第二十一章 恶意代码与应用攻击

21.1 恶意代码

恶意代码对象包括广泛的代码形式的计算机安全威胁，威胁利用各种网络、操作系统、软件和物理安全漏洞对计算机系统散播恶意载荷

21.1.1 恶意代码的来源

恶意代码来自相当有经验的软件开发人员以及一些脚本小子，目前大量病毒被反病毒机构证明准许任何具有极少技术知识的人制造病毒并在互联网上传播

21.1.1 病毒

病毒具有两个主要功能，传播和破坏，这都能产生任何针对系统或数据机密性、完整性和可用性的负面影响

1. 病毒传播技术
    - 病毒必须包括能够在系统之间进行传播的技术，4中常见的传播技术：
        - 主引导记录病毒：系统读取受到感染的MBR,病毒会引导系统读取并执行在另一个地方的代码，从而将病毒加载到内存中，并可能触发病毒有效载荷的传播
        - 文件程序感染病毒：感染不同类型的可执行文件，并且在操作系统师徒执行这些文件时触发，文件程序感染病毒的变种是同班病毒，利用与合法操作系统文件类似但又稍有不同的文件名来躲避检查
        - 宏病毒：应用程序为了协助重复任务的自动执行而实现某些功能
        - 服务器注入病毒：通过成功破坏受信进程，能够绕过主机上运行的任何防病毒软件的检测
2. 容易受到病毒攻击的平台
    - 大多数计算机病毒被设计成破坏windos上运运行的活动
3. 反病毒机制
    - 使用特征型反病毒时，软件包的有效性只依赖于基础性的病毒定义文件的有效性
    - 许多反病毒人廉使用基于启发式的机制来检测潜在的恶意软件感染
    - 大多数现代反病毒软件产品能够检测、删除和清除系统上的大量不同类型的恶意代码
    - 其他的软件包也提供了辅助反病毒功能
4. 病毒技术：
    - 复合病毒：使用多种传播技术师徒渗透只防御其中一种方法的系统

    - 隐形病毒：通过对操作系统的实际篡改来欺骗反病毒软件包认为所有的事情都工作正常，从而将自己隐藏起来
    - 多态病毒：在系统间传输时，多态病毒实际上会修改自己的代码，这种病毒的传播和破坏技术保持相同，但每次感染新的系统时病毒的特征略有不同
    - 加密病毒：使用密码输来躲避检测，通过修改在磁盘上的存储方式来躲避检测
5. 骗局
    - 病毒骗局如收到携带病毒的电子邮件

21.1.3 逻辑炸弹

- 逻辑炸弹是感染系统并且在达到一个或多个满足的逻辑条件前保持休眠状态的恶意代码

21.1.4 特逻辑木马

- 特洛伊木马是一种软件程序，表面上友善，实质上承载恶意的有效载荷，具有对网络和系统的潜在破坏能力，另一变种-勒索软件，勒索软件感染目标计算机，然后使用加密技术来对加密存储在系统上的文档、电子表格和其他文件

21.1.5 蠕虫

蠕虫包含的破坏潜力与其他恶意代码对象相同，还具有额外的手段，不需要任何人干预就可以传播自己

1. Code Red蠕虫：系统管理员必须确保他们为连接Internet的系统使用了软件供应商所发布的适当的安全补丁
2. 震网病毒：高度复杂的蠕虫使用各种高级技术来传播，似乎从中东开始传播

21.1.6 间谍软件与广告软件

- 间谍软件会监控你的动作，并向暗中监视你活动的远程系统传送重要的细节
- 广告软件使用多种技术在被感染的计算机上现时广告

21.1.7 对策

- 针对恶意代码的主要防护手段就是使用反病毒过滤软件，至少在三个关键区域考虑反病毒软件过滤
    - 客户端系统
    - 服务器系统
    - 内容过滤系统： 对入站和出站电子邮件以及Web流量进行内容过滤，是非常明智的
- 零日漏洞的存在，使得必须在组织中拥有强大的补丁管理城西，确保应用及时更新

21.2 密码攻击

- 攻击者获得对系统非法访问的最简单技术之一就是获悉已授权系统用户的用户名和密码

21.2.1 密码猜测攻击

- 攻击者只师徒猜测用户的密码

21.2.2 字典攻击

- 密码攻击者使用自动化工具运行自动的字典攻击

21.2.3 社会工程学攻击

- 社会工程学是攻击者用于获得系统访问权限的最有效工具之一

21.2.4 对策

- 安全人员应该经常提醒用户选择安全密码进行保密的重要性
- 为用户提供安全密码所需的知识，告诉他们攻击者在猜测密码时所使用的技术，并且为用户提供一些有关如何建立强密码的建议

21.3 应用程序攻击

21.3.1 缓冲区溢出

- 缓冲区溢出漏洞存在于开发人员不正确的验证用户的输入，输入太大，影响存储在计算机内存中的其他数据

21.3.2 检验时间到使用时间

- 时间型漏洞，当程序检查访问许可权限的时间大大遭遇资源请求的时间时，就会出现这种问题

21.3.3 后门

- 没有记录到文档中的命令序列，允许软件开发人员绕过正常的访问限制

21.3.4 权限提升和rookit

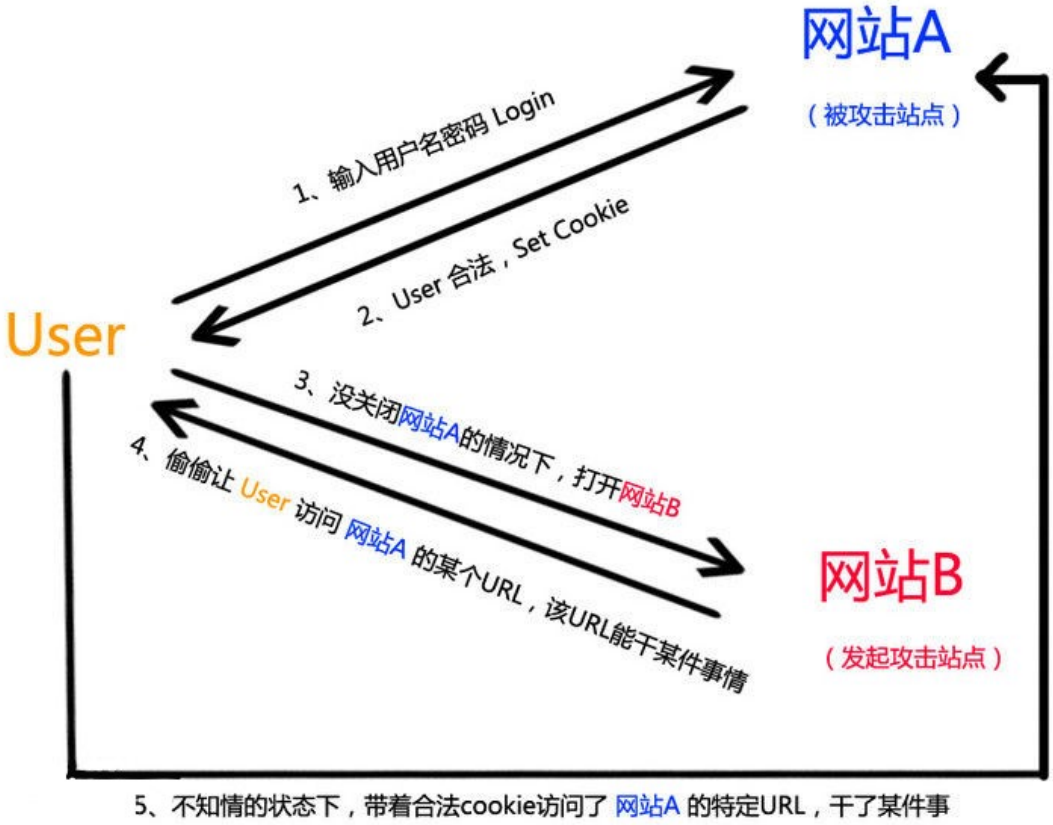
- 攻击者权限提升攻击的最常见方法之一就是通过rookit
- 系统管理员必须关注针对其环境所使用操作系统而发布的最新补丁，并且始终坚持应用这些修正措施

21.4 Web应用的安全性

21.4.1 跨站脚本（XSS）攻击

- 当web应用程序包含反射式输入类型时，就容易出现跨站脚本攻击
- 防御跨站攻击的方法：确定许可的输入类型，然后通过验证实际输入来确保其与制定模式匹配
    - 非持久型XSS（反射型），攻击者欺骗受害者处理一个用流氓脚本编写的URL，从而偷取受害者敏感信息cookie、会话ID等。攻击原理是利用动态网站上缺少适当的输入或者输出确认。
    - 持久型XSS（存储型、第二顺序），通常针对的是那些让用户输入存储在数据库或其他任何地方（论坛、留言板、意见簿等）的数据网站。攻击者张贴一些包含恶意JavaScript的文本，在其他用户浏览这些帖子时，他们的浏览器会呈现这个页面并执行攻击者的JavaScript。
    - 文件对象模型（Document Objec Model，DOM，也叫本地跨站脚本）XSS，DOM是标准结构布局，代表着浏览器中的HTML和XML。在这样的攻击中，像表单字段和cookie这样的文档组件可通过JavaScript被引用。攻击者利用DOM环境来修改最初的客户端JavaScript。这使受害者的浏览器执行由此而导致的JavaScript代码。
- **CSRF：**攻击通过在授权用户访问的页面中包含链接或者脚本的方式工作。
    - 例如：一个网站用户Bob可能正在浏览聊天论坛，而同时另一个用户Alice也在此论坛中，并且后者刚刚发布了一个具有Bob银行链接的图片消息。设想一下，Alice编写了一个在Bob的银行站点上进行取款的form提交的链接，并将此链接作为图片src。如果Bob的银行在cookie中保存他的授权信息，并且此cookie没有过期，那么当Bob的浏览器尝试装载图片时将提交这个取款form和他的

cookie，这样在没经Bob同意的情况下便授权了这次事务。



21.4.2 SQL注入攻击

SQL注入攻击使用了Web应用程序不期望的输入

1. 动态Web应用程序：如果web应用程序存在缺陷，可能导致某写使用SQL主攻击的用户能以不期望和未授权的方式篡改数据库
2. SQL注入攻击：违反隔离性，直接完成攻击内在数据库的SQL事务
3. 防御SQL注入
    - 执行输入验证：输入验证操作能够限制用户在表单中输入的数据类型
    - 限制用户特权：web服务器使用的数据库账号应当具有尽可能最小特权集
    - 使用存储过程：利用数据库存储过程来限制应用程序执行任意代码的能力

21.5 侦查攻击

侦查可以让攻击者找到弱点，利用他们的攻击代码直接攻击

21.5.1

IP探测：针对目标网络而实施的一种网络侦察类型

21.5.2

端口扫描

21.5.3

漏洞扫描：利用已知漏洞数据库，通过探测目标来定位安全缺陷

- 只有操作系统升级到最新的安全补丁级别，才有可能几乎完全修复漏洞扫描程序中的所有漏洞

21.5.4 垃圾搜寻

垃圾搜寻的防护方法：

- 使攻击者的行动变得困难
- 垃圾保存在一个安全的地方

21.6 伪装攻击

- 为了获得对没有访问资格的资源的访问权限，最简单的方法就是假冒具有适当系访问许可权限的人

21.6.1 IP欺骗

- 怀有恶意的人重新配置他们的系统，使其具有可信系统的IP地址，然后视图访问其他外部资源的权限
- 防止IP欺骗的措施
  - 具有内部源IP的地址包不能从外部进入网络
  - 具有外部源IP地址的包么不能从内部离开网络
  - 具有私有IP地址的包不能从任何一个方向通过路由器

21.6.2 会话劫持

- 会话劫持攻击指的是怀有恶意的人中途拦截已授权用户与资源之间通信数据的一部分