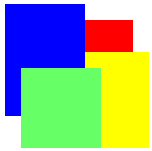


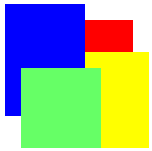
第5章 网络攻击与防御

董建润

南京邮电大学

关键知识点

- 危害网络系统及其网络系统连接的信息系统的安全性的行为统称为“网络攻击”。
- 网络攻击可以分成两大类：基于系统渗透的攻击和基于拒绝服务的攻击。
- 拒绝服务(DOS)攻击并没有泄露网络系统的信息，也没有在网络系统中放置恶意代码。但是，拒绝服务攻击已成为一种最为严重的网络攻击形式。
- 通常基于访问控制系统的审核记录进行攻击检测，网络攻击检测分成特征检测方法和异常检测方法。



主要内容

- 网络攻击定义
- 网络攻击分类
- 网络攻击检测系统
- 网络攻击检测方法
- 网络恶意代码与防御
- 僵尸网与防御

网络攻击定义

- 危害网络系统及其网络系统中连接的信息系统的安全性的行为统称为“网络攻击”。
- 传统网络攻击对应英文的Network Intrusion，表示对网络系统以及连接在网络系统中的信息系统、计算机系统等的一种侵入行为。现在对应的英文术语是Network Attack，表示一类网络破坏行为。
- “网络侵入”实际上是破坏网络系统的保密性和完整性，“拒绝服务” 是破坏网络系统的可用性。
- 网络侵入和拒绝服务可以统称为“网络攻击”。

网络攻击历史

- 计算机和网络攻击一览表

年代	对计算机和网络系统的攻击
1980至1985	口令猜测、自我复制恶意代码、口令破解
1985至1990	探测已知缺陷、关闭日志、网络蠕虫、恶意侵入、后门攻击
1990至1995	虚假分组、劫持会话、自动探测扫描、报文嗅探、GUI入侵工具
1995至2000	大规模的拒绝服务攻击、对浏览器的恶意代码攻击、先进扫描技术、基于Windows的远地可控特洛伊代码、电子邮件传播恶意代码、大规模传播特洛伊木马代码、分布攻击工具
2000至今	分布式拒绝服务攻击、大量变种网络蠕虫、基于电子邮件传播恶意代码、基于网页链接传播恶意代码、通过恶意代码获取用户身份信息和账户信息收益、防取证技术、复杂的攻击控制工具

近几年网络攻击的变化趋势

- 以下是2014-2015网络攻击变化趋势

1	Malware	⇑
2	Web Based Attack	⇑
3	Web Application Attack	⇑
4	Botnets	⇓
5	Denial of Service	⇑
6	Physical Damage	⇔
7	Insider Threat	⇑
8	Phishing	⇔
9	Spam	⇓
10	Exploits Kits	⇑
11	Data Focused Attack	⇔
12	Identity Theft	⇔
13	Information Leakage	⇑
14	Ransomeware	⇑
15	Cyber Espionage	⇑

Notation: ⇑ Increasing, ⇓ Decreasing, ⇔ Same

系统渗透类攻击

- 按照Stephen D. Crocker的观点，目前网络攻击可以分成两大类：基于系统渗透的攻击和基于拒绝服务的攻击。
- 基于系统渗透的攻击是攻击者发现被攻击网络系统的(技术或配置)漏洞之后，对网络系统进行的非授权的访问和其他操作。
- 网络系统的漏洞包括技术漏洞（例如缓存区溢出这类软件漏洞）和配置漏洞（例如易于猜测的口令、没有修改默认口令、没有关闭远程登录等）。

系统渗透类攻击的防范

- 防范渗透型网络攻击方法：
  - 完整设计坚固的网络体系结构—可以做到；
  - 正确实现这种网络体系结构—可以做到；
  - 严密配置这些网络系统的安全控制策略—可能做到；
  - 严格训练网络用户的安全意识和防范能力—难以做到。
- 迄今为止，防范渗透型网络攻击的效果并不理想。原因在于：无意识地使用具有逻辑漏洞的系统，系统安全配置有误差。
  - 解决方案：人工智能的引人，阿兰·图灵的思路：由机器破解机器的编码→由机器防御机器的攻击！

拒绝服务类攻击

- 基于拒绝服务的网络攻击原理：攻击者利用某些手段(植入恶意软件、启动恶意软件的无效网络访问等)，在网络系统中造成大量虚假而正常的网络访问或者网络服务请求，使得网络设备或者网络服务器无法提供正常的服务。
- 拒绝服务(DOS)攻击没有泄露网络系统的信息，也没有渗透到网络系统或应用系统。但拒绝服务攻击已成为互联网上一种最为严重的网络攻击。
  - 注：分布式拒绝服务还需要渗透较多的联网主机

拒绝服务类攻击的操作方式

- 目前在互联网上具有较大威胁的网络攻击是分布式拒绝服务(DDOS)攻击。DDOS攻击是一种结合渗透攻击和拒绝服务攻击的网络攻击方式，它分成两个阶段：
- 第一个阶段是渗透阶段，渗透到尽可能多的计算机系统，在被渗透的系统中设置恶意代码，使得该计算机处于网络攻击者可控制的状态。
- 第二个阶段是DDOS攻击阶段，攻击者向所有已被渗透的主机发出虚假但正常的网络或应用访问指令。

传统的网络攻击

- D. Denning在1987年罗列了一些典型的网络攻击行为：
  - 试图闯入：尝试着侵入
  - 假冒者或成功闯入：已经侵入
  - 合法用户的渗透：合法用户访问非授权资源
  - 特洛伊木马：植入的恶意代码，主要的攻击
  - 病毒：具有繁殖、传播和破坏作用的代码
  - 拒绝服务：恶意地过度占用网络资源

现代网络攻击

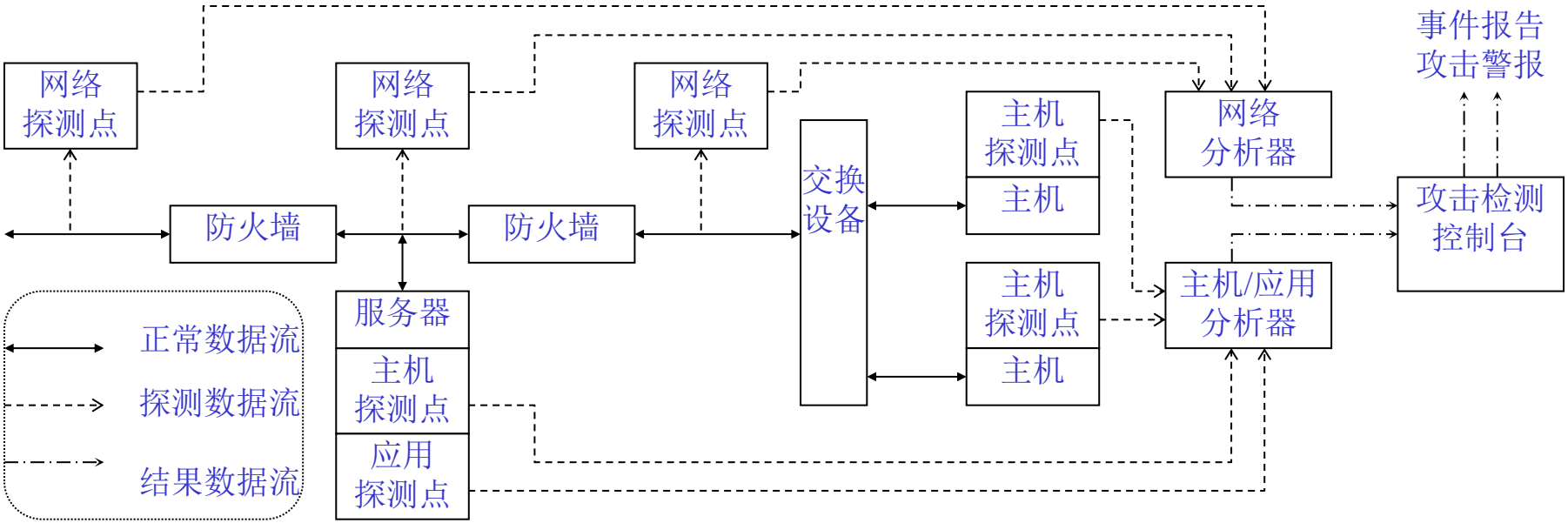
- **中间人(man in the middle, MITM)攻击**：这是一类在两个或多个网络被害人之间的交互信道上截获、修改报文，或者改变或假冒网络被害人进行的网络攻击。
- **僵尸网(botnet)和暗网(darknet)**：僵尸网是指被网络恶意软件感染、并受恶意软件控制的计算机的在线集合。僵尸网造成DDoS攻击、获取敏感信息，发送垃圾邮件等网络安全威胁。暗网是指其网络服务器或网络操作隐形的网络。暗网可能造成隐形恶意软件传播、DDoS攻击等网络安全威胁。
- **隐形恶意软件(stealth malware)**：这是一类可以躲避检测、在很长时间以自然而睡眠方式传播，用于收集敏感信息，潜伏在关键网络位置发起攻击的恶意软件

高级持续威胁攻击

- 高级持续威胁(APT)攻击是一类缓慢、低速移动的、受某个命令和控制中心控制的、可以形成长期持续性网络威胁的攻击，其目的是渗透并长期潜伏在目标网络中，根据控制指令窃取目标数据或破坏甚至摧毁目标系统。
- APT攻击的特征：高级：采用了高级的攻击工具和攻击方法；持续：有组织的持续攻击；威胁：是一类受控的、具有极高威胁的网络攻击。
- APT攻击的五个阶段：目标侦查：收集目标环境的信息；建立据点：成功地侵入目标；侧向移动：在目标网络内移动寻找目标数据或系统；窃取/破坏：根据指令发送敏感数据或破坏目标系统；后窃取/后破坏阶段：根据指令继续窃取数据或破坏系统，直到收到停止行动、清除证据、退出目标网络的指令。

一个网络攻击检测系统结构图

- J. McHugh等人提出了一个典型的网络攻击检测系统的总体结构。



一个典型网络攻击检测系统结构图

网络攻击检测分类

- 按照检测的对象不同，网络攻击检测可以分为基于网络的攻击检测技术和基于主机的攻击检测技术。
- 实际上网络攻击检测技术的本质区别在于检测的方法的不同，正确的网络攻击检测分类方法应该是：基于攻击检测方法的分类。
- 网络攻击检测中，按照攻击检测的方法不同，将网络攻击检测分成特征检测方法和异常检测方法。
- 数据挖掘、机器学习等技术应用于网络攻击检测，可以构成智能网络攻击检测技术(智能的机器防御攻击)。

网络入侵检测涉及的方面

- 输入数据类型：网络入侵检测的关键涉及到输入数据的特征，这些数据特征涉及到相关的数据模型，例如数据属性、数据类型、数据结构等。数据模型越能准确描述行为特征，就越能准确进行异常检测。
- 合适的近似度测量：也是采用合适的统计分析方法，其中包括数据的统计采样和统计分析方法等。涉及到统计分析相关的理论和方法。
- 标记数据：标记相关的数据是正常，还是异常。异常行为是动态变化的，如果缺少正常行为模式的数据支撑（缺少正常行为训练数据），则可能将正常数据标记为异常。

网络入侵检测涉及的方面-1

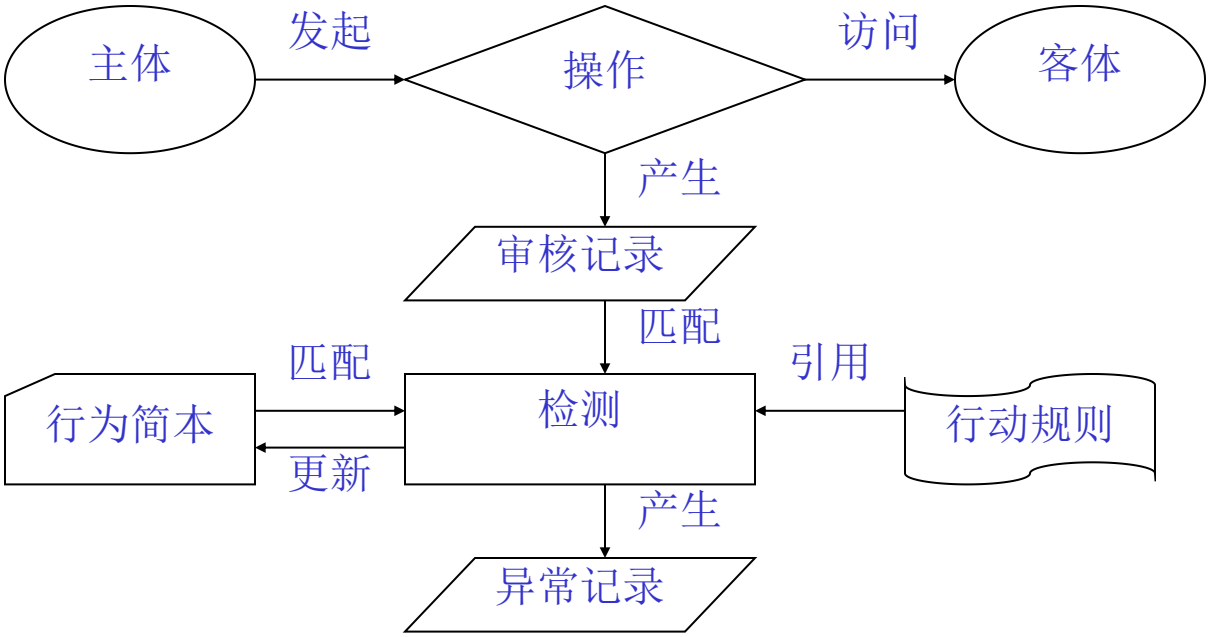
- 基于标记数据的分类方法：异常检测可以在监督、半监督和非监督方式下执行，监督方式下具有正常行为和异常行为的训练数据；半监督方式下仅仅具有正常行为的训练数据；非监督方式下没有任何训练数据，机器学习和深度学习从数据中训练(学习)，但需要评判是否满足实时性要求
- 相关的特征识别，异常检测也采用特征识别方法，但这里主要采用基于特征选择的特征识别方法，特征选择方法包括特征子集生成、评估和验证三个步骤。
- 异常报告的输出：异常报告输出通常有两种方式：其一，综合了与简本或特征集的差距和方差、与相邻简本或特征的影响度等得分；其二，综合了非监督分类或聚类方法产生的群的大小和密度等的标签。

网络攻击的异常检测方法

- 基于异常检测的网络攻击检测模型是从网络访问控制模型中自然引申出的一种攻击检测模型，所以，这种模型已经成为一个典型的网络攻击检测模型。
  - 注：基于特征的攻击检测方法常应用网络病毒的检测，通过识别病毒特征之后，采用的过滤方法。这是一类攻击发生之后的防范方法。
- 异常检测模型包括主体、客体、审核记录、行为简本、异常记录和行动规则。

传统的异常检测方法的结构图

- 传统的异常检测系统结构图(与访问控制关联)



IDES异常检测系统结构图

异常检测方法的举例

- 审核记录用一个6元组表示：<主体，动作，客体，例外情况，资源使用，时间戳>。
- 网络系统中大部分操作都涉及多个客体，审核记录采集系统必须首先将多客体相关的操作分解成单客体相关的操作，然后再生成审核记录。
- 例如Smith执行的将GAME.EXE文件拷贝到<Library>目录的命令：

COPY GAME.EXE TO <Library>GAME.EXE

- 由于Smith没有“写”<Library>目录的权限，所以，该命令执行失败

异常检测方法的举例(续1)

- 针对这项操作需要采用以下3条审核记录表示：

(Smith, execute, <Library>COPY.EXE, 0, CPU=0002, 11058521678)

(Smith, read, <Smith>GAME.EXE, 0, RECORD=0, 11058521679)

(Smith, write, <Library>GAME.EXE, write-viol, RECORD=0, 11058521680)

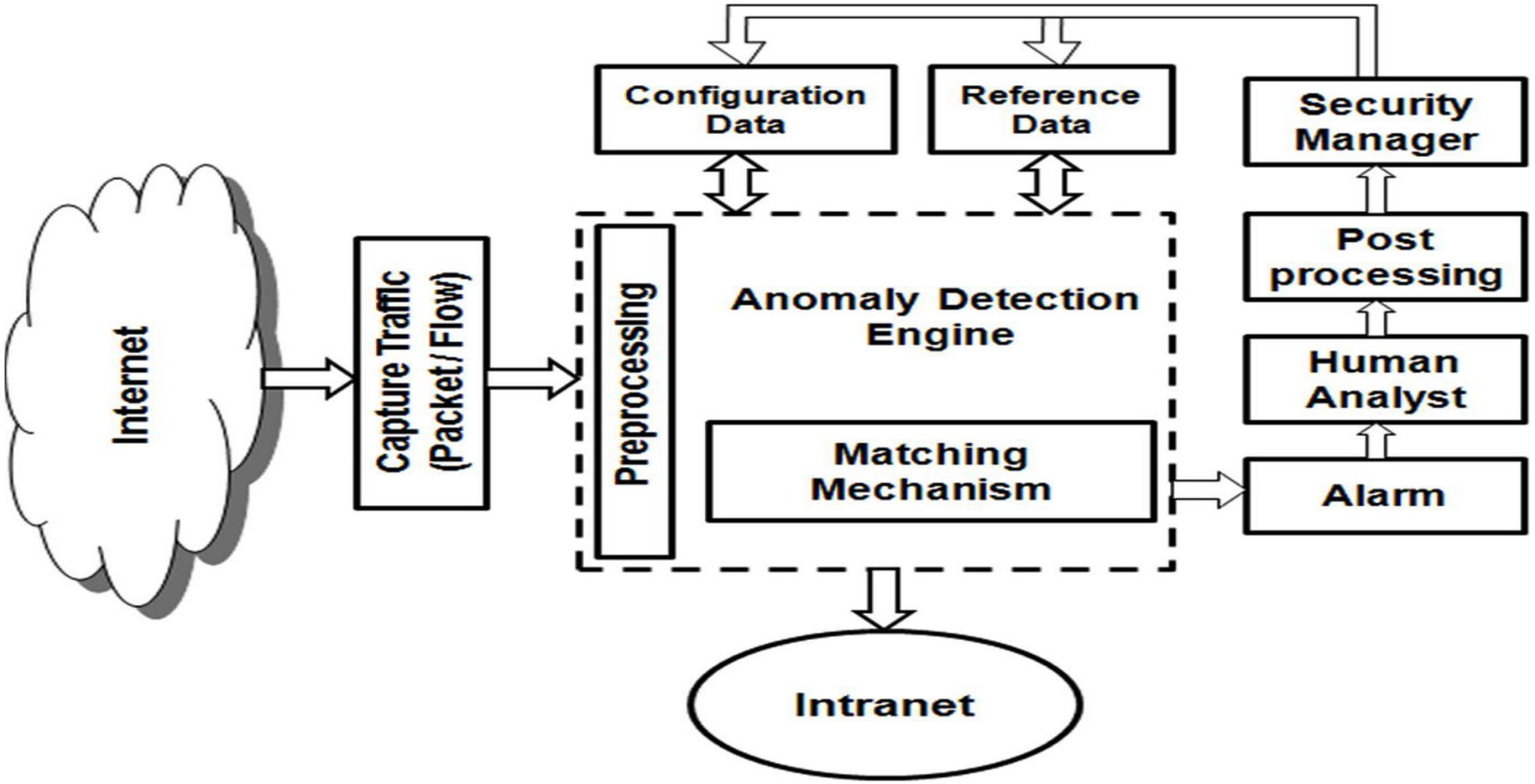
- 通过分析主体对客体的异常访问，发现正在实施或已经实施的网络攻击；通过分析主体对客体访问的例外情况分析，发现潜在的网络攻击—— 数据挖掘

攻击检测与审核记录

- 如果使用基于网络的攻击检测系统，则审核记录由网络探测器产生；如果使用基于主机的攻击检测系统，则审核记录由主机探测器产生。
- 审核记录的产生时间决定了攻击检测的能力。
  - 如果审核记录是在操作开始时就实时生成，则攻击检测系统可以利用审核记录，检测潜在的攻击或者正在进行的攻击；
  - 如果审核记录是在操作完成后产生，则攻击检测系统可以利用审核记录，检侧已经实施的、或已经完成的攻击。

异常检测系统实现的通用结构

- 基于异常检测的网络入侵检测系统实现结构如下

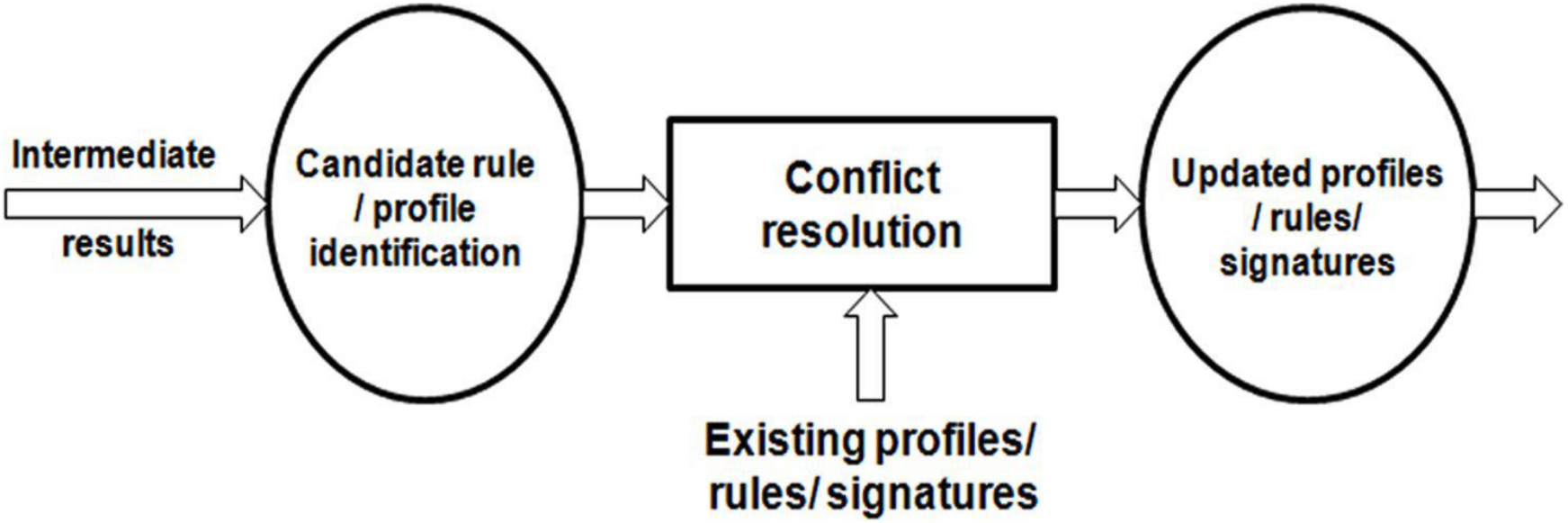


异常检测系统的实现结构-1

- 虽然已经开发了很多实际部署的网络入侵检测系统(NIDS)，但如何开发一个有效的基于异常检测的网络入侵检测系统结构，仍然是一个值得探索的问题。
- 异常检测引擎：这是网络入侵检测系统(NIDS)的心脏，用于在线或离线检测任何入侵的出现。其本身包括预处理和匹配机制两个子部件，预处理子部件主要采用特征检测方法，检测是否存在已知的入侵行为；匹配子部件主要采用异常检测方法，检测可能的异常行为，包括匹配由已知的简本(profile)定义的分类或聚类行为模式，这种匹配要求快速和准确。
- 参考数据：用于存储已知的入侵特征或入侵规则、以及存储正常行为的简本，后续处理部件与安全管理员部件协同操作之后，可以更新相关的简本、特征和规则。

异常检测系统的实现结构-2

- 配置数据：用于存储入侵检测的中间结果，例如部分生成的入侵特征等，中间结果必须与已知的简本、特征和规则集成之后，才能产生一致的、更新的简本、特征和规则。



异常检测系统的实现结构-3

- 报警：用于响应异常检测引擎产生的警报，可以由判断是否是误报，是否需要提交人工分析等。
- 人工分析：负责基于异常检测引擎提供的报警信息进行分析和解释，并且决定是否需要采取必要的行动，同时决定是否需要进行报警信息的后续处理，用于更新参考数据。
- 后续处理：后续处理产生的报警信息，用于诊断与实际攻击的吻合程度，如果与实际攻击不符，则需要更新相关参考数据。
- 捕获流量：在分组层面和分组流层面捕获相关的通信流量，例如采用网络嗅探工具
- 安全管理员：识别新的网络入侵，更新相关简本、特征和规则

异常检测算法的相关概念

- 异常检测模型的核心是基于行为简本的统计检测算法。
- 行为简本刻画了主体针对客体的正常行为特征，这些行为特征是采用已经观察到的统计尺度和统计模型进行描述的。
- 某个统计尺度是指表示在一个时间段内统计度量值的某个随机变量x，这种时间段可以指一个固定的时间间隔，或者两个审核相关事件相继发生的时间间隔（例如登录和注销、程序启动和程序结束、文件打开和文件关闭等时间间隔）。

攻击检测算法的基本原理*

- 异常检测基本方法：从审核记录中提取某个或者某些随机变量新观察到的采样值，利用统计模型对照行为简本进行分析，就可以判断审核记录对应的操作是否属于异常行为，进而可以判断是否是可疑的网络攻击。
- 统计模型对随机变量x没有事先假定任何分布特性，所有关于随机变量x的统计特性都是通过观察获得的。所以，行为简本需要有一个生成和更新的过程。

检测模型的统计尺度

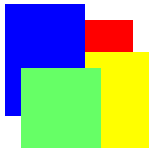
- IDES检测模型定义了3种类型的统计尺度，包括事件计数器，间隔计时器和资源度量器。
- 事件计数器，表示在一段时间内满足某种特征的审核记录出现次数的随机变量。例如在一个小时内登录次数、在一次登录中执行某个命令次数的随机变量等。
- 间隔计时器，表示两个相关事件之间时间长度的随机变量，也就是两个相关审核记录的时间戳之差。例如对一个帐户连续登录的时间长度。
- 资源度量器，表示在一段时间内，审核记录中特定的“资源使用”消耗资源数值的随机变量。例如某个程序在单次执行过程中消耗的CPU时间总数。

检测统计模型

- 定义：对于一个随机变量x和n个观察$x_1, x_2, \ldots, x_n$，x统计模型的目标是确定新的观察x_{n+1}相对于已有的观察是否异常。
- 操作模型，该模型基于这样一种假设：通过将随机变量x的新观察值与某个门限值比较，可以确定新的观察是否异常。
  - 这里用于比较的门限值是根据已有的正常行为的观察，统计产生的。
  - 操作模型可以应用于检测尝试性的渗透攻击和某些拒绝服务攻击
  - 例如在较短的时间段内，登录口令失败的次数超过10，则是一种异常的登录行为。

检测统计模型(续1)

- 平均和标准方差模型，该模型基于这样一种假设：如果随机变量*x*的新观察值处于对已有观察值统计生成的置信区间之外，则新的观察是异常的。
- 该模型可以应用于在一段时间内或者在两个相关事件之间累积的事件计数器、间隔计时器和资源度量器。
- 这种模型不需要有关正常行为的知识，可以完全根据对已有观察的统计设置和调整置信区间。所以，这种模型比操作模型更加准确、适应性更强。



检测统计模型(续2)

- 多元模型，这是一个扩展的平均和标准方差模型，它基于两个或者多个统计尺度进行判断，适用于从多个统计尺度可以更加准确的检测异常行为的应用场合。
  - 例如从一个程序的CPU使用时间和输入/输出单元的使用频率两个尺度可以更准确地判断一个程序是否正常。
- 马尔可夫过程模型，该模型只能应用于时间计数器尺度。在该模型中，将不同类型的事件看作不同的状态变量，采用状态变迁矩阵描述状态之间变迁的频率。

检测统计模型(续3)

- 马尔可夫模型的应用：如果根据已有的状态和状态变迁矩阵，可以判断一个新观察的状态出现概率极低，则这个新的观察就是一个异常事件。这种模型可以应用于对一组规范命令序列执行过程的异常判断。
- 时间序列模型，该模型同时考虑了观察$x_1, \ldots, x_n$到达的顺序、时间间隔及其这些观察点的取值。针对这个模型，如果新的观察在某个时间出现的概率极低，则这个新的观察是异常的。

网络蠕虫*

- “网络蠕虫”是一类自我复制和自我传播的、无需人工介入传播的恶意代码。蠕虫的表现形式与网络上的移动代码有些类似，
  - 移动代码是一类网络上自我传播和自我执行的程序或者代码。移动代码是为了方便网络管理和配置而设计和部署的一类可以在网络上自动传播的合法代码，这类移动代码一般称为“移动智能体”或“移动代理”。
- 网络蠕虫依然是互联网上的头号安全威胁，它曾经在国际上造成了上百亿美元的经济损失，防范和控制网络蠕虫已经成为网络安全研究的头等大事—现在的防火墙可以防范网络蠕虫。
  - 网络蠕虫一般通过被攻击的网站或虚假网站、电子邮件、共享文件而传播。

莫里斯蠕虫

- 第一个在因特网上造成重大影响的网络蠕虫是莫里斯(Morris)蠕虫，该网络蠕虫爆发于1988年11月2日晚。莫里斯蠕虫结束了因特网乌托邦式信任时代，开始因特网防火墙技术、因特网病毒防范新时代。
- 恶意代码是一种特殊的程序，用于执行非授权的，通常是有害的或不受欢迎的动作。“病毒”就是一种恶意代码，是一种具有自我繁殖能力的恶意代码。
- “蠕虫”也是一种恶意代码，它不仅具有自我繁殖能力，而且可以在网络上自主传播的一种恶意代码。

网络蠕虫相关概念

- “特洛伊木马”是一种看似有用的计算机程序，但是，它隐藏了潜在有害的功能。“特洛伊木马”也是一种恶意代码，它与“病毒”和“蠕虫”的区别是：无自我复制的能力，即无自我繁殖的能力。
- “远地访问特洛伊”是一种“特洛伊木马”，一旦被执行，它允许非授权的用户远地访问控制被它攻破的系统。
- “后门”，有时也称为“陷门”，是程序设计者在程序中设置的一项功能，它允许设计者完全的或者部分的非授权访问执行该程序的系统。

网络蠕虫分类

- D. Kienzle和M. Elder通过对蠕虫特征的分析，将蠕虫分成以下3种类型：电子邮件蠕虫，Windows文件共享蠕虫和传统蠕虫。
- 电子邮件蠕虫是利用电子邮件机制在网络上传播的一类蠕虫；现在演变成为短信蠕虫
- 传统蠕虫是采用莫里斯蠕虫设计技术，利用TCP/IP协议的连接在网络上传播的一类蠕虫。
- 内核蠕虫，隐藏在系统内核中、不易被检测和清除的一类恶意代码。

电子邮件蠕虫

- 在2003年前后爆发的网络恶意代码起始于电子邮件蠕虫的发明。从1998年末首次出现电子邮件概念性蠕虫开始，目前已经存在几百种电子邮件蠕虫。
- 另外还有一些与电子邮件蠕虫传播方式相似的文件共享应用类蠕虫，例如基于客户端到客户端协议的应用，美国在线(AOL)的即时消息(AIM: AOL Instant Messenger)，因特网中继交谈(IRC: Internet Relay Chat)以及多种对等文件共享系统。
- 这些利用其他应用协议进行网络攻击的蠕虫与电子邮件蠕虫的变种十分接近，主要是欺骗用户执行不可信的文件，感染、传播和启动蠕虫。

电子邮件蠕虫(续1)

- 电子邮件蠕虫是一种程序，它被执行后，会利用本地系统上用户的电子邮件功能向其他主机传播自身程序。
- 早在1987年，一种称为“圣诞树”的特洛伊木马就利用电子邮件在网络上传播恶意代码。这种电子邮件蠕虫或者电子邮件病毒通常被戏称为“邮递员”，从1998年开始，这种“邮递员”在恶意代码制造者中十分流行，因为编写这种“邮递员”比较简单，网络上可以找到辅助工具和编写指南。

电子邮件蠕虫(续2)

- 初期电子邮件蠕虫只是简单使用被攻破的计算机系统中本地邮件程序和/或者邮件应用编程接口(API)，向一个或者多个地址发送自己的拷贝。
- 随后电子邮件蠕虫利用它们自己的简单邮件传送协议(SMTP)引擎传播蠕虫(例如Magistr蠕虫)，而不再依赖于被攻破系统中的邮件功能。
- 后来电子邮件蠕虫充分利用自身携带的SMTP引擎与互联网上流行的开放邮件中继系统交互，传递蠕虫(例如Sircam蠕虫)，并且利用自身SMTP引擎和开放邮件中继系统假冒邮件报文头(例如Klez蠕虫)。

传统蠕虫

- 传统蠕虫很大程度上是按照1988年的莫里斯蠕虫模型设计的一类网络蠕虫，这类蠕虫主要利用TCP/IP协议建立的直接连接，在因特网上发起攻击，查找可以连接的网络主机中操作系统和应用软件的漏洞或者猜测可侵入系统的用户帐户口令。
- 传统蠕虫在传播和执行方面具有较高的独立性，除了电子邮件和文件共享之外，还具有其他的传播方式。
- 编制传统蠕虫比较复杂，目前在因特网上传播的传统网络蠕虫也比较少。

传统蠕虫(续)

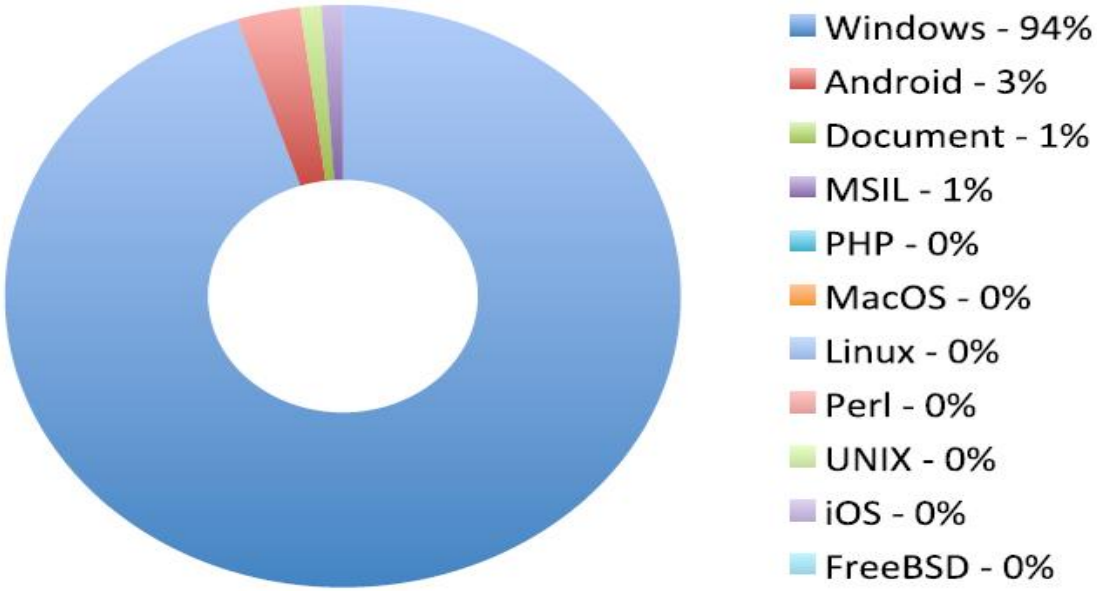
- 根据D. Kienzle和M. Elder的分析，从1988年到2003年，具有特色的传统网络蠕虫仅仅包括Morris(莫里斯)、Nimda(尼姆达)、Code Red(红色代码)和Slammer(大满贯)这4种蠕虫。
- 莫里斯蠕虫是第一个在因特网上泛滥成灾的网络蠕虫，该蠕虫的设计和实现引入了许多关键的网络攻击技巧，例如口令猜测攻击、利用漏洞攻击、自我隐藏攻击、以及多方位攻击等。这些网络攻击技巧今天还被许多网络蠕虫使用。莫里斯蠕虫是迄今为止最为复杂的网络蠕虫。

互联网蠕虫特征

- 互联网蠕虫的生命期分成四个阶段：目标寻找，蠕虫传送，蠕虫激活，蠕虫感染，其中前两个阶段涉及到网络操作；后两个阶段涉及到主机操作。主机内的蠕虫与病毒类似，属于计算机防病毒技术的范畴，这里不做详解。
- 根据蠕虫在网络环境下的操作，可将蠕虫特征归纳为四类：
  - 目标寻找模式，寻找漏洞网络、漏洞主机、漏洞应用的模式，包括：盲扫描、攻击名单、网络拓扑、被动侦听、网页搜索。
  - 传播模式，网络环境下传播蠕虫代码的模式，包括：自携带、第二通道、嵌入、僵尸网。
  - 传输模式，传输蠕虫所采用的网络协议的模式：TCP、UDP。
  - 代码格式，网络环境下传递蠕虫代码格式：单形态、多形态、元形态。

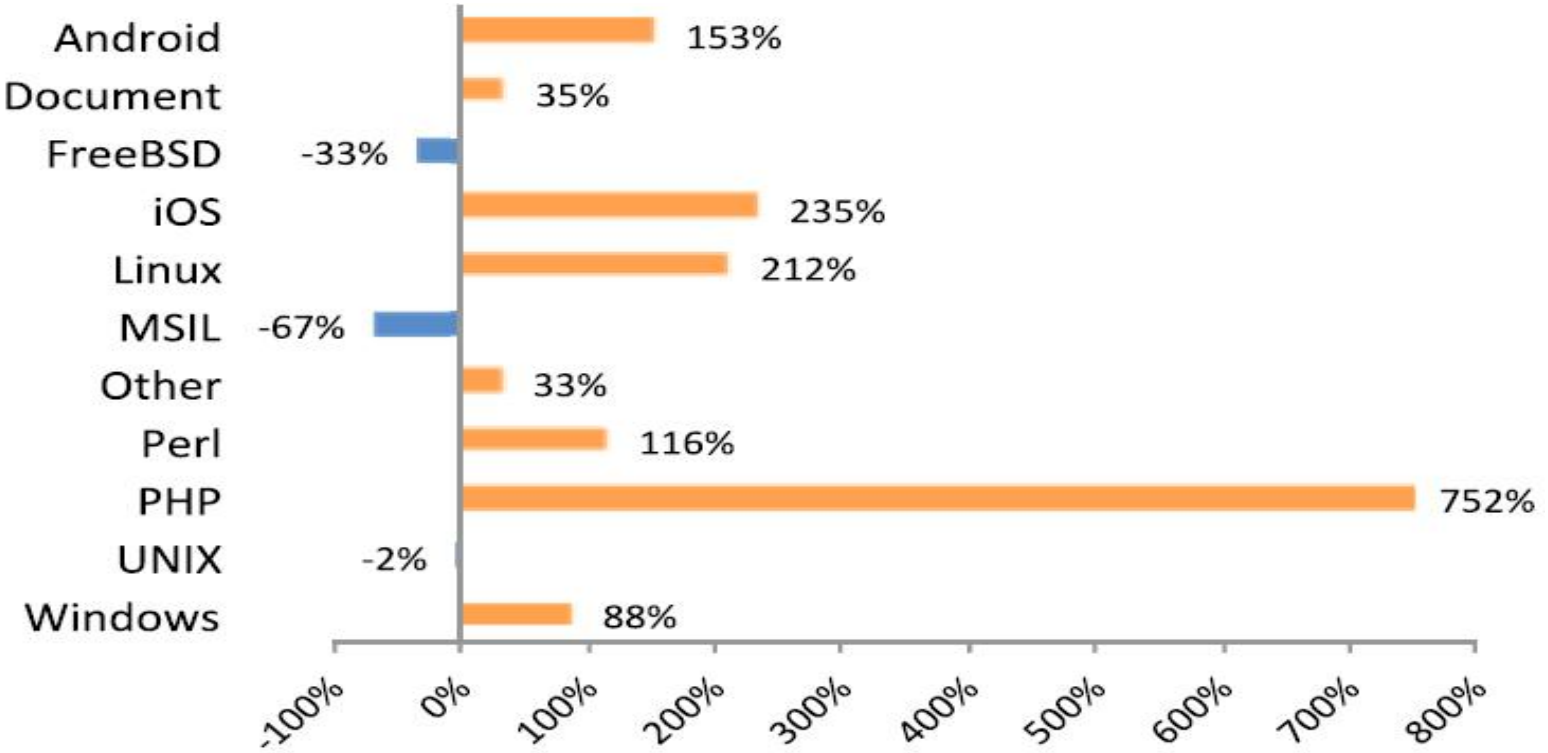
隐形恶意软件

- 这是一类可以躲避检测、在很长时间可以自然而睡眠方式传播，用于收集敏感信息，潜伏在关键网络位置发起攻击的恶意软件。恶意软件在不同平台类型中的占比如下：



(a) Proportion of Malware by Platform Type

不同平台的恶意软件发展趋势



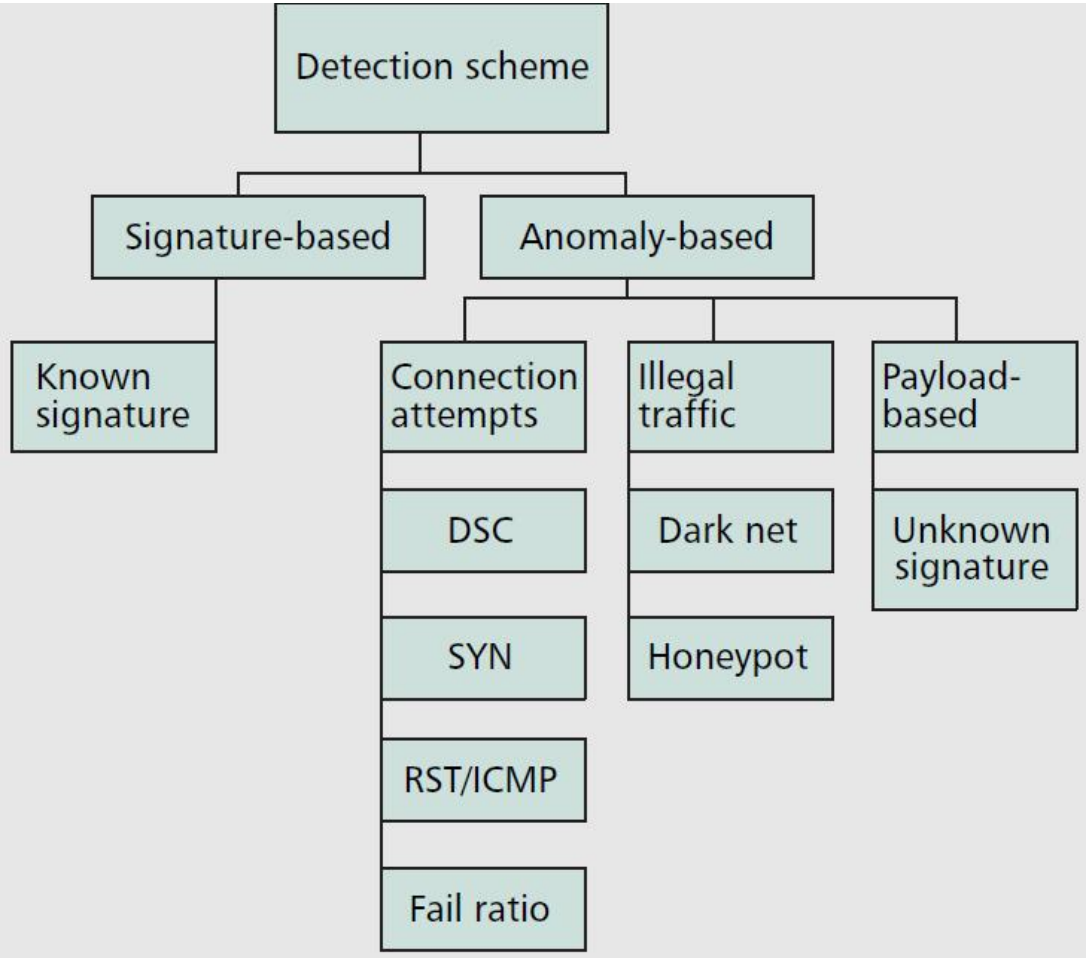
(b) Growth Rate in Malware Proportion by Platform Type

恶意软件的研究重点

- 对于Windows的1.35亿恶意软件样本的基数，每年88%的增长率，也就是每年新增1.18亿个新的Windows恶意软件；
- 安卓恶意软件有450万样本，而在2015年发现的苹果的iOS的恶意软件样本总共不到7万个。
- 重点在于Windows恶意软件的研究，安卓系统中的恶意软件隐身技术与Windows中的相似。

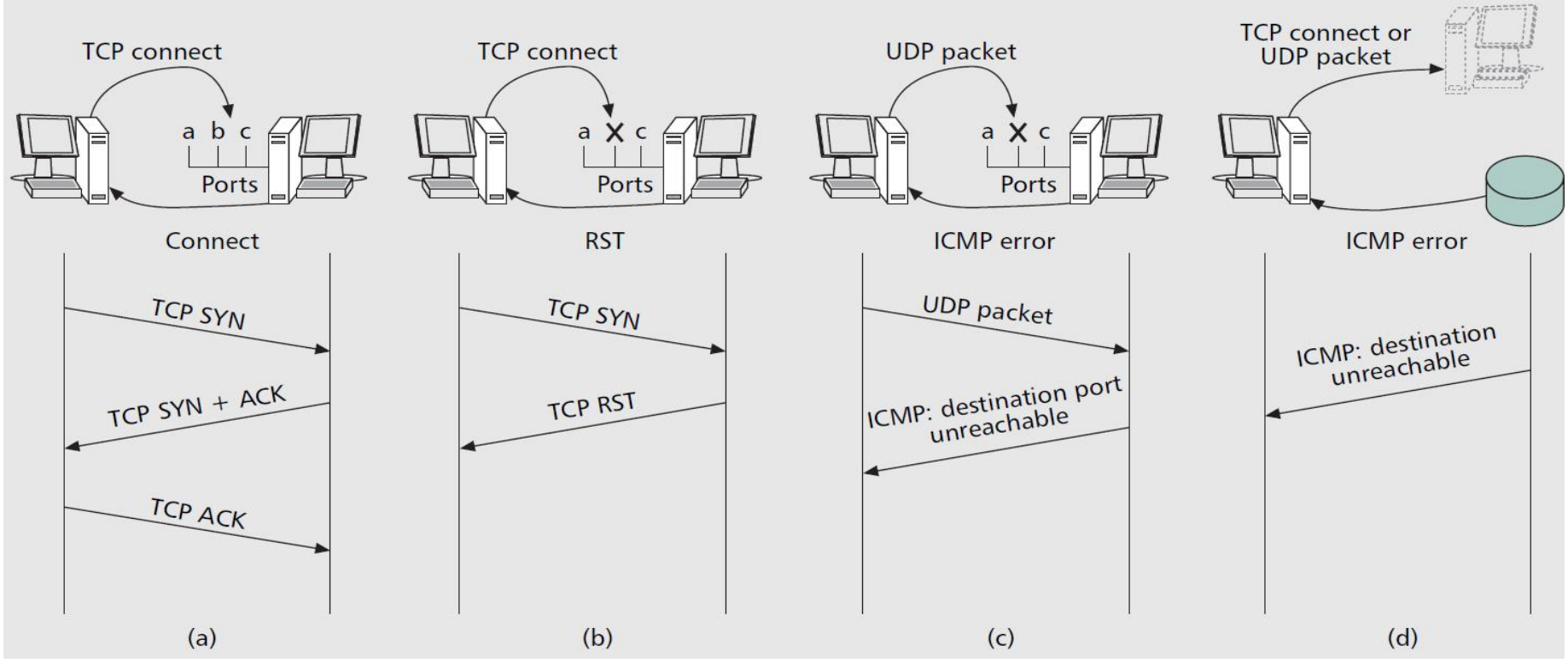
互联网蠕虫检测方法

- 互联网蠕虫的检测方法分成两类：
  - 特征检测方法
  - 异常检测方法，包括连接异常检测方法；流量异常检测方法；分组内容异常检测方法



连接异常的蠕虫检测

- 连接尝试过程中的异常现象：a) 成功TCP连接次数过多；b) TCP目的端口关闭；c) UDP目的端口关闭；d) 目的IP地址不存在。可以用于检测蠕虫的扫描。

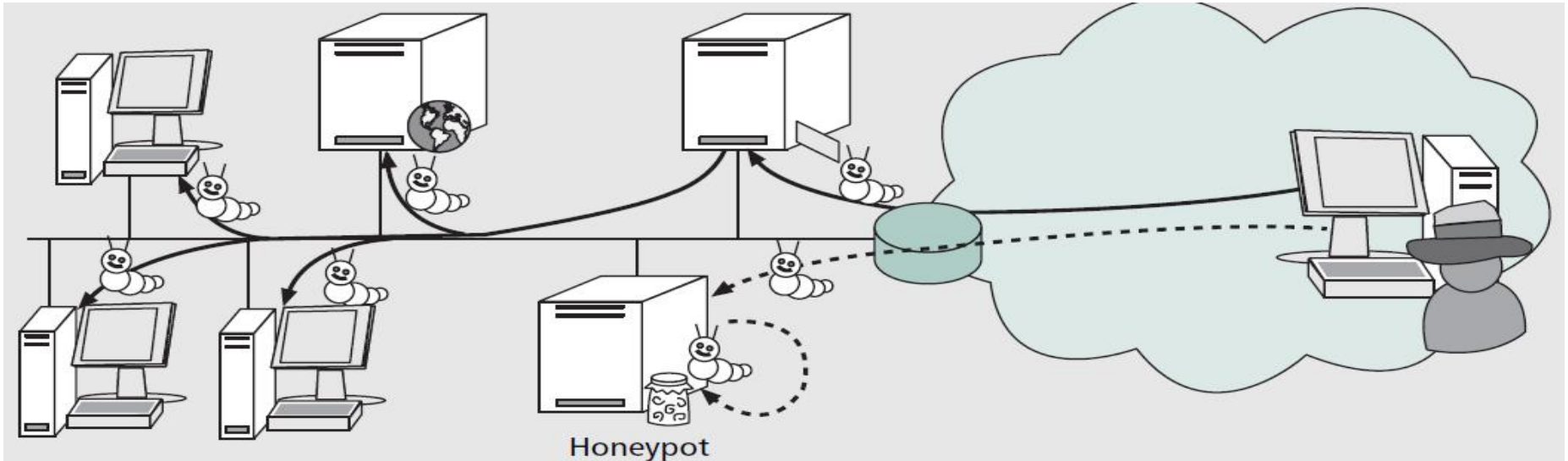


连接异常的蠕虫检测说明

- 从某个主机发出的TCP的连接请求报文(SYN)数在某个时间段内超过某个阈值，则可以认为这个主机正在扫描（a）。
- 按照TCP/IP协议，如果目的主机的IP地址不存在，则返回ICMP主机不可达的分组（d）。
- 如果TCP连接的目的端口关闭，则返回TCP RST分组（b）；
- 如果UDP报文中的目的端口关闭，则返回ICMP目的不可达的分组（c）。

蜜罐检测和隔离方法

- 蜜罐是一个网络中不提供任何真实服务，但却是一个易攻击(例如对于公开的系统漏洞没有打补丁)的主机或应用系统，这是一类安全资源，其价值在于被探测、攻击和捕获。
- 由于蜜罐是没有任何正常的服务请求，任何对于蜜罐的服务请求都是异常的。蜜罐获得的数据较少，但价值很高。可以检测盲扫描、攻击名单扫描、拓扑类蠕虫，但无法检测被动蠕虫。



基于分组内容的蠕虫检测

- 基于分组传递信息的检测可以检测对于网络协议栈的漏洞攻击、或者对于主机的服务漏洞的攻击，这是基于网络的攻击检测。
- 而针对应用漏洞的攻击检测则无法基于分组传递行为，必须基于分组内容进行检测。因为这类攻击的连接都已经正常建立，分组传递不再有任何异常。
- 基于分组内容的检测包括：对于正常分组内容长度的统计分析；分组的特定应用（与分组的端口号相关）与分组内容长度关联的统计分析等。

蠕虫的隔离

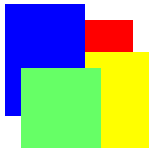
- 降速：采用反馈回路延迟可疑的流量，在不同网络功能层采取限速技术等。由于蠕虫传播速度极快，这些降速机制必须是自动的，最好是能够自动阻隔。
- 阻隔：当发现类似蠕虫行为，则蠕虫行为发起的网络源（包括主机源、网关源等）必须与其他网络隔离，避免更多的网络或主机被感染。
- 诱捕：采用在蜜罐主机中部署多个虚拟机和虚拟地址、端口的方式，使得进入蜜罐的蠕虫无法再向外传播。

蠕虫的阻隔方法

- 蠕虫的阻隔与降速是联动使用的：
  - 当检测到蠕虫行为的警告达到第一级阈值，则系统首先是降速，避免误报；
  - 当情况恶化并且达到了第二级警告阈值，则采用阻隔。
- 阻隔包括两类方法：
  - 基于分组内容的阻隔：丢弃携带蠕虫代码特征内容的分组；丢弃扫描到关闭端口而报错的分组。
  - 基于分组地址的阻隔：对于发出蠕虫的主机地址进行阻隔，通常在网关上设立黑名单。

网络攻击防范的核心技术分析

- 网络攻击防范的核心技术：确保网络正常运行
  - 核心内容包括：识别所有可能的网络攻击，采用不变应万变的方法，增强自身系统功能系统和防御系统能力，最大限度地防范所有可能的网络攻击。
- 不变应万变的方法：基于网络系统行为分析的攻击防范方法
  - 这种网络攻击防范方法包括：首先是网络系统行为数据的采集、汇聚和融合技术，其次是网络系统行为数据的挖掘技术，然后是网络攻击的识别和防范技术。



重点回顾

- 网络攻击分类
- 网络攻击检测分类
- 网络攻击检测方法
- 异常检测模型及算法
- 网络蠕虫及其检测

思考题

(1) 网络侵入是否一定是网络攻击？网络攻击是否一定需要进行网络侵入？为什么？

(2) 网络攻击目前可以分成哪几种类型？试对照网络安全的3个特性，分析这几类网络攻击的意图有何不同。

(3) 现代网络攻击分成哪几类：这几类网络攻击有何特点？

(4) 目前有哪几种网络攻击分类方法？这些攻击分类方法各自有何优点和不足？

思考题(续)

（5）异常检测系统识别网络系统中的异常行为的原理是什么？异常检测的实现系统主要包括哪些功能模块？

（6）什么是恶意代码？什么是互联网蠕虫？互联网蠕虫与通常的计算机病毒有何本质区别？

（7）根据互联网蠕虫的网络中的行为，可以将互联网蠕虫的特征分成哪几类？

（8）如何根据互联网蠕虫的特征进行互联网蠕虫检测和隔离？最不易造成互联网蠕虫检测误判的是哪种检测方法？