构建基于 Snort 的分布式入侵检测系统

廖光忠　陈志凤

(武汉科技大学计算机科学与技术学院　武汉 430081)

摘　要　分布式入侵检测是入侵检测发展的一个新方向，本文讨论了一种基于 Snort，协同使用 ACID，MySQL，PHP，Apache 的分布式入侵检测系统的设计与实现，并对这一实现进行总结，提出今后的发展方向。

关键词　分布式，Snort，ACID，入侵检测

1　引言

在过去的 20 年里，网络技术在不断发展，攻击者水平也在不断提高，攻击工具与攻击手法日趋复杂多样，促使网络安全产品不断更新换代，使得 IDS 产品从一个简单机械的产品发展成为综合各种技术的智能化产品。我们需要做的就是如何将这些技术有效的融合在入侵检测系统中，并且能够做到与其他网络安全设备协同工作，提高整个系统的安全性。本文讨论一种基于 Snort 的分布式入侵检测系统 DIDS(Distributed Intrusion Detection System)的设计与实现，由 Snort，ACID，php，MySQL，Apach 构成的基于 Snort 的分布式入侵检测系统。

2　Snort 剖析

2.1　简介

Snort 是一个强大的轻量级的网络入侵检测系统。是基于特征检测的 IDS，使用规则的定义来检测网络中有问题的数据包。它具有截取网络数据报文，进行网络数据实时分析、报警，以及日志的能力，能够进行协议分析，对内容进行搜索/匹配。它能够检测各种不同的攻击方式，例如：缓冲区溢出、隐秘端口扫描、CGI 攻击、SMB 探测、OS 指纹特征检测等等。并对攻击进行实时报警，可以将报警信息写到 syslog、指定的文件、UNIX 套接字或者使用 WinPop-up 消息，其日志文件可以是 tcpdump 格式，也可以是解码的 ASCII 格式。此外，Snort 具有很好的扩展性和可移植性。它支持插件体系，可以通过其定义的接口，很方便地加入新的功能。Snort 使用一种灵活的规则语言来描述网络数据报文，因此可以对新的攻击作快速的翻译。

Snort 有三种工作模式：嗅探器、数据包记录器、网络入侵检测系统。嗅探器模式仅仅是从网络上读取数据包并作为连续不断的流显示在终端上。数据包记录器模式把数据包记录到硬盘上。网路入侵检测模式是最复杂的，而且是可配置的。

2.2　Snort 组成

Snort 可以分成 5 个主要的组件：包捕获/解码引擎、预处理器插件、检测引擎、输出插件。第一个是捕包/解码装置。Snort 依赖一个外部捕包程序(libpcap)来抓包。在包被以原始状态捕获后，要送给包解码器。解码器是进入 Snort 自身体系的第一步。包解码器将特殊协议元素翻译成内部数据结构。在最初的捕包和解码完成后，由预处理程序处理流量。许多插入式预处理程序对包进行检测或操作后将它们交给下一个组件：检测引擎。检测引擎是 Snort 的核心模块，对从预处理器发送过来的包依据预先设置的规则进行检查，一旦发现数据包中的内容和某条规则匹配，就通知报警模块。最后一个组件是输出插件，它对可疑行为产生警报。图 1 是 Snort 的体系结构示图。

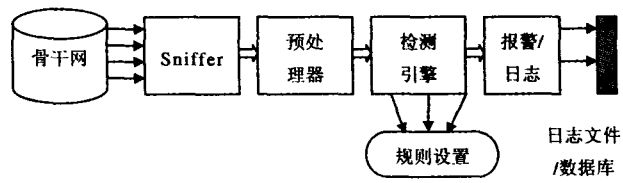

图 1　Snort 的体系结构

2.3　Snort 规则

Snort 规则被分成两个逻辑部分：规则头和规则选项。规则头包含规则的动作，协议，源和目标 IP 地址与网络掩码，以及源和目标端口信息；规则选项部分包含报警消息内容和要检查的包的具体部分。下面是一个规则范例：

alert tcp any any → 192. 168. 1. 0/24 111 (content:"|00 01 86 a5|"; msg: "mountd access";)

该规则的含义：对来自任何地址的数据包，目的

廖光忠　硕士，讲师，研究方向：计算机网络及信息安全、多媒体、系统软件开发；陈志凤　硕士研究生，研究方向：信息安全、软件工程。

地为111端口，且数据包中包含字符串“|00 01 86 a5|”的流产生报警，报警消息为：mountd access。

括号前的部分是规则头，括号内的部分是规则选项。规则选项部分中冒号前的单词称为选项关键字。不是所有规则都必须包含规则选项部分，选项部分只是为了使对要收集或报警，或丢弃的包的定义更加严格。组成一个规则的所有元素对于指定的要采取的行动都必须是真的。当多个元素放在一起时，可以认为它们组成了一个逻辑与(AND)语句。同时，Snort规则库文件中的不同规则可以认为组成了一个大的逻辑或(OR)语句。

规则的头包含了定义一个包的who，where和what信息，以及当满足规则定义的所有属性的包出现时要采取的行动。规则的第一项是“规则动作”(rule action)，“规则动作”告诉Snort在发现匹配规则的包时要干什么。在Snort中有五种动作：alert、log、pass、activate和dynamic。

(1)Alert-使用选择的报警方法生成一个警报，然后记录(log)这个包。

(2)Log-记录这个包。

(3)Pass-丢弃(忽略)这个包。

(4)activate-报警并且激活另一条dynamic规则。

(5)dynamic-保持空闲直到被一条activate规则激活，被激活后就作为一条log规则执行。

你可以定义你自己的规则类型并且附加一条或者更多的输出模块给它，然后你就可以使用这些规则类型作为Snort规则的一个动作。

下面语句创建一条规则，记录到系统日志和MySQL数据库

```
ruletype redalert
{
type alert output
alert_syslog: LOG_AUTH LOG_ALERT
output database: log, mysql, user=snort db-
name=snort host=localhost
}
```

关于规则的详细说明参见相关文[1,2,5,7]。

Snort的不足：

Snort的安装十分困难，另外Snort依赖libpcap进行捕包，而libpcap在网络流量超出百兆负荷时它就无法正常运行，容易丢包，从而可能使Snort出现漏报，降低性能。分布式入侵检测采用多传感器方案，一定程度上可以缓解这种状况。同时它缺乏易于使用的图形用户界面的管理工具，没有通过寻呼或电子邮件传送报警的方法，呈现出紊乱的没有经过组织的警报信息显示方式。而这个分布式系统充分体现了图像用户界面下各种配置的易于操作和管理。

3 分布式入侵检测系统的设计与实现

本系统采用典型的三层体系，如图2所示。

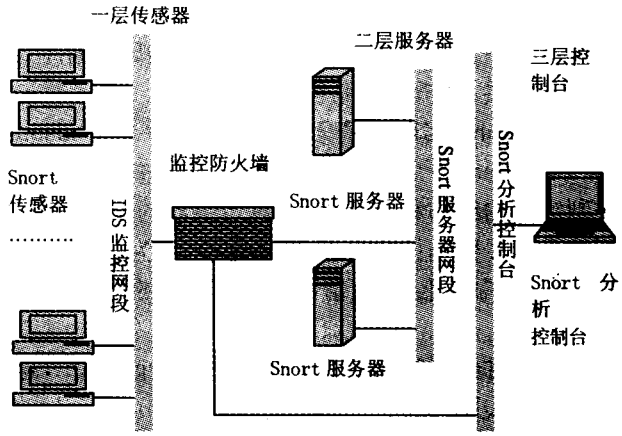

图2 Snort三层体系图

第一层：传感器层

它对经过的流量进行监控，抓包并将数据包传给第二层，传感器必须被放置在要对其进行监控入侵的网段，为了安全起见，传感器上除了安装相应的Snort支撑应用程序之外，其他无关的应用程序都不要安装(避免漏洞攻击，造成传感器工作不正常甚至瘫痪)。传感器上配置两块网卡：一块用来捕获数据包，设为混杂模式使用无IP地址的网卡进行监听以保证网络入侵检测系统自身的安全；另一块作为管理接口，与服务器通讯。Snort应用程序正是安装在传感器上，传感器通过捕获数据包，从受控网段收集数据，然后将包直接送给Snort应用程序，由Snort检测引擎对数据包进行分析检测，然后将报警数据发送给Snort服务器，通过另一块网卡接入内网，并为其分配内网所使用的私有IP地址，以便从内网访问分析控制台程序ACID。通过启用Apache服务器的用户身份验证和访问控制机制并结合SSL以保证系统的访问安全。

第二层：服务器层

从传感器收集报警数据并将其转换成用户可读的形式，并导入一个关系数据库MySQL进行组织管理。Snort本身并不需要关系数据库，它可以用其它方式记录报警，比如syslog，但为了方便且高效，我还是选择将报警数据存入关系数据库MySQL。Snort服务器的主要作用是完成对入侵数据的收集和分类。当服务器接收到传感器发送的警报后，就将这些警报存储到入侵数据库，进一步组织和管理。本文讨论的Snort服务器是基于Linux平台的，选择Linux是由于它出色的稳定性，安全性，并且是开源操作平台。服务器组件的安装：想要配置一个完善的Snort服务器，就必须要安装一些

Snort 服务器组件。这里的入侵数据库是 MySQL：目前使用最广泛的开放源代码的 SQL 数据库软件；MySQL 是一个多线程的 SQL 服务器，具有快速，易用等优点。分析入侵数据的工作由 ACID 完成，ACID 是一个基于 PHP 的分析引擎，这里主要用来处理由 IDS 产生的报警数据库即 MySQL 数据库。

第三层：分析员控制台

是用 php 实现的基于 ACID(Analysis Console for Databases)的管理组件。控制台的主要目的对控制消息的编码和发送，以及对传感器的规则配置，不涉及入侵数据的实际检测或管理 。为了可以远端浏览、控制入侵检测情况，我们需要的重要功能有：

(1)数据库的支持：MySQL 或 PostgreSQL，这里用 MySQL。

(2)图形库 GD 的支持：用于产生图形的图形库。

(3)套结字支持：执行本地的 whois 查询。

ACID 本质上是一些 PHP 脚本，提供 Web 浏览器和存储报警数据的数据库之间的接口。由于 ACID 脚本是用 PHP 语言编写的，因此需要让 Web 服务器支持 PHP4，因此在用源代码构建的时候，需要使用下列配置选项：

./configure [config option] ——with-mysql——with-gd——enable-sockets

ACID 控制台还提供强大的搜索功能，用户可根据时间、IP 地址、端口号、协议类型以及数据净荷(payload)等多种条件的灵活组合在入侵事件数据库中进行查询，以帮助网管人员进行分析；同时根据入侵情况实时设置相关规则，发送控制消息配置传感器。

控制台所发送的控制消息如果以明文方式发送，将非常危险，这里采用 ssl 对传输数据进行加密。SSL 协议(安全套接字协议)是为了弥补一些已被广泛应用的明文协议(如 www)的安全缺陷而设计的加密通讯协议，它可以用服务器的安全证书对通讯内容进行加密，以防止黑客中途截听通讯内容。

对传感器规则的配置主要思想：

1)中央警告及规则数据库存储整个系统的规则，当需要修改某个传感器的规则时，找到其在主控制台上注册的编号就可以对其规则进行增删。

2)主控台记录注册传感器规则的 Mysql 库表元 rules (sid、sensor，aler-type，protocol，route，msg)。控制台对传感器规则的添加和删除是通过 rules 元组中的 sensor 标记位决定的。也就是说，在数据库中并不是根据每个注册的 Sensor 存储规则，而是对某个特定规则而言，用 32b 标志位来标记规则所属的注册主机，比特位由低到高表示注册主机的先后顺序，这样设计的优点可避免同一条规则在主控台数据库中重复存储，避免数据冗余。

3)通过 ACID 地图表上清晰看出当前网络入侵状态，比如触发安全规则的网络流量中各种协议所占的比例、警报的数量、入侵主机和目标主机的 IP 地址及端口号等，可针对不同情况实时调整传感器规则。

总结与展望 论文实现的分布式入侵检测系统由中央控制台负责管理各个分布的传感器协同工作。分布式系统采用的三层体系，有利于充分利用处理器资源，实现实时性；采用分布式入侵检测系统将多个传感器接入不同监控网段，可以适当地解决 Snort 在高速网络上出现丢包的情况，在大型企业网络中也能胜任；由于各个传感器具有独立分析的功能，就不会形成瓶颈，一个传感器的失效(由于收到攻击或其他原因导致瘫痪)，不会导致整个系统的瘫痪，系统所受的影响只是局部入侵检测性能的降低。由于 IDS 系统本身容易收到攻击，我们尽量关闭不需要的服务，并使用防火墙保护使用的服务，尽可能使用加密和公匙进行身份认证，及时给所使用的应用程序打上最新补丁。

分布式入侵检测虽然可以解决一些现实问题，但是这个系统相当依赖于人的主观意识，一旦有入侵事件，也需要有人在控制台现场进行判断(而且判断不可能 100%准确)，然后做出对应的决策，这就不可避免地存在一个延时，有时可能是致命的。因此我们今后的研究方向应该是在 DIDS 中尽量摒弃人的因素，由系统根据入侵事件自动做出响应。比如当发现某一 IP 地址触发报警时，立即阻塞来自该 IP 的所有流量。当然这需要相当的智能，因为不恰当地阻塞流量可能引起系统无法正常提供服务，这一点也可能被攻击者利用。

参 考 文 献

1 Koziol J. Intrusion Detection with Snort [M]

2 Caswell B, Beale J, Foster J C, Posluns J. Snort 2.0 Intrusion Detection [M]

3 吴玉. 构建基于 Snort 的入侵检测系统. 微电子学与计算机，2005，22(7)

4 潘军，李祥和. 入侵检测系统 snort 的不足与改进. 计算机安全，2004(1)

5 韩腊萍，余雪丽. 一个分布式入侵检测系统框架设计. 计算机工程，2004，30(13)

6 李晓芳，姚远. 入侵检测工具 Snort 的研究与使用. 计算机应用与软件，2006，26(3)