Snor

本词条缺少**信息栏**，补充相关内容使词条更完整，还能快速升级，赶紧来编辑吧！

在1998年，Martin Roesch先生用C语言开发了[开放源代码](http://baike.baidu.com/view/1708.htm)(Open Source)的[入侵检测系统](http://baike.baidu.com/view/20936.htm)Snort.直至今天，Snort已发展成为一个多平台(Multi-Platform),实时(Real-Time)流量分析，网络IP数据包(Pocket)记录等特性的强大的网络入侵检测/防御系统(Network Intrusion Detection/Prevention System),即NIDS/NIPS.Snort符合通用公共许可(GPL——GNU General Pubic License),在网上可以通过免费下载获得Snort,并且只需要几分钟就可以安装并开始使用它。snort基于libpcap。

目录

1[简介](http://baike.baidu.com/link?url=0OVpzpiB7_s_moyFbtFZ1C2UjBzAKZLQOoL0MYGVBvvBNlTEv9P3cwvdJ-BMbrRS_PR5PYLGARoJgiiGxg7hl_" \l "1)

2[原理](http://baike.baidu.com/link?url=0OVpzpiB7_s_moyFbtFZ1C2UjBzAKZLQOoL0MYGVBvvBNlTEv9P3cwvdJ-BMbrRS_PR5PYLGARoJgiiGxg7hl_#2)

3[工作过程](http://baike.baidu.com/link?url=0OVpzpiB7_s_moyFbtFZ1C2UjBzAKZLQOoL0MYGVBvvBNlTEv9P3cwvdJ-BMbrRS_PR5PYLGARoJgiiGxg7hl_#3)

4[部署](http://baike.baidu.com/link?url=0OVpzpiB7_s_moyFbtFZ1C2UjBzAKZLQOoL0MYGVBvvBNlTEv9P3cwvdJ-BMbrRS_PR5PYLGARoJgiiGxg7hl_#4)

5[运行](http://baike.baidu.com/link?url=0OVpzpiB7_s_moyFbtFZ1C2UjBzAKZLQOoL0MYGVBvvBNlTEv9P3cwvdJ-BMbrRS_PR5PYLGARoJgiiGxg7hl_#5)

6[不足](http://baike.baidu.com/link?url=0OVpzpiB7_s_moyFbtFZ1C2UjBzAKZLQOoL0MYGVBvvBNlTEv9P3cwvdJ-BMbrRS_PR5PYLGARoJgiiGxg7hl_#6)

7[相关条目](http://baike.baidu.com/link?url=0OVpzpiB7_s_moyFbtFZ1C2UjBzAKZLQOoL0MYGVBvvBNlTEv9P3cwvdJ-BMbrRS_PR5PYLGARoJgiiGxg7hl_#7)

8[封包记录](http://baike.baidu.com/link?url=0OVpzpiB7_s_moyFbtFZ1C2UjBzAKZLQOoL0MYGVBvvBNlTEv9P3cwvdJ-BMbrRS_PR5PYLGARoJgiiGxg7hl_#8)

9[入侵侦测](http://baike.baidu.com/link?url=0OVpzpiB7_s_moyFbtFZ1C2UjBzAKZLQOoL0MYGVBvvBNlTEv9P3cwvdJ-BMbrRS_PR5PYLGARoJgiiGxg7hl_#9)

10[主要指令](http://baike.baidu.com/link?url=0OVpzpiB7_s_moyFbtFZ1C2UjBzAKZLQOoL0MYGVBvvBNlTEv9P3cwvdJ-BMbrRS_PR5PYLGARoJgiiGxg7hl_#10)

1简介[编辑](http://baike.baidu.com/link?url=0OVpzpiB7_s_moyFbtFZ1C2UjBzAKZLQOoL0MYGVBvvBNlTEv9P3cwvdJ-BMbrRS_PR5PYLGARoJgiiGxg7hl_)

Snort有[三种](http://baike.baidu.com/view/7116015.htm)工作模式：[嗅探器](http://baike.baidu.com/view/68282.htm)、[数据包](http://baike.baidu.com/view/25880.htm)记录器、网络[入侵检测系统](http://baike.baidu.com/view/20936.htm)。[嗅探器](http://baike.baidu.com/view/68282.htm)模式仅仅是从网络上读取[数据包](http://baike.baidu.com/view/25880.htm)并作为连续不断的流显示在[终](http://baike.baidu.com/view/698000.htm)端上。[数据包](http://baike.baidu.com/view/25880.htm)记录器模式把数据包记录到硬盘上。网络[入侵检测](http://baike.baidu.com/view/16487.htm)模式是最[复杂](http://baike.baidu.com/view/399895.htm)的，而且是可配置的。我们可以让snort分析网络[数据](http://baike.baidu.com/view/38752.htm)流以匹配用户[定义](http://baike.baidu.com/view/25538.htm)的一些规则，并根据检测结果采取一定的动作。

2原理[编辑](http://baike.baidu.com/link?url=0OVpzpiB7_s_moyFbtFZ1C2UjBzAKZLQOoL0MYGVBvvBNlTEv9P3cwvdJ-BMbrRS_PR5PYLGARoJgiiGxg7hl_)

Snort能够对网络上的数据包进行抓包分析，但区别于其它嗅探器的是，它能根据所定义的规则进行响应及处理。Snort 通过对获取的数据包，进行各规则的分析后，根据规则链，可采取Activation（报警并启动另外一个动态规则链）、Dynamic（由其它的规则包调用）、Alert（报警），Pass（忽略），Log（不报警但记录网络流量）五种响应的机制。   
　　Snort有数据包嗅探，数据包分析，数据包检测，响应处理等多种功能，每个模块实现不同的功能，各模块都是用插件的方式和Snort相结合，功能扩展方便。例如，预处理插件的功能就是在规则匹配误用检测之前运行，完成TIP碎片重组，http解码，telnet解码等功能，处理插件完成检查协议各字段，关闭连接，攻击响应等功能，输出插件将得理后的各种情况以日志或警告的方式输出。

3工作过程[编辑](http://baike.baidu.com/link?url=0OVpzpiB7_s_moyFbtFZ1C2UjBzAKZLQOoL0MYGVBvvBNlTEv9P3cwvdJ-BMbrRS_PR5PYLGARoJgiiGxg7hl_)

Snort通过在网络TCP/IP的5层结构的数据链路层进行抓取网络数据包，抓包时需将网卡设置为混杂模式，根据操作系统的不同采用libpcap或winpcap函数从网络中捕获数据包；然后将捕获的数据包送到包解码器进行解码。网络中的数据包有可能是以太网包、令牌环包、TCP/IP包、802.11包等格式。在这一过程包解码器将其解码成Snort认识的统一的格式；之后就将数据包送到预处理器进行处理，预处理包括能分片的数据包进行重新组装，处理一些明显的错误等问题。预处理的过程主要是通过插件来完成，比如Http预处理器完成对Http请求解码的规格化，Frag2事务处理器完成数据包的组装，Stream4预处理器用来使Snort状态化，端口扫描预处理器能检测端口扫描的能力等；对数据包进行了解码，过滤，预处理后，进入了Snort的最重要一环，进行规则的建立及根据规则进行检测。规则检测是Snort中最重要的部分，作用是检测数据包中是否包含有入侵行为。例如规则alert tcp any any ->202.12.1.0/24 80（msg：”misc large tcp packet”；dsize：>3000；）这条规则的意思是，当一个流入202.12.1.0这个网段的TCP包长度超过3000B时就发出警报。规则语法涉及到协议的类型、内容、长度、报头等各种要素。处理规则文件的时候，用三维链表来存规则信息以便和后面的数据包进行匹配，三维链表一旦构建好了，就通过某种方法查找三维链表并进行匹配和发生响应。规则检测的处理能力需要根据规则的数量，运行Snort机器的性能，网络负载等因素决定；最后一步就是输出模块，经过检测后的数据包需要以各种形式将结果进行输出，输出形式可以是输出到alert文件、其它日志文件、数据库UNIX域或Socket等。

4部署[编辑](http://baike.baidu.com/link?url=0OVpzpiB7_s_moyFbtFZ1C2UjBzAKZLQOoL0MYGVBvvBNlTEv9P3cwvdJ-BMbrRS_PR5PYLGARoJgiiGxg7hl_)

Snort的部署非常灵活，很多操作系统上都可以运行，可以运行在window xp，windows2003，linux等操作系统上。用户在操作系统平台选择上应考虑其安全性，稳定性，同时还要考虑与其它应用程序的协同工作的要求。如果入侵检测系统本身都不稳定容易受到攻击，就不能很好的去检测其它安全攻击漏洞了。在Linux与Windows操作系统相比较之下，Linux更加健壮，安全和稳定。Snort的运行，主要是通过各插件协同工作才使其功能强大，所以在部署时选择合适的数据库，Web服务器，图形处理程序软件及版本也非常重要。Snort部署时一般是由传感器层、服务器层、管理员控制台层三层结构组成。传感器层层就是一个网络数据包的嗅探器层，收集网络数据包交给服务器层进行处理，管理员控制台层则主要是显示检测分析结果。部署Snort时可根据企业网络规模的大小，采用三层结构分别部署或采用三层结构集成在一台机器上进行部署，也可采用服务器层与控制台集成的两层结构。

5运行[编辑](http://baike.baidu.com/link?url=0OVpzpiB7_s_moyFbtFZ1C2UjBzAKZLQOoL0MYGVBvvBNlTEv9P3cwvdJ-BMbrRS_PR5PYLGARoJgiiGxg7hl_)

Snort的有三种模式的运行方式：嗅探器模式，包记录器模式，和网络入侵检测系统模式。嗅探器模式仅仅是从捕获网络数据包显示在终端上，包记录器模式则是把捕获的数据包存储到磁盘，入侵检测模式则是最复杂的能对数据包进行分析、按规则进行检测、做出响应。

6不足[编辑](http://baike.baidu.com/link?url=0OVpzpiB7_s_moyFbtFZ1C2UjBzAKZLQOoL0MYGVBvvBNlTEv9P3cwvdJ-BMbrRS_PR5PYLGARoJgiiGxg7hl_)

Snort入侵检测系统适应多种平台，源代码开放，使用免费，受众多用户喜爱，但也有不少缺点。Snort之所以说他是轻量型就是说他的功能还不够完善，比如与其它产品产生联动等方面还有待改进；Snort由各功能插件协同工作，安装复杂，各软件插件有时会因版本等问题影响程序运行；Snort对所有流量的数据根据规则进行匹配，有时会产生很多合法程序的误报。

**Snort企业部署实战**

**1 背景**

      我们知道企业网络目前威胁来自两个位置：一个是内部，一个是外部。来自外部的威胁都能被防火墙所阻止，但内部攻击都不好防范。因为公司内部人员对系统了解很深且有合法访问权限，所以内部攻击更容易成功。IDS为信息提供保护，已经成为深度防御策略中的重要部分。IDS与现实世界里的防窃报警装置类似。他们都对入侵进行监控，当发现可疑行为时，就向特定的当事人发出警报。IDS分为两类：主机IDS(HIDS)和网络IDS(NIDS)。HIDS安装在受监控主机上，拥有对敏感文件的访问特权。HIDS利用这一访问特权对异常行为进行监控。NIDS存在于网络中，通过捕获发往其他主机的流量来保护大量网络设施。HIDS和NIDS都有各自的优点和缺点，完整的安全解决方案应包括这两种IDS，对于这一点比较难做到。不了解这一领域的人常常认为IDS就像一把\*\*\*，能解决所有安全问题。例如有的单位花了大笔的钱购置了商业IDS由于配置不当反而搞得练练误报一下子就把数据库塞满了，大量丢包，进而崩溃。这种态度使人们以为只要将IDS随便安放在网络中就万事大吉了，不必担心任何问题，实际上远非如此。没有人会认为Email服务器直接连在Internet上就能正确运作。同样，你也需要正确的计划IDS策略，传感器的放置。下文以开源软件Snort的安装与维护为例，介绍正确地安装和维护IDS的思路和方法。

**2 安装Snort**

**2.1安装准备工作**

我们在安装前我们要知道我们需要监控的内容，理想的状况是对一切进行监控。所有网络设备和任何从外部到企业的连接都处在Snort的监视之下。尽管这一计划对小公司只有几十台机器是很可能实现的，但是当大型企业中连接上天台网络设备时，这成了难以施展的艰巨任务。为了加强snort检测的安全性，最好能为监控网段提供独立的智能交换机，如果你需要配置分布式的配置，可以吧服务器和控制台接在一个交换机上，二其他传感器放置在不同的物理位置，但这样的成本会有所增加。Snort IDS的维护问题是无法回避的。你迟早要对Snort特征更新并编写定制的规则，所以你还需要一个懂得维护IDS的专业人士。

**2.2深入Snort**

Snort包含很多可配置的内部组件，它们对误报、漏报以及抓包和记录日志等的性能都有很大影响。能深入了解Snort的内幕有助于有效地利用Snort监控人侵。还会帮助你根据自己的网络定制Snort，并且避免它的一些常见缺陷。

**2.2.1用Libpcap输送Snort包**

Snort没有自己的捕包工具，它需要一个外部的捕包程序库：libpcap。Snort利用libpcap独立地从物理链路上进行捕包，它可以借助libpcap的平台为一个真正的与平台无关的应用程序。直接从网卡捕包的任务由libpcap承担。这一捕获原始包的工具是由底层操作系统提供给其他应用程序使用的。Snort需要数据保持原始状态，它利用的就是原始包所有的协议头信息都保持完整，未被操作系统更改的特性来检测某些形式的攻击。由于利用libpcap获取原始包，一次只能处理一个包，这不是最好的方法，这也制约了它对千兆网络进行监控的瓶颈。

**2.2.2包解码器**

包一被收集到Snort必须对每一个具体的协议元素进行解码。在包通过各种协议的解码器时，解码后的包数据将堆满一个数据结构。包数据一被存人数据结构中，就会迅速被送到预处理程序和检测引擎进行分析。

**2.2.3预处理程序**

Snort的预处理分为两类。它们可以用来针对可疑行为检查包或者修改包以便检测引擎能对其正确解释。预处理的参数可以通过snort.Conf配置文件调整。

预处理器：

 Frag2

 Stream4

 Stream4\_reassemble

 Http\_decode

 RPC\_decode

 BO

 Telnet\_decode

 ARPspoof

 ASNI\_decode

 Fnord

 Conversation

 Portscan2

 SPADE

**2.2.4检测引擎**

检测引擎将流量与规则按其载人内存的顺序依次进行匹配。是Snort的一个主要部件。

**2.2.5输出插件**

Snort的输出插件接收Snort传来的入侵数据。输出插件的目的是将报警数据转储到另一种资源或文件中。

**2.2.6 Snort的性能问题**

Snort有效工作的性能可能会受到以下几种选择的限制：硬件、操作系统和连网的组件。

对snort的性能影响最大的是snort的配置设定以及规则集设置。内部瓶颈则主要出现在包解码阶段，要snort检查包的容，那么它比一般的规则都要更加耗费系统资源。启用的检查包内容的规则越多，snort的运行就需要越多的系统资源。如果要激活预处理程序中的某些设置选项，就会需要消耗额外的系统资源。最明显的例子就是启用在frag2预处理程序和stream4预处理程序中的“最大存储容量(memcap)”选项。如果您打算激活大量耗费资源的预处理程序选项，最好确定有足够的硬件资源的支持。我曾经遇到过一个用户花了大笔的钱购买了最先进的 IDS由于配置不当，连检测100M网都出现了丢包现象。

**1）SPAN端口监控**

在监控时我们必然需要做SPAN，SPAN端口监控是另外一种在现有网络结构中引入监控网段的方法。Cisco交换机的中高端产品都有SPAN端口或镜像端口。Span端口既可以是一个专用端口，也可以通过该端口实现交换机上所有的端口的配置选项设定。利用SPAN端口的特点实现监控功能是一种实用的方法。使用SPAN端口监控法并不会给所要监控的网络引入单点错误的问题。与网内Hub监控法相比，这是使用SPAN端口监控最大的优点。

注意：镜像顺序问题：当所监控的网络要升级为高带宽网络时，可以先是只镜像一个端口，对snort的性能观察一段时间，并根据需要进行调整。当snort的这个端口调整好了之后，可以切合实际的、循序渐进的增加别的端口，要注意的是，千万不能一下子增加过多的端口。用SPAN端口监控法将会降低本生交换设备的性能用SPAN端口会使交换设备的内存负担过重，从而使设备的性能下降。对流量的映是一个非常耗费内存的过程，

**2.2.7.安装Snort**

操作系统：Red Hat Enterprise Linux 5.5

数据库：MySQL：mysql-5.1

Web服务器：Apache：httpd-2.2

WEB语言：PHP：php-5.4

首先我们需要安装MySQL 、Apache（必须安装**mod\_ssl模块**） 、PHP、并进行配置Apache，其详细安装过程可以参见《Linux企业应用案例精解》

**1）安装主程序**

#tar zxf snort-2.8.5.2.tar.gz

#cd snort-2.8.5.2

#./configure --with-mysql=/usr/local/mysql & make & make install

创建配置文件目录mkdir /etc/snort

创建日志目录mkdir /var/log/snort

**2）安装snort规则**

tar zxf snortrules-snapshot-2860.tar.gz

tar zxf snortrules-snapshot-CURRENT.tar.gz

mv rules/ /etc/snort

cp \* /etc/snort/

修改/etc/snort/snort.conf文件

监听的本地网段var HOME\_NET 192.168.150.0/24

有五行以output database: 开头的行，将其“#”号去掉。

**3） 创建snort数据库**

mysql> create database snort;

mysql> connect snort;

mysql> source /usr/local/src/snort-2.8.5.2/schemas/create\_mysql;

mysql>grant CREATE,INSERT,SELECT,DELETE,UPDATE on snort.\* to snort;

mysql>grant CREATE,INSERT,SELECT,DELETE,UPDATE on snort.\* to snort@localhost;

另外有兴趣的读者可以尝试使用phpMyadmin这一工具，phpMyAdmin是一个基于web的MySQL数据库管理工具。它能够创建和删除数据库，创建/删除/修改表格,删除/编辑/新增字段，执行SQL脚本等。

**2.2.8启动snort**

正确安装并配置完成以后，下一步我们需要启动snort

#snort -c /etc/snort/snort.conf

为了snort安全应避免用root身份运行snort，这时需要创建专用的用户和组

#useradd snort ,如果是redhat在创建用户的同时就创建了snort组

#snort –u snort –g snort –U –d –D –c /etc/snort/snort.conf

接下来就需要安装Acid+Adodb+Jpgraph, ACID(Analysis Console for Incident Databases)是snort使用的标准分析员控制台软件。ACID是一个基于PHP的分析引擎，它能够搜索、处理snort产生的数据库。下面是安装及配置过程。这一过程也非常简单将adodb和jpgraph的tar包复制到Apache根目录下，解开acid包后，修改acid\_conf.php配置即可。注意Acid配置参数都在acid\_config.php文件里，所有的值都必须放在双引号内(“),而且后面要加上分号（;）必须现已SSL模式启动Apache，定位到ACID的主页https://IP地址/acid/，如图1所示。

[](http://img1.51cto.com/attachment/201208/100741933.jpg)

图1 ACID界面

**2.2.9提高性能**

如果是监控10/100M的网络还行，如果流量过大就需要提高snort的监控性能，目前最经济的方法是，在双网卡上运行Snort程序，可以配置Snort来侦听多个网卡，问题是Snort每个命令行选项(-i)只接受一个网卡。有种在多种网卡上运行Snort的方法：

 为每个网卡运行一个独立的Snort进程；

 通过绑定Linux内核的特征将所有的网卡绑定在一起。

用Snort监控多个网卡时选择哪种方法取决于你的环境和优先级等多种因素。运行多个Snort进程会增大工作量，并浪费大量的无法接受的处理器时间周期。如果你有可用的资源来运行两个或多个Snort进程，那么你应该考虑一下数据管理问题。假设所有的Snort实例以同样的方式配置，那么同样的攻击会被报告多次。这会令人侵检测系统管理员头疼，尤其是启用时报警的时候。当你面对不同的网卡有不同的入侵检测需求时，为每个网卡分配单个Snort程是最理想的。如果你为每个网卡都分配了一个独立的Snort进程，那么你就为每个网卡创建一个类似虚拟的传感器。在一个机器上架设几个“传感器”，你就可以为每个独立的Snort进程载入不同的配置、规则和输出插件。这最适合于独立的Snort进程。另一方面，如果你不能这样、或者不想为每个网卡启用额外的Snort进程，你可以将两个网卡绑定在一起。这样当你启用Snort时，就能用 -i命令选项指定个已被绑定的网卡(如bond0)。

为了实现这个目的，请编辑/etc/modules.conf，加入如下行：

alias bond0 bonding

现在，每次重启机器，你都需要在将IP地址信息分配给网卡之后输入下而的侖今桌渐活绑定的网卡：

ifconfig bondup

ifenslave bond0 eth0

ifenflave bond0 eth1

注意,你可将这些命令放在一个脚本里，在系统启动时运行该脚本。当运行Snort时，可以按如下方式使用的bond0网卡：

snort < options> -i bond0

**3 维护Snort**

当你安装好系统后就必然会对系统进行维护，或是对Snort做一些重要的改动以保持它的相关性，比如升级规则集，修改配置选项，最后升级Snort应用程序本身。如果你运行的是多个传感器构成的分布式系统，虽然这些手工方法也是可取的，但手工修改多个传感器就会变得相当困难，还容易出错。

这是我们需要管理助手SnortCenter,它是一款基于Web方式升级和维护Snort配置的管理应用软件。是一款用于远程管理Snort传感器的应用软件。它用的是PHP/MySQL Web界面，安装完成启动界面如图2所示。

[](http://img1.51cto.com/attachment/201208/100818424.jpg)

图2 SnortCenter 界面

特征：

l snort后台进程状态监视器；

l 远程snort停止/启动/重启；

l snortcenter用户的访问控制；

l 传感器组。

l ACID集成；

SnortCenter包括基于PHP的管理应用软件和SnortCenter代理。SnortCenter管理控制台安装在Snort服务器上，而SnortCenter传感器代理被安装在所管理的传感器中。SnortCenter增强了可能安装在分布式系统上的Snort，服务器端需要如下的软件包：

l MySQL;

l Apache;

l PHP;

l ADODB;

l OpenSSL;

l cURL

这里，除了cURL软件包，其他的软件包应该都是大家比较熟悉的，因为绝大多数的操作系统都包括这些软件包，SnortCenter管理控制台可运行在Windows、Linux和BSD系统上。SnortCenter传感器代理需要安装在基于UNIX操作系统的Perl上。该代理在一些附加的预编译程序帮助下可以运行在基于Windows的传感器上。

**3.1 SnortCenter的安装**

前提是要安装并配置好Acid，因此你应该预先在作为Snort服务器的机器上安装MySQL，Apache，PHP，ADODB和OpenSSL。你可以将SnortCenter安装在Linux系统上了。

**3.1.1 SnortCenter管理控制台**

在安装SnortCenter之前惟一还需要安装的软件包是cURL，这是一个不需要用户干涉通过URL传输文件的命令行工具，它用于管理和控制Snort传感器。你可以通过如下命令行检查在Red Hat上是否安装了该软件包：

Rpm -qa | grep curl

该命令行将会查询包含了curl字符串的软件包，如果你没有安装cURL，可以去网上下载。

下一步在WEB根目录下建立snortcenter目录，将下载文件包解压到这个目录里，然后就通过配置config.php文件来配置SnortCenter 。

对于这个配置文件需要说明的有以下几点：

l DBlib\_path设定Adodb库的位置。

l url\_path该变量应设为cURL可执行文件的位置。

l DBtype这里设置你所安装的数据库的类型

l DB\_dbname这是你在下一步中要创建的SnortCenter数据库名

l DB\_host DB\_host是Snort服务器的主机名。如果SnortCenter管理控制台和数据库安装在同一台计算机上，应将该参数设为localhost；

l DB\_user SnortCenter登录数据库所用的帐号。

l DB\_ password数据库—用户的密码；

l DB\_ port DB\_ port是数据库运行的端口号。

保存修改并关闭config.php，下一个任务是建立DB\_dbname变量指定的数据库，首先需要登录MysqL数据库,然后创建SnortCenter数据库，命令如下：

>create database snortcenter;

创建好数据库之后，在Web浏览器中就可以看到SnortCenter管理控制台（地址为https: //localhost/snortcenter）了。这里建立了SnortCenter需要的所有表。你也可以用位于tarball的snortcenter db.Mysql脚本创建它们。这就完成了SnortCenter管理控制台部分的安装。第一次登录时，你需修改用户名admin和口令。

**3.1.2安装SnortCenter传感器代理**

要完成SnortCenter的安装，还需在你想用SnortCenter管理的传感器上安装SnortCenter传感器代理。安装基于UNIX的代理需要Perl、OpenSSL和Perl模块Net::SSLeay。前面我们已经在传感器上安装了OpenSSL和Perl，现在只需要进行Net::SSLeay模块的安装。你可以在网址http://search.cpan.org下载该模块。

下载并安装Net::SSLeay，首先在源目录下执行下列命令：

Perl Makefile.pl

Make install

安装好Net::SSLeay模块后，需创建SnortCenter传感器代理所用的目录，即创建下列目录：

l 程序目录：/usr/local/snortcenter

l 配置目录：/usr/local/snortcenter/conf

l 日志目录：/usrAocal/snortcenter/log

l 策略目录：/usr/local/snortcenter/rules

接着，你还需为SnortCenter创建一个SSL证书。用下面的命令行创建它：

#openssl req -new -x509 -days 365 -nodes -out snortcenter.pem -keyout snortcenter. pem

将snortcenter.pem文件复制到/usr/local/snortcenter/conf目录下。现在你就可以准备安装SnortCenter传感器代理了，在http://users.pandora.be/larc/download/下载合适的版本。

将文件解压并移动到/usr/local/snortcenter/目录下。运行安装的shell脚本：

#./setup.sh

安装脚本会向你提出许多问题。你已经为Snort和SnortCenter创建所需的文件夹，当询问时依次输入这些目录。代理可以运行在任何端口上的，可以任意指定，但要记住你选择的是哪一个端口。指定SnortCeneter管理和侦听的网卡IP地址。当出现启用SSL选项时，选择Yes。你也应该注意记住代理的登录名和口令，在管理器控制台中输入认证信息。最后的选项是设置Snort服务器的IP地址。这样就完成了SnortCenter传感器代理的安装。重复这个安装过程，为你的Snort环境中的每个传感器安装代理。

注意：配置snortcenter,要想升级传感器的多种配置，必须首先在snortcenter管理控制台中添加他们。