第八章 入侵检测

- 8.1 入侵者
- 8.2 入侵检测
- 8.3 分析方法
- 8.4 基于主机的入侵检测
- 8.5 基于网络的入侵检测
- 8.6 分布式或混合式入侵检测
- 8.7 入侵检测交换格式
- 8.8 蜜罐
- 8.9 实例系统：Snort

8.1 入侵者

- 入侵者对于某些形式的黑客技术的使用是关键的安全威胁之一，这里的入侵者通常指的是黑客或者破解者。
- 威瑞森（Verizon）[VERI16]根据他们所做的调查，指出92%的破坏是由外部人员造成的，14%是内部人员所为，其中某些破坏同时涉及到外部人员和内部人员；他们还指出内部人员应该对少数非常大的数据集损坏负责。
- 赛门铁克（Symantec）[SYMA16]和威瑞森[VERI16]同时还指出，不但恶意黑客行为普遍在增加，而且针对组织中的个人及他们使用的IT 系统的攻击也在增加。这种趋势凸显了使用深度防御策略的必要性，因为这类有目标的攻击可能有专门的设计来绕过诸如防火墙和基于网络的入侵检测系统（Intrusion detection systems, IDS）之类的边界防御。

8.1 入侵者-网络罪犯

- 他们是个人或者以金钱回报为目的的犯罪组织的成员。
- 为了达到获利的目的，他们的行为可能包括身份窃取、金融凭证窃取、公司间谍、数据窃取或者数据勒索。
- 他们通常很年轻，一般在网络上[ANTE06]进行交易。
- 他们一般是东欧、俄罗斯或者东南亚的黑客，在类似于DarkMarket.org 或theftservices.com 这样的地下论坛会面、交流心得、买卖数据和合作攻击。
- 不少诸如[SYMA16]这样的年度报告已经说明了这类网络犯罪行为导致的巨大且还在增长的损失，因此有必要采取措施来解决这类威胁。

8.1 入侵者-活动家

- 他们通常是工作在内部的个人，或者更大的外部攻击者组织的一员。
- 他们的动机通常是社会或者政治事业。
- 他们也作为黑客主义者而为人们所熟知，但技能水平通常很低。
- 他们攻击的目的主要是促进和宣传他们的事业，通常采取的手段是破坏网站、拒绝服务攻击、窃取和散布能导致攻击目标妥协或者对其进行负面宣传的数据等。
- 最近众所周知的例子，包括Anonymous和LulzSec等组织从事的活动、切尔西（Chelsea）（从前叫布兰德利，Bradley）·曼宁（Manning）和爱德华·斯诺登（Edward Snowden）从事的活动。

8.1 入侵者-国家资助的组织

- 他们是由政府所资助的黑客组织。
- 目的是进行谍报或者破坏活动。
- 这类活动就是人们所熟知的高级持续威胁（Advanced Persistent Threats, APTs）活动，因为在这个类别中许多攻击涉及长时间的隐蔽性和持久性。
- 近期的报告，比如[MAND13]和爱德华·斯诺登揭露出的信息等，都表明了一些国家及他们的盟友的攻击活动的普遍性和广泛性。

8.1 入侵者-其他

- 他们是以上未列出的以其他目的为动机的黑客。
- 包括以技术挑战或者同行的尊敬和名声为目的的典型黑客或破解者，以及那些负责寻找新的缓冲区溢出漏洞[MEER10]的黑客。
- 另外，由于攻击工具的广泛可用性，还有一类“嗜好性黑客（hobby hackers）”使用这些工具来探究系统和网络的安全性，他们是上面那几类黑客的潜在新生力量。

8.1 入侵者技能等级-学徒

- 他们是那些拥有最低技术水平的黑客，仅仅会使用现有的攻击工具包。
- 他们很可能在攻击者中占最大比例，包括了许多犯罪者和活动家黑客。
- 考虑到他们使用现有的攻击工具，因此这些攻击者也非常容易防御。
- 因为他们使用现有的脚本（工具），人们也称他们为“脚本小子”（script-kiddies）

8.1 入侵者技能等级-训练有素者

- 他们是那些拥有足够技术的黑客，可以修改和扩展攻击工具来使用新发现的或者购买的漏洞；或者来攻击不同的目标组织。
- 他们也可能发现和利用与已知漏洞相类似的新漏洞。
- 许多有这些技能的黑客可以调整攻击工具为他人使用，并且存在于上面列出的各类入侵者中。
- 攻击工具的改变会使识别和防御这些攻击更加困难。

8.1 入侵者技能等级-高手

- 他们是那些拥有高级技术的黑客。
- 有能力发现标志性的新漏洞，或者编写全新的强力攻击工具包。
- 一些更为知名的典型黑客都可以归为这一类，很显然，他们中的一部分人会被某些政府组织所雇佣，从事APT攻击活动。
- 这使防御这类攻击最为困难。

8.1 入侵实例

实施电子邮件服务器的远程根目录泄露	破坏一个Web 服务器	猜测和破解口令	复制一个存有信用卡账号的数据库
在未授权的情况下浏览敏感数据，包括工资记录和医疗信息等	在工作站上运行数据包嗅探器来捕获用户名和口令	利用匿名FTP服务器的权限错误发送盗版软件和音乐文件	拨号到一个不安全的调制解调器，以获得内网的访问权限
	伪装成管理人员，呼叫帮助平台，重置该管理人员的电子邮件口令并学习新的口令	在未授权的情况下使用一个无人值守的、已登录的工作站	

8.1.1 入侵者行为

目标获取和信息收集	• 攻击者使用公开可用的信息（技术和非技术信息）以及使用网络探索工具来映射目标资源，以识别和表征目标系统。
初始访问	• 对目标系统的初始访问通常可以利用远程网络漏洞，通过远程服务中使用的猜测弱身份验证凭证；或通过系统中的恶意软件安装，使用某种形式的社会工程或路过式下载攻击。
权限提升	• 这是在系统上采取的行为，提升攻击者可用的权限以实现其在目标系统上的期望目标。
信息收集或系统利用	• 攻击者访问或修改系统信息或资源，或导航到另一个目标系统的行为。
维持访问权限	• 攻击者安装后门或者其他恶意软件，或者通过改变认证证书或者改变系统的其他配置，以使攻击者在初始攻击后继续访问。
覆盖痕迹	• 攻击者破坏或者修改审计日志来消除攻击活动的证据，或者使用工具包或其他措施来隐藏悄悄地安装的文件或代码。

8.1.1 入侵者行为

表 8-1 入侵者行为示例

（a）目标获取和信息收集
探测公司网站来获取相关信息，例如公司组织结构、人员、关键系统，以及具体的（specific）网站服务器和采用的操作系统等细节。 使用 DNS 查询工具收集目标网络的相关信息，这些查询工具包括 dig、host 等，或者查询 WHOIS 数据库。 使用 NMAP 等工具映射网络以获取可访问的服务。 向客户服务联系人发送查询电子邮件，查看有关邮件客户端、服务器和所使用操作系统的信息响应，以及个人响应的详细信息。 确定潜在的有漏洞的服务，例如有漏洞的 Web 内容管理系统（Web CMS）
（b）初始访问
暴力破解（猜测）用户的 Web 内容管理系统（CMS）口令。 利用 Web CMS 插件的漏洞来获取系统访问权限。 将具有链接到 Web 浏览器漏洞的钓鱼式电子邮件发送给关键人员
（c）权限提升
使用漏洞扫描系统以查找本地可以利用的应用程序。 攻击任意一个带漏洞的程序来获得高级访问权限。 安装嗅探程序来捕获管理员口令。 利用捕获到的管理员口令来访问特权信息

8.1.1 入侵者行为

（d）信息收集或系统利用
扫描文件来寻找想要的信息。 将大量的文档传送到外部存储库（repository）。 使用猜测的或者捕获的口令来访问网络中的其他服务器
（e）维持访问权限
安装远程管理工具或者带后门的 rootkit 来方便以后的访问。 在以后对网络进行访问时使用管理员口令。 修改或者破坏系统上运行的反病毒程序或者 IDS 程序
（f）覆盖痕迹
使用 rootkit 隐藏安装在系统中的文件。 编辑日志文件来移除入侵过程中生成的相关记录

8.2 入侵检测-术语补充

- 安全入侵：未经授权绕过系统安全机制的行为。

- 入侵检测：一种硬件或软件功能，该功能用于收集和分析计算机或网络中各个区域的信息，以识别可能的安全入侵。

8.2 入侵检测系统IDS

- IDS包括三个逻辑组件:
  - 传感器（sensors）：传感器负责收集数据。传感器的输入可以是包含入侵证据的系统的任何一部分。传感器输入的类型包括网络数据包、日志文件和系统调用痕迹。传感器收集并向分析器转发这些信息。
  - 分析器（analyzers）：分析器从一个或多个传感器或其他分析器接收输入。分析器负责确定是否发生了入侵；此组件的输出表明是否发生了入侵，可以包含支持入侵发生结论的证据。分析器可以提供指导，用于判断什么活动是入侵导致的。传感器的输入也可以被存储起来用于将来的分析，这些输入可以在存储器或者数据库组件中进行检查。
  - 用户接口（user interface）：IDS的用户接口使用户能够查看系统输出或控制系统的行为。在某些系统，用户接口可以看作是经理、主管或者控制台组件。

8.2 入侵检测系统IDS

- IDS通常根据分析数据的来源和类型进行分类，如：
  - 基于主机的IDS（Host-based IDS, HIDS）：监测一台主机的特征和该主机发生的与可疑活动相关的事件，例如进程识别器、进程产生的系统调用等，用作可疑活动的证据。
  - 基于网络的IDS（Network-based IDS, NIDS）：监测特定的网段或设备的流量并分析网络、传输和应用协议，用以识别可疑的活动。
  - 分布式或混合式IDS（Distributed or hybrid IDS）：将来自大量传感器（通常是主机和基于网络的）的信息组合在一个中央分析器中，以便更好地识别和响应入侵活动。

8.2.1 基本原理

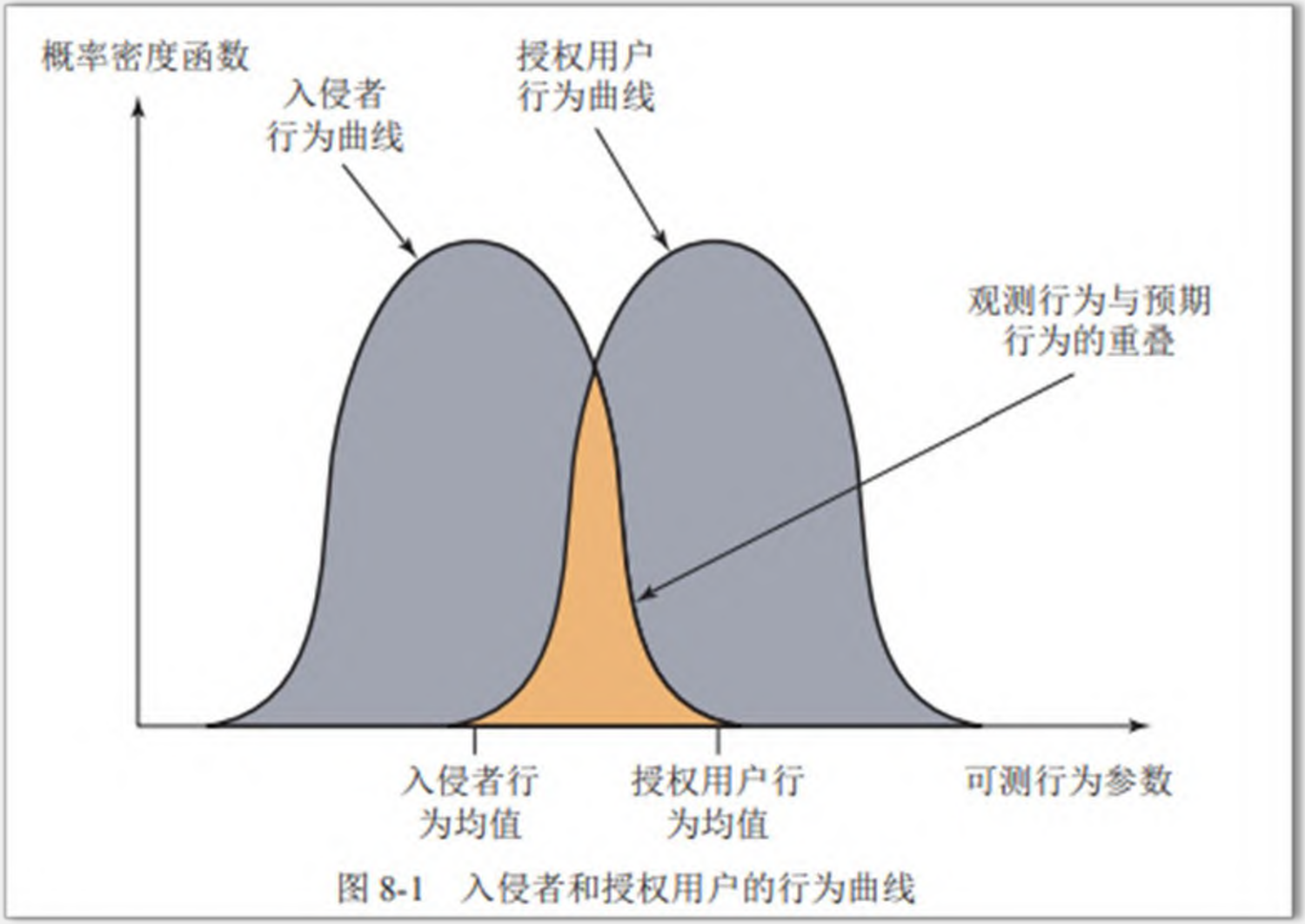


图 8-1 入侵者和授权用户的行为曲线

8.2.2 基率谬误

- 为了实用性，IDS应该能检测到绝大多数的入侵，同时保持可接受级别的误报率。如果只检测到有限比例的实际入侵，则系统给人以安全的假象。另外，如果系统在没有入侵的时候频繁报警（误报），则系统管理员要么开始忽略报警，要么浪费很多时间分析误报。
- 遗憾的是，由于所涉及的概率性质，很难同时满足具有高检测率和低误报率的标准。一般来讲，如果实际入侵数比系统的合法使用数低，则误报率将很高，除非测试用例是很容易区别的。这是基率谬误（base - rate fallacy）现象的一个示例。[AXEL00]对已有的IDS进行了研究，指出当前入侵检测系统不能解决基率谬误的问题。

8.2.3 IDS要求

- [BALA98]列出理想的IDS必须满足的条件如下：

- 能够不间断地运行，而且人的参与尽可能少。
- 具有容错功能，系统崩溃时，它必须能够很快恢复和重新初始化。
- 抵御破坏。IDS必须能够监测自身，检测是否已被攻击者修改。
- 对于正运行的系统增加最小的开销。
- 能够根据被监测系统的安全策略进行配置。
- 能够自动适应系统和用户行为变化。
- 能够扩展以监测更多的主机。
- 能够提供很好的服务降级，也就是说，如果IDS的某些组件停止工作，无论出于何种原因，其余部分应受到尽可能少的影响。
- 允许动态重新配置；即能够重新配置IDS，而不必重新启动。

8.3 分析方法

- IDS通常使用以下几种方法之一来分析传感器得到的数据进而检测到入侵：
  ① 异常检测（anomaly detection）：包括采集有关的合法用户在某段时间内的行为数据，然后分析当前观察到的行为，以较高的置信度确定该行为是合法用户还是入侵者的行为。
  ② 特征或启发式检测（Signature or Heuristic detection）：使用一组已知恶意数据模式（特征）或者攻击规则（启发式）组成的集合来与当前的行为进行比较，最终确定这是否是一个入侵者。这种方法也被称为误用检测，仅仅可以被用来识别有模式或者规则的已知攻击。
- 实质上，为了识别恶意或未经授权的行为，异常方法都旨在定义正常或预期之中的行为。特征或基于启发式的方直接定义恶意或未经授权的行为，并可以快速且有效地识别已知的攻击。然而，只有异常检测才能够检测出未知的0-day 攻击，这是因为它是用已知的正常行为去识别异常行为。由于存在这种优势，如果不是我们下面讨论的收集和分析数据的困难性，以及较高的误报率，很明显异常检测将是首选的方法。

8.3.1 异常检测

统计法

对被观测行为的分析使用单因素、多因素、或者观察指标的时序模型。

基于知识法

使用专家系统，根据一组对合法行为建模的规则对观察到的行为进行分类。

机器学习法

使用数据挖掘技术从训练数据中自动确定合适的分类模型。

8.3.2 特征或启发式检测

- 特征或启发式检测是通过观测系统中的事件来检测入侵。该方法是利用一组特征模式数据或者一组特征化的规则来确定观测到的数据究竟是正常的还是异常的。
- 特征方法是用一个大的已知恶意数据模式的集合去匹配系统中或发送到网络中的数据。
  - 特征集合需要足够大，这样可以在尽可能减小误报率的同时，检测到最多的恶意数据。
  - 该方法被广泛应用于反病毒产品、网络流量扫描代理、以及NIDS中。它的优点是相对较低的时间和资源开销，以及它的广泛可用性。
  - 缺点则是需要大量的精力来实时识别和检查新的恶意软件并为它们创建特征，以便系统能够识别它们。此外它也没有办法检测到没有任何特征可言的0-day攻击。

8.3.2 特征或启发式检测

- 基于规则的启发式识别是采用规则来识别已知的渗透或者利用已知漏洞进行的渗透。规则还可用来识别可疑行为，即使该行为并未超出已建立的可用模式范围。
  - 通常，系统中使用的规则与特定的机器和操作系统有关。
  - 开发这样的规则最有效的方法是来分析从Internet上收集到的攻击工具和脚本。
  - 这些规则可以作为由知识渊博的安全人员制定的规则的补充。正常的过程是采访系统管理员和安全分析员以收集一套已知的渗透场景和威胁目标系统安全的关键事件。

8.4 基于主机的入侵检测

- 基于主机的IDS（Host-based IDSs, HIDSs）向易受攻击的或敏感的系统中添加专用的安全软件层，实例包括数据库服务器和管理系统。基于主机的IDS以多种方式监测系统上的活动，目的是检测系统上的可疑行为。在某些情况下，IDS可以在任何损害发生之前阻止攻击，但它的主要目的还是检测入侵、记录可疑事件，并发送警报。
- HIDS的主要优点是，它可以检测外部和内部入侵，这一点是基于网络的IDS或者防火墙所不及的。正如我们先前所言，基于主机的IDS可以使用异常、特征、启发式方法来检测受监视的主机上的未授权的行为。我们首先介绍一下用于HIDS的常见的数据源和传感器，然后继续讨论异常、特征和启发式方法如何应用在HIDS中，最后再研究分布式HIDS。

8.4.1 数据源和传感器

- 入侵检测的一个基本组件是用来收集数据的传感器。一些用户不间断的活动记录必须作为输入提供给IDS的分析组件。常见的数据源包括：
  - 系统调用踪迹（System call traces）
  - 审计（日志文件）记录（Audit (log file) records）
  - 文件完整性校验和（File integrity checksums）
  - 注册表访问（Registry access）
- 传感器从选定的数据源来收集数据，从中过滤掉不需要的信息，将其记录为标准化的格式，最终发送结果到本地或者远程的IDS分析器中。
- 相对于入侵检测而言，审计记录在计算机安全中起到了更为广泛的作用。

8.4.2 异常HIDS

- 由于在UNIX和Linux系统收集合适数据较为容易，因此基于异常的HIDS主要是在UNIX和Linux系统上实现的。
- 尽管一些早期的工作使用了审计或财务记录，但主要还是基于系统调用踪迹。系统调用为应用程序提供了一系列和底层操作系统交互的函数，是程序访问系统内核的方法。因此它们提供了进程活动的详细信息，这些信息可以被用来确定行为是正常的还是异常的。
- 之后，系统调用记录会由一个适当的决策引擎进行分析。
- 传统上讲，Windows 系统并没有使用基于异常的HIDS，这是由于广泛使用动态链接库（Dynamic Link Library，DLL）造成的。

8.4.3 特征或启发式HIDS

- 基于特征或启发式的HIDS被广泛使用，尤其常见于反病毒程序（A/V）中，或者更准确地说，常见于反恶意软件产品中。基于特征或启发式的HIDS在客户端系统和越来越多的移动设备上被非常普遍地使用，并且还被纳入防火墙上的邮件、Web应用代理及基于网络的IDS中。它们一般使用文件特征数据库（即在已知恶意软件中发现的数据模式），或者使用描述已知恶意行为特征的启发式规则。
- 这些产品在检测已知恶意软件方面非常高效，然而，它们没有能力检测缺乏相关特征或启发式规则的0 - day攻击。它们在Windows系统上被广泛使用，并将继续成为入侵者的攻击目标。

8.4.4 分布式HIDS

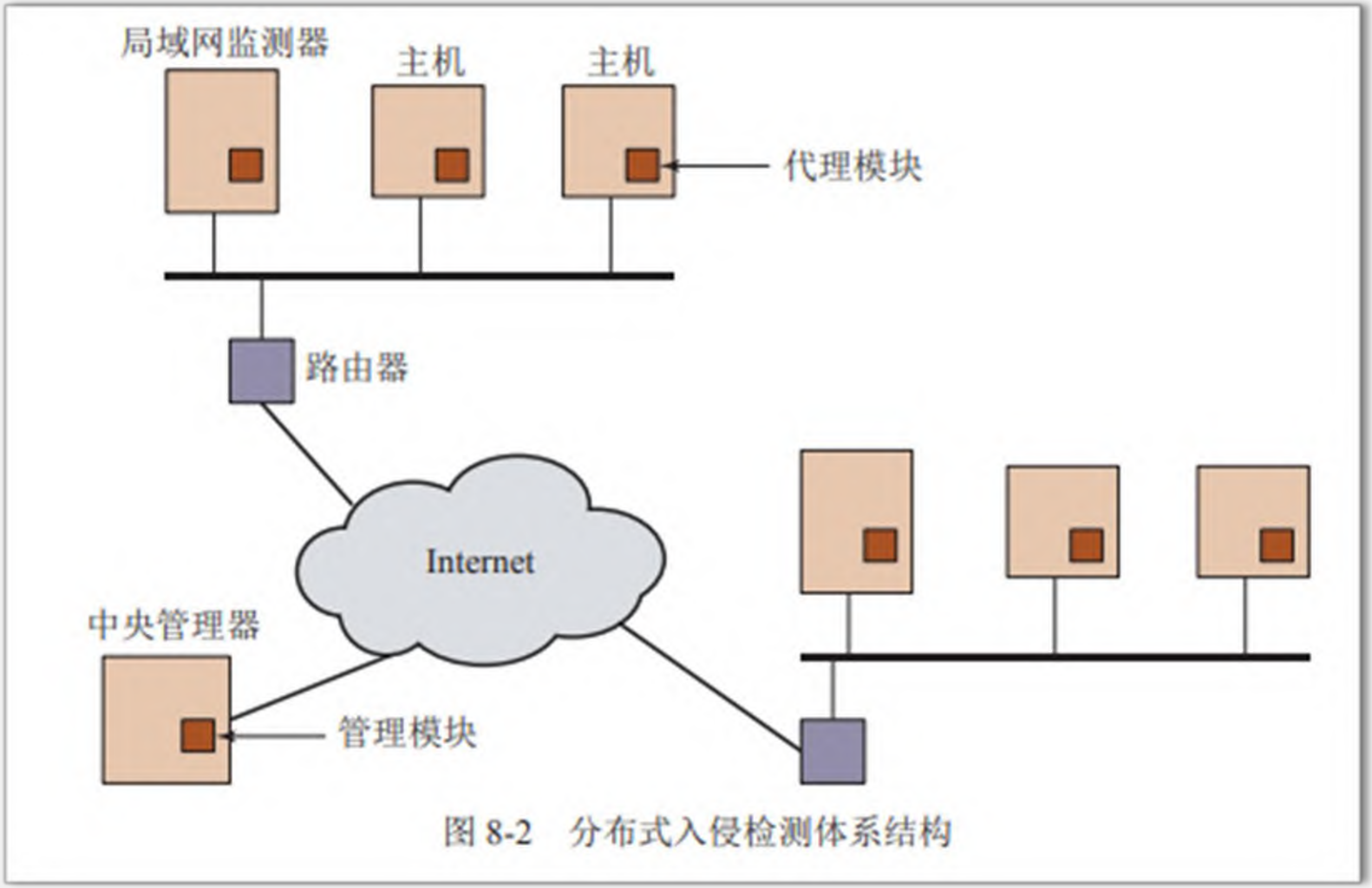


图 8-2　分布式入侵检测体系结构

8.4.4 分布式HIDS

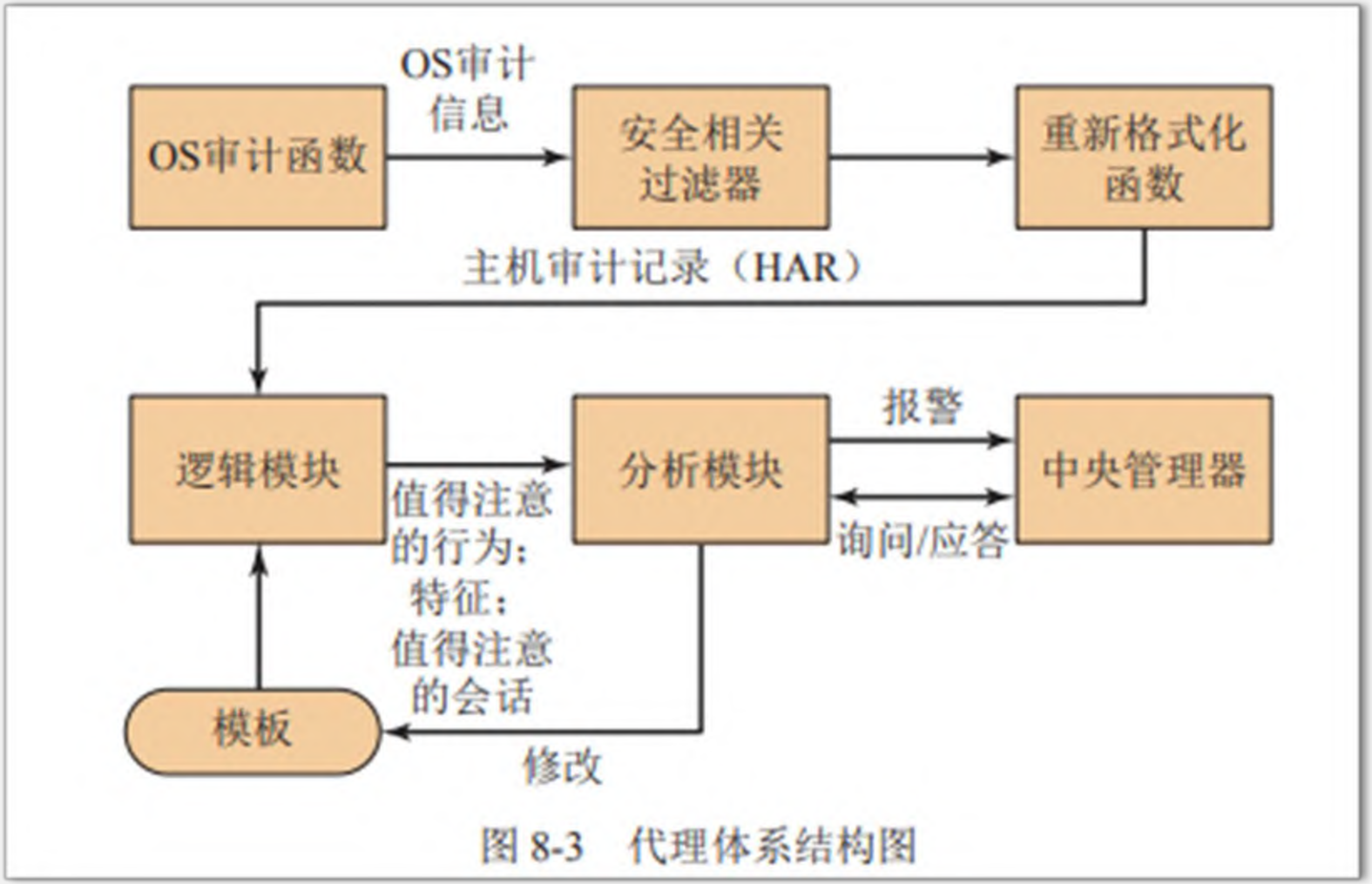


图 8-3　代理体系结构图

8.5 基于网络的入侵检测

- 基于网络的IDS（即NIDS）监控的是一个网络或多个相互连接的网络上选定的位置的网络流量。
- NIDS实时或接近实时地检查流量数据包，试图检测入侵模式。
- NIDS可以检测网络层、传输层和/或应用层协议的活动。
- 基于网络的IDS与基于主机的IDS是不同的，NIDS检测网络上流向潜在的易受攻击的计算机系统的数据包流量，而基于主机的IDS系统检测的是主机上的用户和软件活动。
- 典型的NIDS设备包括：大量传感器用来监控数据包流量、一个或多个服务器负责NIDS管理功能以及一个或多个管理控制台提供人机交互的接口。
- 分析流量模式从而检测入侵的工作可以在传感器、管理服务器或在二者上组合完成。

8.5.1 网络传感器的类型

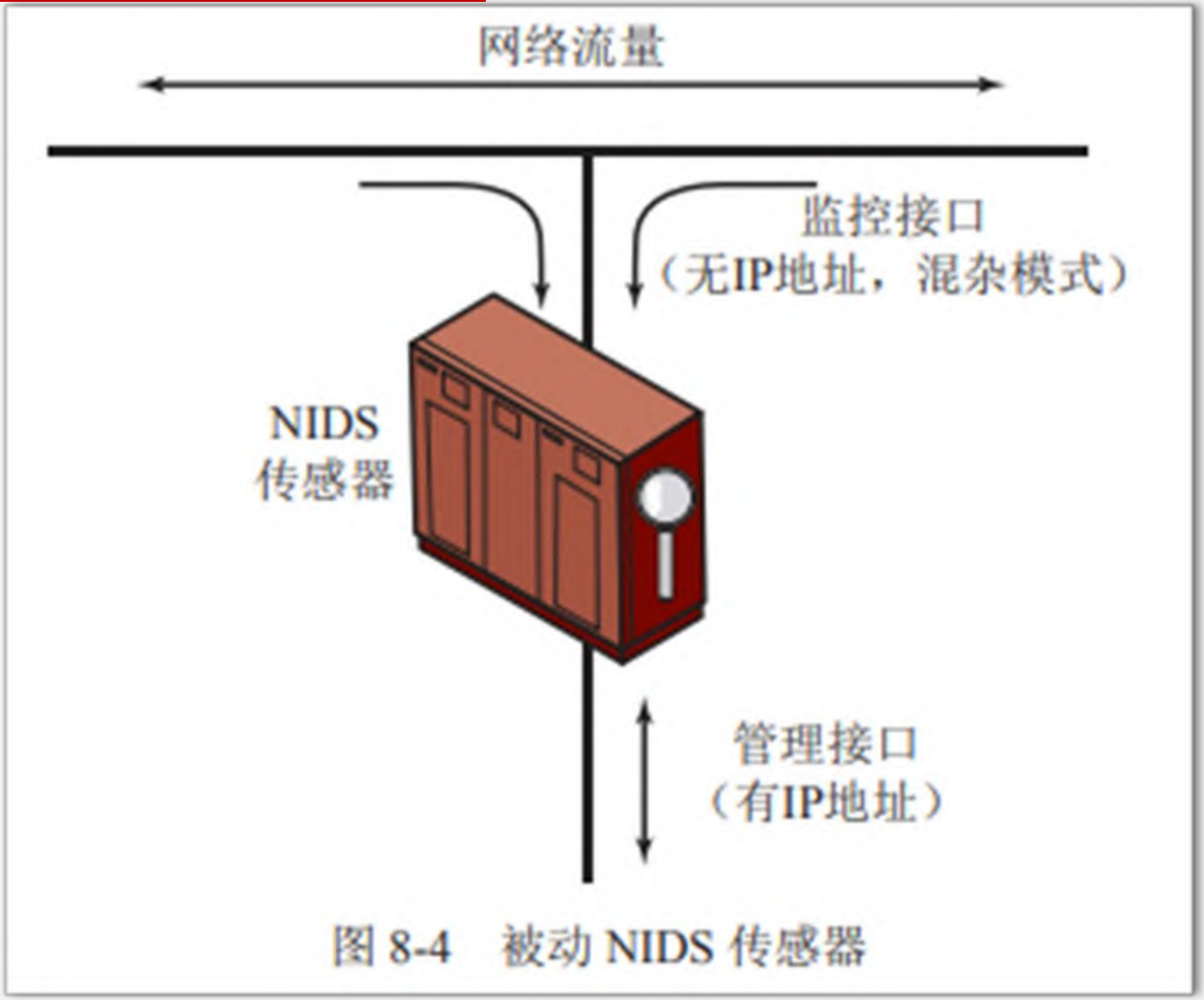


图 8-4　被动 NIDS 传感器

8.5.2 NIDS传感器部署

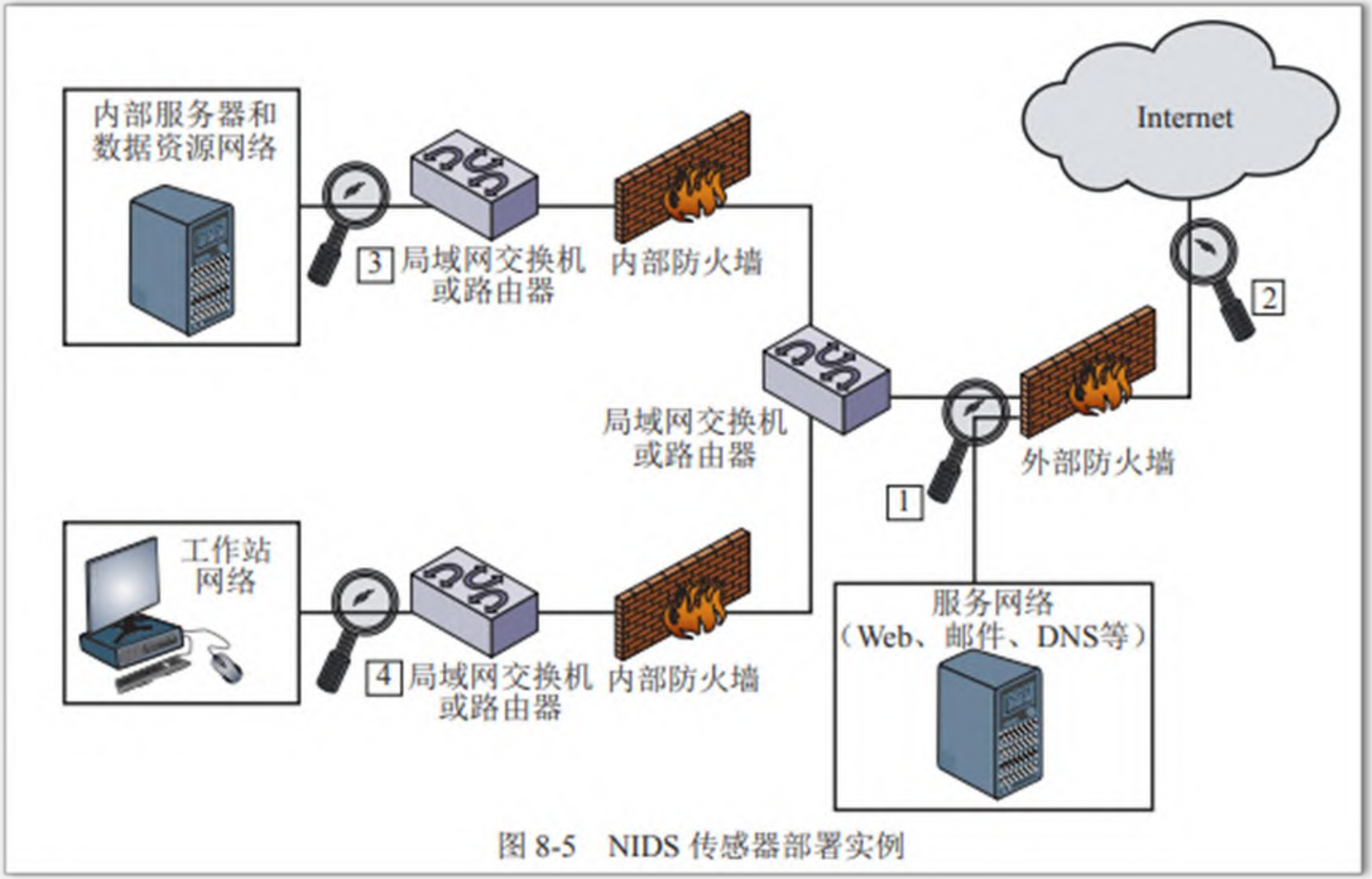


图 8-5　NIDS 传感器部署实例

8.5.3 入侵检测技术

适合于特征检测的攻击类型

- 应用层侦察和攻击
- 传输层侦察和攻击
- 网络层侦察和攻击
- 意外应用程序服务
- 策略违背

适合于异常检测的攻击类型

- 拒绝服务（DoS）攻击
- 扫描
- 蠕虫

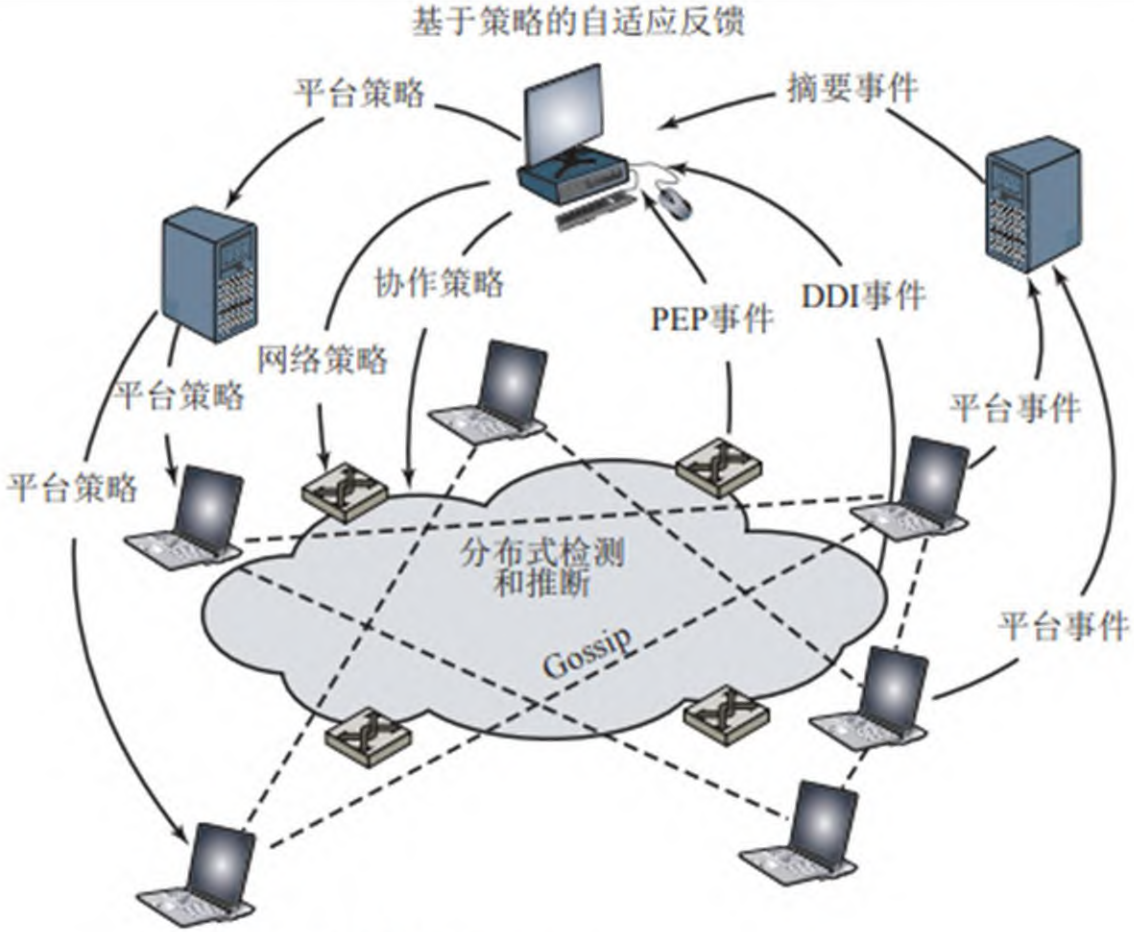
8.5.3 入侵检测技术

- 状态协议分析（SPA）：NIST SP 800-94详细描述了这种异常检测技术，其中检测是通过比较观测的网络流量与预先制定的、供应商提供的正常的流量特征实现的。
- 这与基于组织特定的流量特征的异常检测技术不同。
- SPA通过推断和追踪网络、传输和应用协议的状态，保证网络活动按预期发展。SPA的一个主要缺点是它所需要高的资源占用。

8.5.4 警报日志记录

- 当传感器检测到潜在的危险时，它将发送一个警报并记录与事件相关的信息。NIDS分析模块可以使用此信息来优化入侵检测参数和算法。安全管理员可以使用此信息来设计保护技术。由NIDS传感器记录的典型信息如下：
  - 时间戳（通常是日期和时间）
  - 连接或会话ID号（通常是分配给每个TCP连接或无连接协议的数据包组的连续的或唯一的号码）
  - 事件或警报类型
  - 分级（如优先级、严重性、影响和信任等）
  - 网络层、传输层和应用层协议
  - 源和目的IP地址
  - 源和目的TCP或UDP端口，或者ICMP类型和代码
  - 通过连接传输的字节数
  - 已解码的有效载荷数据，如应用程序请求和响应
  - 状态相关信息（如经过身份验证的用户名）

8.6 分布式或混合式入侵检测



PEP = 策略强制点
DDI = 分布式检测和推断

图 8-6　一个自治企业安全系统的整体体系结构

8.7 入侵检测交换格式

- 为了促进可以运行在各种平台和环境的分布式IDS的开发，需要制定支持协同工作能力的标准。这些标准是IETF入侵检测工作组的工作重点，旨在为入侵检测和响应系统，以及需要与其它机器交互的管理系统的共享信息定义数据格式和交换过程。该工作组在2007年发布了以下RFCs：
  - 入侵检测消息交换要求（Intrusion Detection Message Exchange Requirements, RFC 4766）：这份文档定义了入侵检测消息交换格式（Intrusion Detection Message Exchange Format, IDMEF）的要求，还规定了IDMEF通信协议的要求。
  - 入侵检测消息交换格式（Intrusion Detection Message Exchange Format, RFC 4765）：这份文档描述了入侵检测系统导出信息时的一个数据模型，并解释了使用这个模型的基本原理。文档同时给出了该数据模型使用可扩展标记语言（Extensible Markup Language, XML）的一个实现。另外，XML文档类型定义（Document Type Definition）正在开发当中，示例已经给出。
  - 入侵检测交换协议（Intrusion Detection Exchange Protocol, RFC 4767）：这份文档描述了入侵检测交换协议（Intrusion Detection Exchange Protocol, IDXP），这是在入侵检测系统之间进行数据交换的应用层协议。IDXP支持基于面向连接协议的相互授权、完整性和可信性。

8.7 入侵检测交换格式

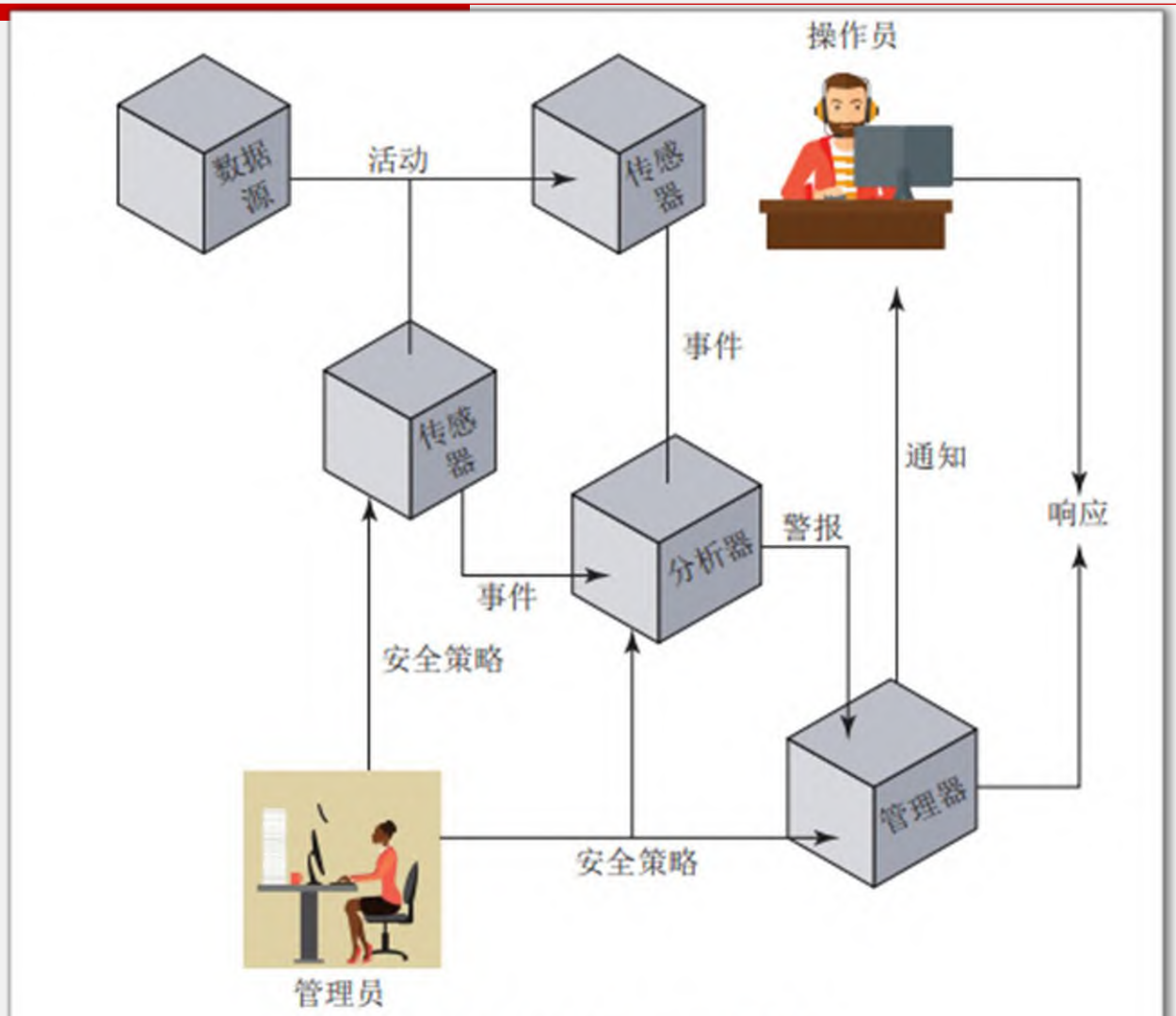


图 8-7 入侵检测消息交换模型

8.8 蜜罐

- 入侵检测技术中一个特别的组件是蜜罐。蜜罐是障人耳目的系统，是为引诱潜在的攻击者远离关键系统而设计的。蜜罐的功能包括：
  - 转移攻击者对重要系统的访问。
  - 收集有关攻击者活动的信息。
  - 鼓励攻击者在系统中能够逗留足够长的时间，以便于管理员对此攻击做出响应。
- 这些系统充满了虚构的信息，这些信息看起来很有价值，但系统的合法用户无法访问。因此，任何对蜜罐的访问都是可疑的。蜜罐系统装备了敏感的监控器和事件记录器，用于检测这些访问和收集有关攻击者的活动信息。因为任何针对蜜罐的攻击在攻击者看来是都是成功的，所以管理员有时间来调动、记录并跟踪攻击者而不必暴露生产系统。
- 蜜罐是一种没有产出的资源。网络以外的任何人与蜜罐进行交互都没有合法的理由。因此，任何与蜜罐系统通信的尝试很可能是一个探测、扫描或者攻击。相反，如果一个蜜罐发起对外通信，则系统可能已被破坏。

8.8 蜜罐

- 蜜罐通常分为低交互蜜罐和高交互蜜罐。
  - 低交互蜜罐（Low interaction honeypot）：该类蜜罐是由能够模拟特定IT服务或系统的软件包构成，它足以提供一种真实的初级交互，但是却无法提供所模拟服务或系统的全部功能。
  - 高交互蜜罐（High interaction honeypot）：该类蜜罐是一个带有完整操作系统、服务、以及应用程序的真实系统，被部署在攻击者能够访问的地方。
- 高交互蜜罐是一个更为真实的目标，很有可能消耗掉攻击者更长的时间。但是它需要极大的资源，并且一旦被攻破，它可以被用来发起对其他系统的攻击。对于运行蜜罐的组织来说，很有可能导致麻烦的法律或声誉问题。
- 低交互蜜罐提供了一个低真实度的目标，它能够在攻击早期识别一些使用本章前面讨论的攻击技术的入侵者。通常来说，这类蜜罐作为一个为即将发生的的攻击提供报警功能的分布式IDS的组件，这样已经足够了。

8.8 蜜罐

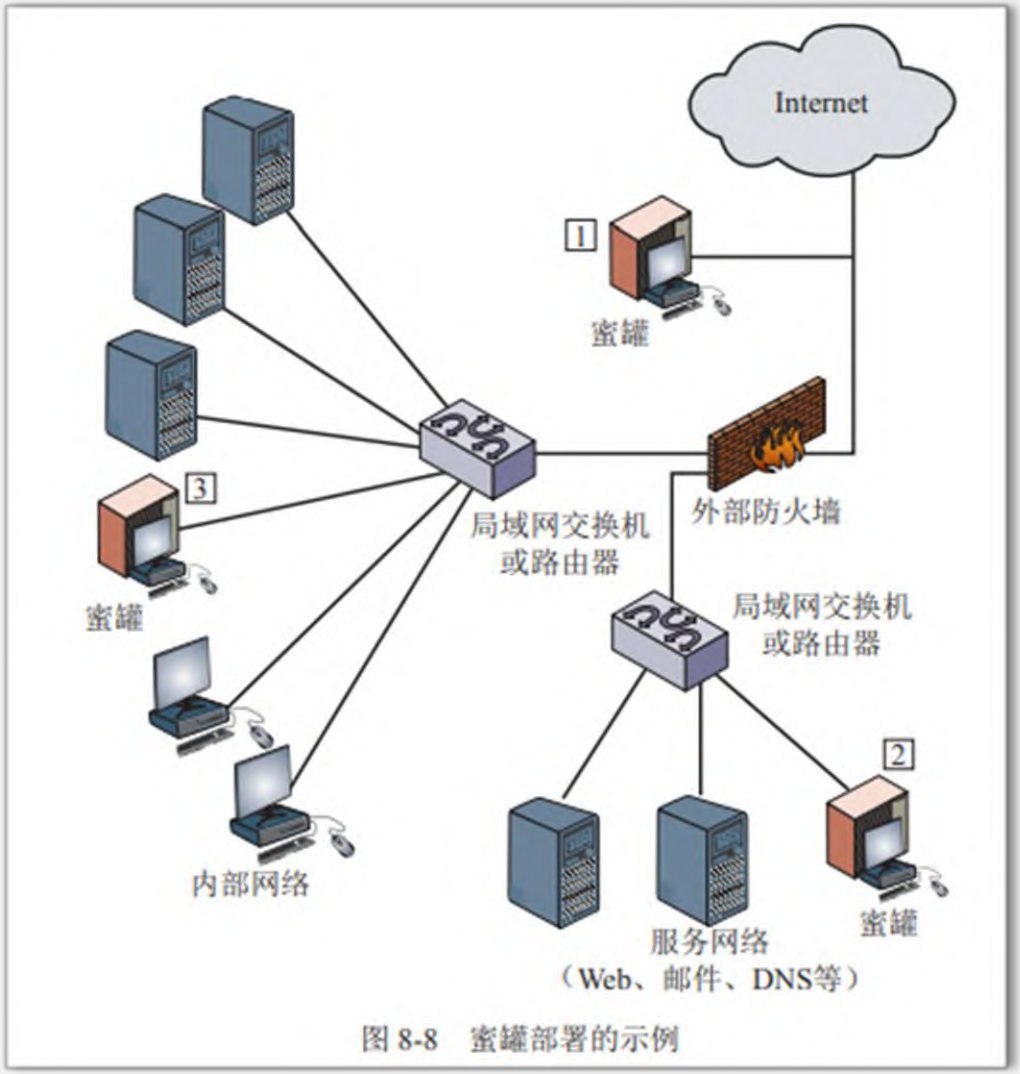


图 8-8　蜜罐部署的示例

8.9.1 Snort体系结构

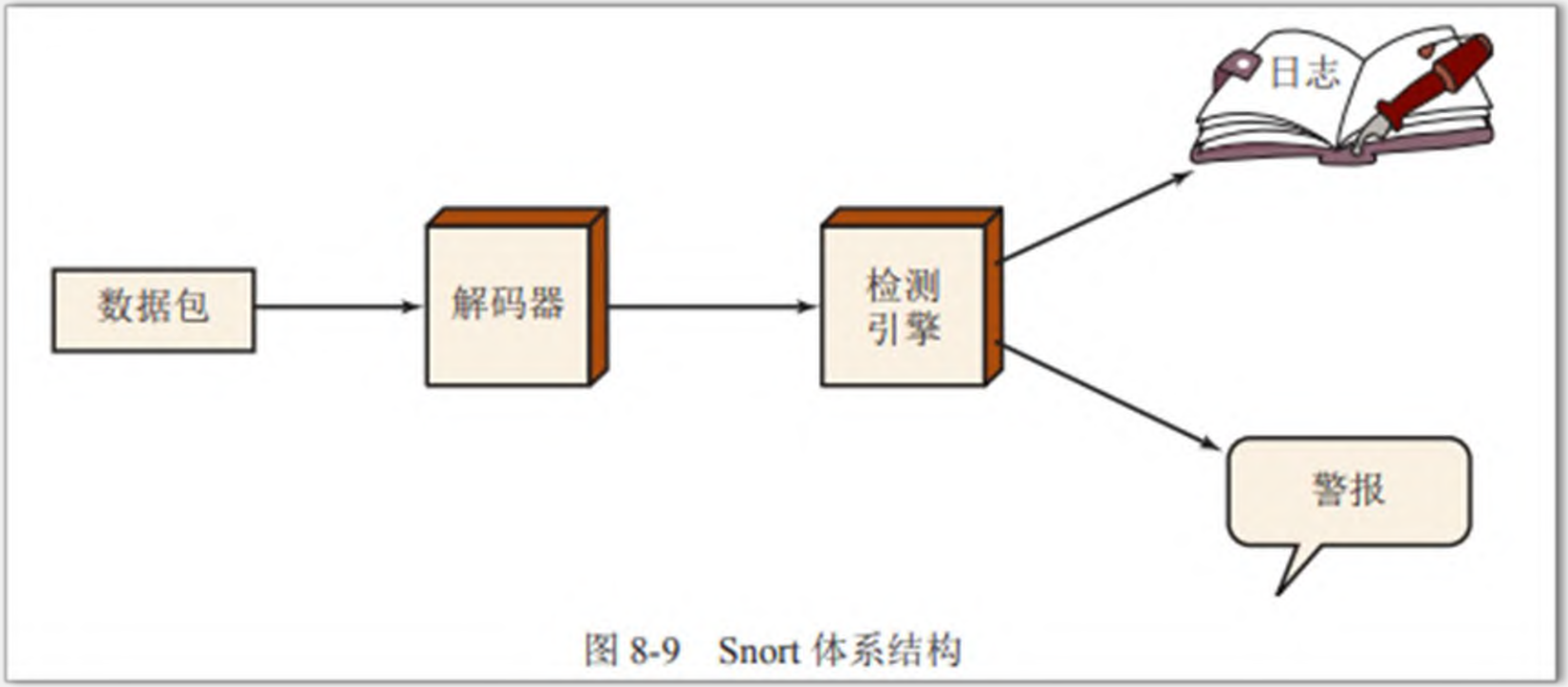


图 8-9　Snort 体系结构

8.9.2 Snort规则

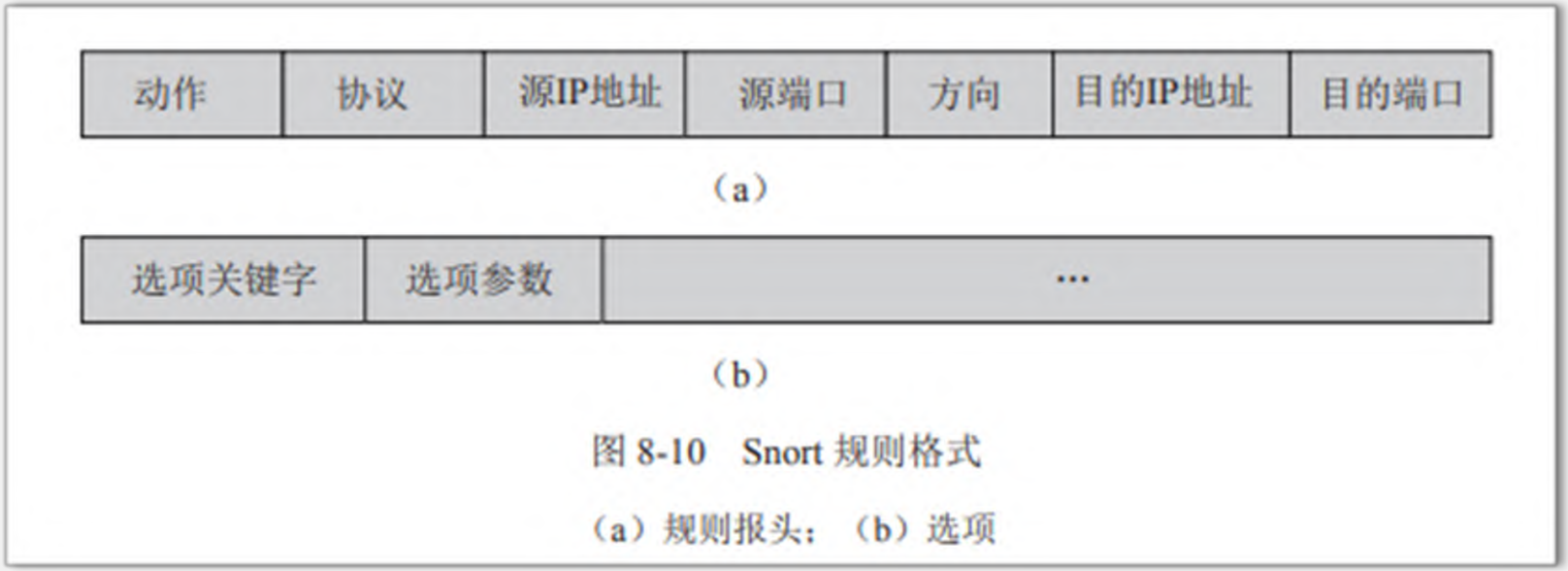


图 8-10　Snort 规则格式

（a）规则报头；（b）选项

8.9.2 Snort规则

表 8-2 Snort 规则动作

动作	说明
alert	使用所选的报警方式生成警报，再将数据包写入日志
log	将数据包写入日志
pass	忽略数据包
activate	报警后再激活另一个 dynamic 规则
dynamic	保持空闲直到被 activate 规则激活，然后作为 log 规则
drop	使 iptables 丢弃数据包并写日志
reject	使 iptables 丢弃数据包并记入日志，然后如果协议是 TCP，则发送 TCP 重置；如果协议是 UDP，则发送 ICMP 端口不可达消息
sdrop	使 iptables 丢弃数据包但不写日志

8.9.2 Snort规则

表 8-3 Snort 规则选项实例

元数据	
msg	当一个数据包生成一个事件时，定义要发送的消息
reference	定义了到外部攻击识别系统的链接，该系统可以提供额外信息
classtype	指出数据包尝试的攻击类型
载荷	
content	使 Snort 对数据包有效载荷中的特定内容（文本或二进制）执行区分大小写的搜索
depth	指定 Snort 在数据包中查找给定模式的搜索深度。depth 修改规则中的前一个 content 关键字
offset	指定 Snort 在数据包中查找给定模式的起始搜索位置。offset 修改规则中的前一个 content 关键字
nocase	Snort 应该在查找给定模式时忽略大小写。nocase 修改规则中的前一个 content 关键字

8.9.2 Snort规则

非载荷	
ttl	检查 IP 的生存时间（time-to-live）值。此选项用于检测 traceroute 尝试
id	检查 IP 的 ID 字段是否为某个特定值。某些工具（漏洞检测、扫描器和其他恶意程序）特别设置该字段用于各种用途，例如值 31337 经常被某些黑客使用
dsize	测试数据包有效载荷的大小。这可以用来检查异常大小的数据包。很多情况下，这对于检测缓冲区溢出是非常有用的
flags	测试 TCP 标志是否为指定设置
seq	寻找指定的 TCP 首部序列号
icmp-id	检查 ICMP ID 值是否为指定值。这很有用，因为某些隐蔽通道的程序在通信时使用静态 ICMP 字段。开发这个选项用来检测 stacheldraht DDoS 代理
后检测	
logto	把与规则相匹配的数据包写入指定的日志文件
session	从 TCP 会话中提取用户数据。很多情况下，查看用户在 telnet、rlogin、ftp 甚至 Web 会话中输入的内容是很有用的