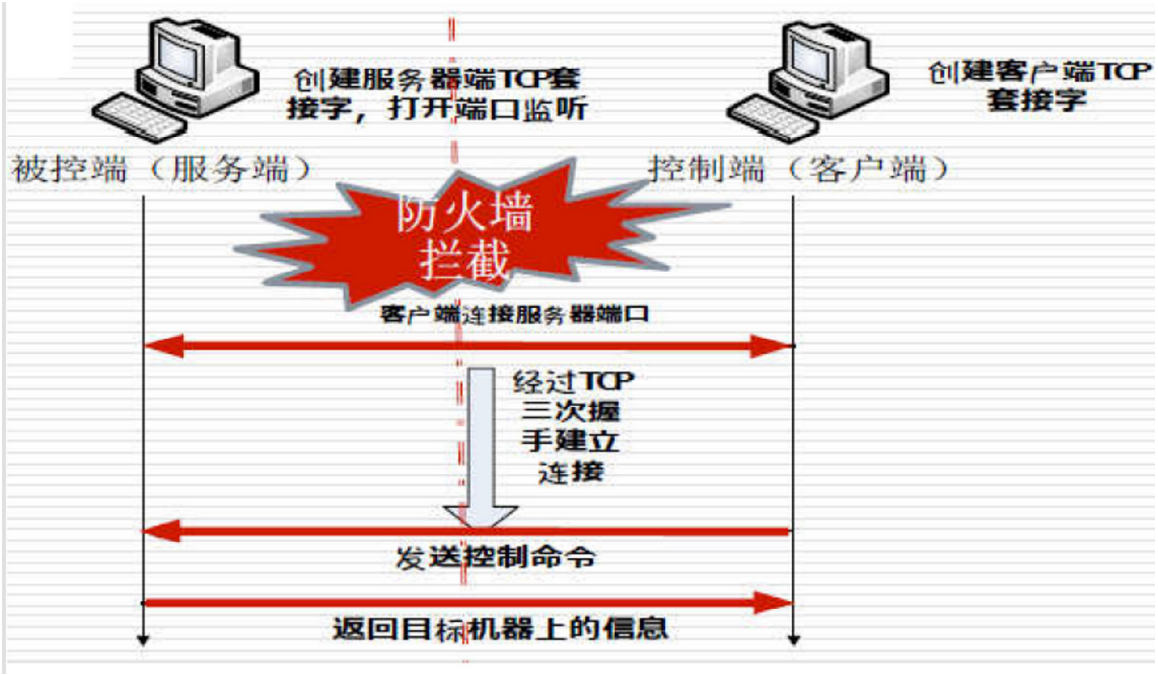
- 网页挂马植入
    - 访问网页后通过浏览器的缓冲区溢出漏洞实现自动下载安装
- 电子邮件植入
    - 附件形式，打开附件被植入
    - 电子邮件与恶意网页相结合，即使不打开附件，选中就会被植入（以 HTML格式发送）
- 文档捆绑植入
    - office文档、 pdf文档漏洞等
- 伪装欺骗植入
    - 更改后缀名（ Unicode翻转字符）、图标伪装
- 捆绑植入
    - EXE捆绑、文档嵌入、多媒体文件、电子书植入
- 其他
    - 特定 U盘植入（故意丢弃、或者工作 U盘、数据拷贝等）
    - 社会工程

5.4 木马的通信方式

- 传输通道构建信息
    - IP地址、端口等信息、第三方网站地址
- 建立通信连接的方式
    - 正向连接 反向连接

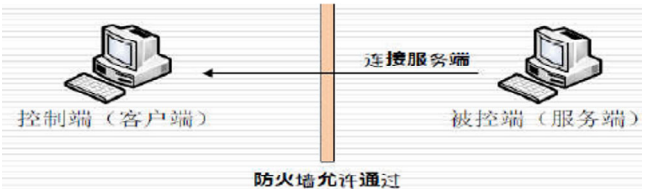
正向连接

- 控制端主动连接被控端

- 

- 优点：
  - 攻击者无需外部 IP地址
  - 木马样本不会泄露攻击者 IP地址
- 缺点
  - 可能被防火墙阻挡
  - 被攻击者必须具备外部 IP地址
  - 定位被攻击者相对困难
  - 被攻击者 IP是否变化？ 目标主机何时上线?

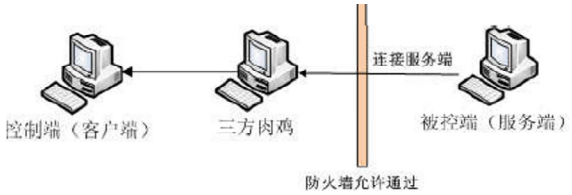
反向连接 -1

- 被控制端主动连接控制端
- 

- 优点：
  - 通过防火墙相对容易
  - 攻击目标随时上线、随时控制
  - 可以控制局域网内的目标
- 缺点：
  - 样本会暴露控制服务器信息（域名或 IP）
  - 攻击者通常应当具有外部 IP

反向连接 -2

- 被控制端和控制端和第三方通信
  - 第三方一般是Web服务器、或已经被控制的肉鸡

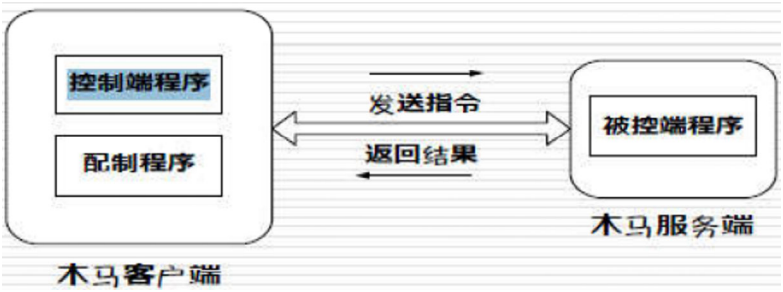
- 

- 优点
  - 可绕过防火墙，自动连接上线，不易被发现（代理）
- 缺点
  - 肉鸡的稳定性需要保障

通信协议

- TCP协议
  - 稳定、易被发现
  - HTTP协议伪装
- UDP协议
  - 和 TCP一样也有正向、反向两种方式
  - 负载比 TCP少，但是可靠性低
- ICMP + TCP / UDP
  - 监听 ICMP报文，以感知木马数据
  - ICMP报文是由系统内核或进程直接处理而不是通过端口
  - 一般不会被防火墙过滤

5.5 远控木马的常见功能

- 木马结构
  - 完整的木马一般由木马配置程序、控制端程序（客户端）和被控制端程序（服务端程序）等三部分组成。
- 

远控木马的常见功能

- 文件管理
  - 获取目标的文件系统信息
  - 功能: 对磁盘文件的浏览/上传下载/执行/删除/修改
- 进程管理
  - 查看、结束或者暂停目标系统进程
  - 功能:
    - 查看目标系统的环境信息 --安装了哪些软件？目前对方正在做什么？
    - 停止或暂停目标系统的相关程序 --如反病毒程序

- 服务管理
  - 查看并管理目标系统的服务
  - 功能：对服务的创建/启动 /停止/删除
  - 作用：
    - 查看目标系统的环境信息 --安装了哪些软件？启动了哪些服务？
    - 停止或暂停目标系统的相关程序 --如反病毒程序
- 注册表管理
- 屏幕控制
- 屏幕截取
- 语音视频截获
- 语音监听
- 键盘记录
- 窗口管理
  - 查看目标主机目前开启了哪些窗口 了解目标用户正在做什么？
- 远程 Shell
  - 交互式或非交互式 Shell
  - 远程交互的 Cmdexe 直接执行命令或第三方程序

木马的关键

- 功能适当［精简灵活］
- 适用性强［功能、权限］
- 高效、稳定、隐蔽［传输］
- 可穿透性
- 自更新、自销毁
- 防追踪、反制对抗
- 持续免杀性能等
  - 特征值、通用主机行为、异常的通信流量

6 恶意代码检测技术

6.1 恶意代码检测对象与策略

- 恶意代码的检测是将检测对象与恶意代码特征（检测标准）进行对比分析，定位病毒程序或代码，或检测恶意行为
- 检测对象主要包括

    - 引导扇区
    - 文件系统中可能带毒的文件
    - 内存空间
    - 主板 BIOS等
    - 网络流量，系统行为等

检测对象 1：引导扇区

- 具有控制权的引导扇区
    - 硬盘主引导扇区
    - 硬盘操作系统引导扇区
    - 可移动硬盘引导扇区
- 检测目标
    - 引导扇区病毒、 MBR木马等

检测对象 2：可能带毒的文件

- 可执行程序
- 数据文件
- 脚本文件
- 网页文件

检测对象 3：内存空间

- 恶意代码在传染或执行时，必然要占有一定的内存空间，部分功能代码驻留在内存中。
    - 部分恶意代码仅存在于内存中
        - 无文件存在，或已自行删除
        - 或被外部动态按需注入
    - 部分恶意代码仅在内存中被还原

病毒的检测策略

- 专用检查技术：针对某个或某些特定已知恶意代码。
    - 反病毒软件必须随着新病毒的不断出现而频繁更新病毒库版本
    - 如文件特征值检测技术
- 通用检测技术：针对已知和未知恶意代码
    - 广义特征描述或一般行为特征作为判定依据
    - 如启发式扫描技术、主动防御技术等。

6.2 特征值检测技术

- 病毒特征值是反病毒软件鉴别特定计算机病毒的一种标志。通常是从病毒样本中提取的一段或多段字符串或二进制串。
- 具体思路 :获取样本-> 提取样本特征-> 更新病毒库-> 查杀病毒
- 特征值检测的两个关键

    - 特征码的提取
    - 特征值的匹配技术：
        - 模式匹配
        - AC算法--有限自动机的多模式匹配算法
- 特征值的提取选择
    - 特征字串：从计算机病毒体内提取、为病毒所特有的特征串。如特定提示信息，特定签名信息等。
    - 感染标记：病毒为避免重复感染而使用的感染标记
    - 从病毒代码的特定地方开始取出连续的、不大于 64且不含空格（ ASCII值为 32）的字节串
- 提取方法
    - 人工提取
        - 反病毒工程师对病毒样本进行分析后，人工确定病毒特征
    - 自动提取
        - 通过软件系统自动提取特定范围内特定长度且具有一定特征的数据
        - 处理不利则可能被别有用心者利用，形成误杀
- 优点：检测速度快、误报率低等优点，为广发反病毒厂商所采用，技术也比较成熟
- 缺点：只能检测已知恶意代码。容易被免杀绕过。
- 针对特征值检测技术，恶意软件如何对抗?
    - 手工修改自身特征 首选利用反病毒软件定位 然后，进行针对性修改
    - 自动修改自身特征 加密、多态、变形等

6.3 校验和检测技术 —预期符合性

- 校验和检测技术：在文件使用 /系统启动过程中，检查检测对象的实际校验和与预期是否一致，因而可以发现文件 /引导区是否感染。
- 预期：正常文件内容和正常引导扇区数据
- 静态可信：可信计算机对主引导扇区和一些系统关键程序进行了校验，从而保障系统启动之后的初始安全

运用校验和检测技术查病毒采用三种方式

- 系统自动监测：将校验和检查程序常驻内存，每当应用程序开始运行时，自动核验当前与预先保存的校验和是否一致。
- 专用检测工具：对被查的对象文件计算其正常状态的校验和，将校验和写入被查文件中或检测工具中，而后进行比较，如 MD5Checker。
- 自我检测：在应用程序中，放入校验和检测技术自我检查功能，将文件正常状态的校验和写入文件自身，应用程序启动比较现行校验和与原校验和值，实现应用程序的自检测。如 QQ软件。

校验和检测对象

- 文件头部

    - 一般比较整个文件效率较低，有的检测仅比较文件的头部
    - 现有大多数寄生病毒需要改变宿主程序的头部
- 文件基本属性
    - 文件的基本属性包括文件长度、文件创建日期和时间、文件属性（一般属性、只读属性、隐含属性、系统属性）、文件的首簇号等
- 文件内容
    - 对文件内容（可含文件的属性）的全部字节进行某种函数运算，这种运算所产生的适当字节长度的结果就叫做校验和。
    - 这种校验和在很大程度上代表了源文件的特征，一般文件的任何变化都可以反映在校验和中。 可以采用一些散列函数，如 MD5 CRC校验
- 系统数据
    - 病毒可能修改相对固定的重要系统数据
        - 如硬盘引导扇区、分区引导扇区、内存中断向量表、 SSDT、设备驱动程序处理例程等。

校验和检测技术优缺点

- 优点
    - 方法简单
    - 能发现未知病毒
    - 目标文件的细微变化也能发现
- 缺点
    - 必须预先记录正常文件的校验和（预期）
    - 误报率高
    - 不能识别病毒名称
    - 效率低

6.4 启发式扫描技术-恶意代码检测经验和知识的软件实现

- 可疑的程序代码指令序列
    - 格式化磁盘类操作
    - 搜索和定位各种可执行程序的操作
    - 实现驻留内存的操作
    - 发现非常用的或未公开的系统功能调用的操作、子程序调用中只执行入栈操作、远距离（超过文件长度三分之二）跳转指令等
    - 敏感系统行为
    - 敏感 API函数（序列）调用功能
- 启发式扫描步骤
    a. 定义通用可疑特征（指令序列或行为）
    b. 对上述功能操作将被按照安全和可疑的等级进行排序，授以不同的权值
    c. 鉴别特征，如果程序的权值总和超过一个事先定义的阈值，则认为“发现病毒”
- 优点 :能够发现未知病毒

- 缺点 :误报率高
- 解决方案
  - 启发式扫描技术 +传统扫描技术
  - 可提高病毒检测软件的检测率，同时有效降低了总的误报率

6.5 虚拟机检测技术

- 在反病毒系统中设置的一种程序机制，他能在内存中模拟一个小的封闭程序执行环境，所有待查文件都以解释方式在其中被虚拟执行行。通常虚拟执行一小部分代码即可
- 为什么需要虚拟机检测技术?
  - 加密、多态、变形病毒的出现
- 加密病毒
  - 真实代码被压缩或加密，但最终需要在内存中还原
- 优点
  - 有效处理加密类病毒
  - 虚拟机技术 +特征值扫描，准确率更高
  - 虚拟机技术 +启发式扫描，有利于检测未知变形病毒

6.6 主动防御技术-行为监控等技术

- 动态监视所运行程序调用各种应用编程接口（ API）的动作，自动分析程序动作之间的逻辑关系，自动判定程序行为的合法性。
- 监控应用程序的敏感行为，并向用户发出提示，供用户选择。
- 常见可疑行为
  - 对可执行文件进行写操作
  - 写磁盘引导区
  - 病毒程序与宿主程序的切换
  - 写注册表启动键值
  - 远程线程插入
  - 安装、加载驱动
  - 键盘钩子
  - 自我隐藏
  - 下载并执行
- 优点：可发现未知恶意软件、可准确地发现未知恶意软件的恶意行为
- 缺点：可能误报、不能识别恶意软件名称,以及在实现时有一定难度。

7 恶意软件样本捕获与分析

7.1 恶意软件样本捕获方法

- 蜜罐
- 用户上报
- 云查杀平台
- 诱饵邮箱
- 样本共享

蜜罐

- 通常是指未采取安全防范措施、并且将模拟的程序漏洞主动暴露在网络中的计算机
- 引诱恶意软件样本来攻击这类蜜罐计算机设备。
- 特点
    - 与一般计算机不同，其内部运行着多种多样特殊用途的“自我暴露程序”和行为记录程序
    - 引诱恶意软件在蜜罐内更加充分的运行，并记录下其行为。

被动型蜜罐

- 在蜜罐主机上模拟漏洞利用攻击所需的部分服务，通过被动的方式捕获主动传播类型的恶意软件，如蠕虫等
- 优点
    - 捕获漏洞利用样本信息
    - 获悉其传播机制
- 缺点
    - 被动式交互、效率低
    - 大规模主动传播的样本减少

主动型蜜罐-客户端蜜罐、沙箱

- 主动型蜜罐的出现
    - 针对客户端软件的恶意软件更为频繁，如 web浏览器等
    - 恶意软件的传播更具针对性和定向性，降低了传统被动型蜜罐的工作效率
    - 需要仿真和模拟更真实的运行环境、更多的行为交互。
- 实现思路
    a. 通过爬虫等主动获取潜在的恶意软件 (载体)
    b. 将其在蜜罐主机内打开、运行，并模拟进行交互
    c. 根据运行特征，发现和收集漏洞利用信息和恶意软件样本。
- 云计算和虚拟化技术使得设计和部署更为真实、更加先进的客户端蜜罐环境变得更加容易和低成本

7.2 恶意软件载体

- 在彻底清除主机内的恶意软件时，需要定位恶意软件的来源、传播方式和存在形式，即载体。

文件感染型病毒

- 特征
  - 需要修改目标文件或系统实现病毒的存储。一般情况下受感染的文件字节长度会增加（也有特例，部分病毒使用压缩功能将代码放置于文件的冗余处使得文件长度不变），或者硬盘、系统引导数据会被修改。
  - 这些被修改的文件和引导数据，则是病毒的存储载体。
- 

- 检测
  - 完整性校验。如果用户对可执行程序或者引导扇区事先进行过校验和计算和存储，则可以通过完整性校验检测出此类恶意软件。
- 恶意软件也可能藏匿于主板 BIOS存储区

蠕虫、木马类恶意软件

- 植入方式
  - 通常不感染可执行文件，而是和正常软件一样“安装”到系统中
  - 安装过程比较隐蔽，多通过漏洞利用直接进入目标主机
- 恶意软件定位
  - 系统出现异常后，通过专业手段分析系统启动项、新启动的进程、线程，来发现和定位这类恶意软件样本

宏病毒

- 可疑文档或 Word、 Excel、 PowerPoint的模板文件是这类病毒的主要载体。

通过电子邮件传播的恶意软件

- 电子邮件的正文，特别是附件（如 exe、 com、 scr， vb、 bat等文件），通常是病毒的主要载体。

脚本类恶意代码

- 通过支持脚本的软件的漏洞传播恶意代码一种常见的且十分有效的方式。
- 用于触发漏洞的恶意代码通常藏匿于脚本中。

- 例如，通过浏览器中 JavaScript、 VBScript触发浏览器漏洞、或者通过 Flash文件中的ActionScript触发漏洞，然后编码在脚本中的恶意代码。

应用程序重打包（ Android App）

- androidAPP具有容易被逆向分析、修改后可以再次打包为可运行 App的特点
- 因此，以流行 App为载体、向其中插入恶意代码后重打包，这种方式是 Android恶意代码传播的一个重要方式。
- 完整性校验：与原版 App进行对比，即可定位出插入到重打包 app中的恶意代码。

7.3 恶意软件样本分析方法

恶意软件样本分析

- 样本分析的意义
  - 理解其工作机理和行为特征
  - 实现或完善相应的安全监测机制
  - 实现对已有恶意软件和未知恶意软件的防御、监测
- 目的
  - 程序有哪些破坏功能?
  - 程序的破坏功能是如何实现的?
  - 程序有哪些网络活动、其活动特征是什么样的?
  - 程序是如何实现系统驻留和自启动的?
  - 程序是否感染系统或其他程序，或网络中的其他主机?
  - 程序是如何进入系统的?
- 出于取证分析、持续威胁的追踪和溯源，还需思考
  - 程序编写者具备哪些编程习惯和特征?
  - 程序使用什么典型攻击手法来攻击主机?
  - 程序反映出攻击者的技术水平如何?
  - 攻击者可能是什么样的团体或组织，存在哪些控制的行为特征模式?
  - 该程序是否与其他恶意软件存在关联?

常用分析方法

- 在线分析
  - 在线病毒扫描
  - 在线行为分析
- 本地静态分析
  - 加壳检测与脱壳
  - 反汇编/反编
  - 资源分析
- 本地动态分析
  - 行为监控分析
  - 调试跟踪

    - 网络监控分析
    - 运行环境仿真
- 网络交互的动态分析
    - 网络连接选择
    - 网络交互环境仿真
    - 数据包捕获分析

分析方法

- 运行环境仿真
    - 时间模拟
    - 文件资源模拟
    - 对抗反虚拟机
    - 对抗反调试
- 行为监控分析
    - 关注文件、注册表、进程、网络、内存等操作
- 网络交互环境仿真
    - DNS构造
    - IP地址分析和模拟
    - 服务器模拟和数据响应模拟

方法选择的一般思路：粗粒度分析→针对性的细粒度分析

- 粗粒度分析
    - 了解恶意软件的恶意行为
    - 在线行为分析、快照比对、行为监控、粗粒度的资源分析等方法
- 细粒度分析
    - 有针对性的研究恶意软件的功能实现、入侵细节、潜在威胁等
    - 静态反汇编、动态调试分析、API监控、环境仿真等方法

7.4 恶意软件样本分析工具

常用分析工具

- 虚拟机环境
    - 提供隔离的样本运行环境 使用虚拟机软件将物理计算机硬盘和内存的一部分以及其它相关硬件资源共享出来，从而虚拟出若干计算机，每台计算机可单独运行操作系统。
    - 虚拟机中的系统与物理主机系统相互隔离，虚拟机之间相互隔离。
- 系统监控
    - 监控样本运行期间的行为和对系统的改变
- 进程监控
- 注册表监控
- 文件行为监控

- 网络行为监控
- HIP类软件
  - 通过 HIPS软件的提示或日志，获取软件的安全相关行为
- 文件类型检测
  - 在静态分析中，通常需要先了解程序的编写语言、编译器信息，或者加壳信息，然后进行分析
  - PEiD：有效识别加壳类型、编译器等
  - Linu下File命令：识别文件类型
- PE文件格式分析
  - 了解 PE文件的格式特征、节信息、代码入口点等
- 静态反编译
  - 将经过编译器编译后的机器码或字节码等还原成便于理解的语言形式的过程为反编译。
- 动态调试
  - 对目标程序进行运行时的单步跟踪，以进一步了解目标代码运行的细节，洞悉目标程序的运行机理
- 网络通信数据分析
  - 涉及网络通信的恶意软件样本，也可以分析其网络通信数据、获取其特征等信息
- 文件原始数据 (RawData)分析
  - 恶意软件样本的文件类型多种多样，有时候需要用特定类型格式解析文件、并用16进制进行查看特定字段数值。
  - 例如在文件格式漏洞样本分析中，需查看 flash文件（ SWF）、 pdf文件、 mp3文件的特定字段

其他

- 使用 PEiD或通过其它方式发现分析对象被加壳，则在静态分析前脱壳或动态调试时尝试脱壳（或跳过对壳代码执行的追踪）
- 根据 PEiD等工具识别出被分析对象的编译工具，可针对性的使用相应逆向分析工具，事半功倍