Snort 入侵检测

Snort 入侵检测

最好的入侵检测系统（IDS）是免费的、开源的 Snort 工具。它拥有大量的用户，而且有商业公司 Sourcefire 的支持，使得 Snort 成为受到欢迎的入侵检测系统工具。这个工具本身是免费的。它所需要的是一些在上面运行的硬件以及安装、配置和维护的时间。Snort 可以在任何操作系统上运行，包括 Windows 和 Linux，但是有人认为它的操作很复杂。本专题的目的是揭示 Snort 的神秘性。

为什么关注 Snort

企业很难决定要不要实施入侵检测系统（IDS）。但是 IDS 要求认真的“照顾和喂养”。而且商务系统的价格也很高。尽管如此，还是存在企业级的开源 IDS，叫做 Snort。那么什么理由让 Snort 如此受关注呢 IDS 的网络适配器。

- Snort：为什么 IDS 值得关注？

Snort 的相关配置和设计

在决定使用 Snort 这种入侵检测系统后，如何进行相关的配置和网络设计呢？本部分中将介绍如何识别和监控网络端口、如何使用交换机和网络分段处理网络设计、IDS 网络传感器的位置、Snort IDS 传感器的操作系统以及

- Snort：入侵检测后如何识别和监控网络端口
- Snort：如何使用交换器和分段处理网络设计
- Snort：IDS 网络传感器应该配置在哪里？
- Snort：为 Snort IDS 传感器确定操作系统
- Snort：如何确定 IDS 的网络适配器

Snort 规则

一旦安装、配置了 Snort，并已经开始工作，要考虑的下一件事情是规则。Snort 规则定义了用于查找网络上潜在恶意流量的方式和标准。没有这些 IDS 规则，Snort 仅仅只是另一个嗅探器。Snort 规则的编写相对简单，因此用户可以自定义规则，另外 Snort.org 等也提供 Snort 的官方规则以及用户共享的规则。此外 Snort 的规则并不适合所有的企业，因此需要更新 Snort 规则

- Snort：修改和编写自定义 Snort 规则
- Snort：如何配置 Snort 变量
- Snort：在哪里查找 Snort IDS 规则
- Snort：如何自动更新 Snort 规则
- Snort：如何破译 Snort VRT 规则的 Oinkcode

Snort 测试

新的 Snort 系统的运行是不是太安静了？不论你是使用新的 Snort，还是在新的平台上配置的——你所担心的低噪音等级。它可能会是严谨的调制（或者过分调制）的系统，或者你可能在一个平静的网段上拥有 IDS，还好还有一些在线测试 Snort 的方法，确保它在你的环境中正常运行。

- Snort：使用 IDS 规则测试 Snort

Snort：为什么 IDS 值得关注？

企业很难决定要不要实施入侵检测系统（IDS）。但是 IDS 要求认真的“照顾和喂养”。而且商务系统的价格也很高。尽管如此，还是存在企业级的开源 IDS，叫做 Snort。而它目前的已经处于“不能失去”的至高的位置。

Snort 工具是免费的，而且可以在现在的任何操作系统以及你所有的陈旧的硬件上运行。在 Snort IDS 上的真正投入是时间和汗水，但是我保证在不到一个星期的时间内，你就可以了解网络中你从来不知道的内容。三个月后，你已经发现了一定数量的网络问题和你不曾知道你有的入侵检测。例如，你可能发现一个错误配置管理工作区，而它在一切上面都使用 SNMP 社区字符串“public”。我们都知道这样不好，对吧？

除了可以了解网络，还可以了解到有趣的恶意代码。有了 Snort，规则就变得强大、灵活，也容易编写了，所以删除最新的恶意软件的最新规则通常是由 Snort 社区的人在发作的其间的几个小时内编写的。在你的本地或实验规则文件中增加一条规则，重启 Snort，然后你就可以进行检测、包含并删除任何可以绕过其他安全层的不稳定因素。

除了可以从 Snort.org 和 Bleedingsnort.com 网站上获得新的 Snort 规则，你也可以自己编写。可能有需要标记的模糊的应用或者协议，你想要使用 Snort 实施基于策略的 IDS。基于策略在某些特定的环境中运行的强大概念。在这个环境中，你可以确定所有已知和被允许的流量，然后会对其它的任何东西发出警告。尽管如此，确定已知和被允许的流量在所有的环境中非常不容易，但是在简单或者被严密控制的环境中要容易一些，所以基于策略的 IDS 并不适用于每个人。

关于 IDS 规则中最易被忽略的是他们都是开源的（特定许可可参见 Snorg.org 和 Bleedingsnort.com 网站）。讽刺的是，ISS RealSecure 和 Symantec ManHunt 都存在允许使用 Snort 规则的模块。但是真正的美妙之处在与 Snort 规则是可读的，而在厂商提供

的就不可以。你能获得的最好结果是某些人写的片段，而且是不相同的。对实际规则的访问可以帮你在精确的标准上做出精明的决定，而这些标准可以引发警报和对环境的相关性。

不管你安装 Snort 的需求或理由是什么，我推荐你阅读 Snort.org.上的常见问题解答并加入邮件列表（特别是 Snort 用户）。

（作者：JP Vossen 译者：Tina Guo 来源：TechTarget 中国）

Snort：入侵检测后如何识别和监控网络端口

在分析防火墙日志或者 IDS 警告的时候，可能会碰到不熟悉的资源或者目的端口。分析过程的下一步是确定哪些服务正在使用哪些端口，这样就能决定网络是否存在风险。

识别并开始监控网络端口的最简单的方法是查看每个现代的 TCP/IP 堆栈的服务文件。就是在 Windows 系统下的 C:\WINDOWS\SYSTEM32\DRIVERS\ETC\SERVICES（提示：可以使用记事本来查看或编辑文件——只需双击它，然后从列表中选择记事本），或者大部分 Unix 版本下的/etc/services。Windows 的“find”或者 Unix 的“grep”可以快速搜索到这些文件。通常在默认服务文件中找不到端口，因为他们只列出可以得到的网络端口和服务的很少的子集。然后就该使用 Web 了：

互联网端口数据库是带有 Web 和 DNS 界面的网络端口的大型数据库，和一个可以用于替换或者补充主要的服务文件的刻下在 uber-services 文件。这个站点从来都没让我失望过。

SANS Institute 有大量的安全信息，包括（有点儿陈旧的）一般特洛伊使用的端口列表。

DoSHelp.com 有相似的列表。

Nmap 的列表包含 2200 个端口。

Internet Assigned Numbers Authority (IANA) listing 中提供了指定端口数量的确定资源。这里的关键词是“指定”。破解者不能为他们的恶意项目记录端口数，而且用户可以为了安全，通过不明的物质或者防火墙把服务移动到不同的端口。

一旦你发现了使用可疑端口的服务，不要做任何假设。首先。它真的是看起来那样吗？还是有人改变了端口数？有些端口通常用于不止一种服务，那么哪个是它呢？这种服务是环境允许的吗？应该有它吗？下面的工作可以帮助发现到底发生了什么。

Foundstone 有一个叫做 fpor 的命令行功能，而 SysInternals 有一个叫做 TCPView 的 GUI 项目。这两个工具可以 向你显示你的 Window 电脑上打开的 Tcp 和 UDP 端口，以及使用它们的项目和程序。这些项目出现在简单的 ZIP 文件中，你可以解压、使用，然后删除它们——不需要安装。

在 Unix 上，只要使用‘netstat -anp | less’或者更好一点，使用‘lsof -Pni’。lsof (LiSt Open Files)和 Linux 分布在一起，虽然它通常不是默认安装的。和名字显示的一样，它可以把打开的文件列表，但是因为 Unix 中的所有都是文件，这个工具要做的比名字中显示的多得多。我高度推荐使用它。

Nmap 最近增加了服务和网络端口扫描器。还有些其他的工作也可以作这些工作，但是 Nmap 可能是最好最简的。它还在 Winsows 上运行。和通常一样，在你编写可以这么做的许可前，一点不要端口扫面任何东西。

如果其他的都失败了，试一试 Google 搜索，但是不要对你的发现作过多的设想。目的是为了验证你的环境中发生了什么——为什么你收到了警告，为什么这个日志会产生，它是恶性的还是良性的。你要比 Web 上的任何人都了解你的网络。

(作者：JP Vossen 译者：Tina Guo 来源：TechTarget 中国)

Snort：如何使用交换器和分段处理网络设计

你已经决定要配置 Snort，这种网络入侵检测系统（IDS），而且已经理解了它基本上是一个嗅探器。如何监控不同的网段，特别是当使用网络交换机或者 VLAN（虚拟局域网）时？答案当然是“看情况而定”。

在确定了预定、选好了 IDS 产品后——例如 Snort——你需要确定你需要多少传感器，可以负担多少。在决定需要多少传感器前，必须理解 Snort，或者其它的 IDS 只能监控他们看到的。在核心路由器和网络中的旧时代，这项任务相对简单_你可以购买你能负担得起的尽可能多的入侵检测系统，并根据风险等级和重要性在每个网段都配置一个 IDS。

网络交换机，不像其它的网络中心，并不向网段上的每个端口发送所有的流量。基本上有三种方法。第一个是回头时在战略中心使用网络中心，有时不太赞成，因为它可以减少带宽，并增加中等重要程度的服务网络（又叫做 DMZ），它可能有意义，与 inweita 不贵、非常简单而且有用。

尽管使用网络交换机，查看所有流量的第二个方法是使用具有 port spanning 或者 port mirroring 功能的智能或者管理交换机，不需要说，这些网络交换机的成本很高，但是他们已经在所有最基本和有意耗费成本的环境中使用了。参考厂商的文件或者 Web 资料，找到如何在摸个硬件上创建 mirror ports 或者 span ports 的详细指南。这里有启发作用的两个思科的指南：

Cisco Catalyst 4000 Series Switches

Cisco Catalyst 6500 Series Switches

每个 VLAN 都需要一个 span port。每个交换机都限制了很少的 span port（有几个原因，其中之一是带宽），所以当设计覆盖范围的时候要考虑这一点。有时，交换机厂商的入侵检测系统可以客服这些限制（例如 Cisco 安全入侵检测系统刀片），其他需要记住的是 span port 通常是只读的，他们通常不参与生成树（spanning tree）。（应该和厂商确认。）

检测（tap）流量的最后一个方法是使用网络监测分路设备(tap)。 有几家公司生产 CAT-5 和光纤的电缆 tap(又叫做网络 tap)。Tap 的价格在几百美元或者更高。他们很容易安装，但是让他们和 IDS 传感器一起工作很难。发送和接收通常被分在两个电缆中，所以传感器需要两个网卡。

不能理解嗅探和交换机和网段如何工作是第一次安装 IDS 遇到的最常见的问题之一。如果你的 IDS 查看不到网络流量，或者只有广播或者流出/流向自己的单向流量，几乎可以确定存在交换/生成端口问题。根据 IDS 传感器的情况，通常可以从设备上运行 tcpdump 或者 windump，校验流量。如果 IDS 传感器上的应用或者其他不能这么做，在连接了相同的交换端口的笔记本上使用上面的工具、Ethereal 或者另外的嗅探器作为 IDS。

(作者：JP Vossen 译者：Tina Guo 来源：TechTarget 中国)

Snort：IDS 网络传感器应该配置在哪里？

在决定安装 Snort 等 IDS 设备、确定了预算、选定了产品并理解了 NIDSes（网络入侵检测系统）如何与网络架构（特别是网络交换机和 VLAN），就需要确定把 IDS 传感器放置在哪里。IDS 传感器实际上是“嗅探”网络并监控网络流量的。为了做出决定，需要回答三个问题：有什么风险、想要监控和保护什么、以及流量如何在环境中流动。

在《Snort：如何使用交换器和分段处理网络设计》中提到，入侵检测系统和网络交换机不是在一起的，所以需要理解你需要保护的流向和流出的资源的流来能够式如何工作的，这样可以确定监控的最好位置。放置 IDS 传感器的最明显的位置是靠近防火墙的地方。但是，是把网络传感器放在防火墙里面、外面、还是里外都放呢？如果你想要查看在连接到互联网时带进来了什么样的恶意邻居，那么一定要把 IDS 传感器放置在防火墙外面。确保它经过了恰当的加固。如果想要关注对边界内部潜在的恶意流量，那么就从内部监控。

因为一些合法的政策、预算和研究的因素，需要查看对连接的“攻击”，但是如果已经留意并反馈了 Snort 等 IDS 请求，就需要把 IDS 传感器放置在防火墙内部。我们都知道互联网上存在很多恶意流量，所以通常不需要浪费资源证明它。

因为 Snort（免费）和运行 Snort 的硬件（便宜）的成本很低，实际的成本是安装、配置和进行监控的时间和劳动力。理想的情况是在 DMZ 中的所有公共服务器都配置 IDS 传感器，在防火墙内的 LAN 上、在以太网上（如果有），以这种顺序。这都取决于环境。要考虑瓶颈、可能的攻击途径和风险。考虑任何可能有用的位置，区分先后顺序，然后按照资源的允许根据列表执行。任何经过良好的策划和调制的入侵检测系统可以成为深度防御策略的优良辅助。而策划不够好而繁琐的 IDS 可能成为麻烦。

（作者：JP Vossen 译者：Tina Guo 来源：TechTarget 中国）

Snort：为 Snort IDS 传感器确定操作系统

确定了要安装 Snort 并决定了 IDS 传感器的位置后，必须决定网络传感器使用的操作系统。答案非常简单。

关于操作系统性能、稳定性和安全问题的激烈争论可以而且马上就会爆发。记住了这一点，决定使用带了非常安全的设备的操作系统的底线就是“使用你了解的”。对于你了解的操作系统，你可以更好的进行配置、加固、管理和解决问题。也可以更好的决定硬件和性能，而企业中的其他人也可能在需要的时候提供支持。如果你和企业实用几个操作系统，那么在内部作一些测试，或者选择运行最好的一个操作系统来配置你计划使用的硬件和 IDS 传感器。

以前也提到过，Snort IDS 是在 Linux 和 Mac OS X 上开发测试的，而且经过了实用 Linux 和 BSD 的人群的测试。这三个平台是首批支持的，而且开发人员很熟悉这些平台。在 Linux 或者 BSD 上实用调制良好的 Snort，就可以从较老的、慢速的或者便宜的硬件上获得可用性能，而在其他操作系统可能不可以。

最后，虽然学习新技术是我们都应该做的，但是保护网络的设备不是实验的好地方。

(作者：JP Vossen 译者：Tina Guo 来源：TechTarget 中国)

Snort：如何确定 IDS 的网络适配器

在决定了 Snort IDS 传感器使用的操作系统后，就需要配置网络。理想状态是最少有两个网络适配器（network interface cards，NICs）。其中一个是用于嗅探，而且应该没有编号——也就是没有分派 IP 地址。另一个应该和通常的一样使用 IP 地址，并仅用于管理。还有，你可以拥有很多的网络接入——编号的和没有编号的——只要硬件和操作系统可以支持。

管理借口应该在受信的网络上，通常是 LAN 或者专注管理的 VLAN 或者网络。可以和平常配置操作系统和环境一样配置。

对于没有编号的借口，在不受信任或者监控的网段上没有 IP 地址可以增加一层安全。因为没有可以做为目标的 IP 地址，这些网段就不容易被攻击，但并不是特别安全。从定义上看，Snort 查看流量。所以，Snort 或者网络数据捕获库中的漏洞可能被用于攻击，以前曾发生过。记住，传感器是安全设备，配置、加固和维护的时候应该记住这些。

Windows、Unix 和 Linux 都支持没编号的接口。例如，要在 Red Hat 或者派生的 Linux 分布上把 eth1 作为没编号得接口，就可以使用你最喜欢的文字编辑器来创建或者编辑/etc/sysconfig/network-scripts/ifcfg-eth1，使之看起来像

```
DEVICE=eth1

ONBOOT=yes
```

在 Windows 下运行没有编号的接口很容易，但是实直觉的计算。例如，在 Windows 2000 下，右击“My Network Places”并选择属性。右击恰当的链接，例如，“Local Area Connection 2”，然后再选择属性。通过检查网络适配器的名称和/或属性（例如，MAC 地址）验证你正在运行的是正确的物理界面，然后不选任何组件，特别是“Client

for Microsoft Networks”和 “Internet Protocol {TCP/IP}”。你可能觉着这样做是关闭适配器，但是并不是。在 ipconfig /all 下，显示不出来，但是如果使用 snort -W 命令就可以。运行 snort -W，并注意接口的号码，你可能会用它嗅探，然后测试 Snort 是不是用 snort -vi 2 等命令工作的。如果以后 Snort 突然停止运行，再次检查 snort -W，因为当改变网络时，Windows 有时会改变接口编号。

在任何情况下，都要确认在配置没有编号的网络接口后，数据可以恰当地发送。你不想要把管理接口插入到不受信任的网络，反之亦然。

(作者：JP Vossen 译者：Tina Guo 来源：TechTarget 中国)

Snort：修改和编写自定义 Snort 规则

Snort 规则强大而灵活，而且编写相对简单。最好从 VRT（Sourcefire Vulnerability Research Team）认证的规则开始，因为他们的编写最好，但是还有其他的规则资源。如果已经下载了已有的 IDS 规则，就可以根据需要修改规则。

在开始前，先查看 Snort.org 的常见疑难问答，可以在这里下载最新的 VRT 规则。（查阅 How to decipher the Oinkcode）。阅读这些规则、修改并实验（当然是在测试环境中）。如果已有的 Snort 规则不能按照想要的方式工作，就可以做些修改。下面是如何做。

所有的 Snort 规则都遵循一个简单的格式，这个格式应该检查一下。首先是 SID 和 Rev（revision）的笔记。SID 是 Snort 规则 ID（也就是 Signature ID）。Snort.org 保留了不到一百万条的“官方”规则，而 Bleeding Snort 使用的 SID 超过了两百万。如果修改一条规则，就在 SID 中增加了一百万，所以可以跟踪到原始的。如果要创建新规则，使用起点是九百万的 SID。当做出改变或者规则更改控制完成时增加修订。

有些 Snort 规则是空地址规则文件。不要使用它，或者你的自定义 IDS 规则可以在下一次安装新的规则的时候重写空文件。创建一个或者两个文件，例如 company_prod.rules 和 company_test.rules，并在 snort.conf 文件中增加包含语句。可以在一台计算机和接口上运行多个 Snort，这对于测试规则很有用，但是在和产品平行的旧的 PC 上运行测试版的 Snort 可能会更早、也更安全。

下面是一条简单的 Snort 规则：

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (msg:"ICMP PING NMAP"; dsize:0; itype:8; reference:arachnids, 162; classtype:attempted-recon; sid:469; rev:3;)
```

这些代码的意思是：如果发现了定义为(default = any)的$EXTERNAL_NET ICMP 内容到定义为(default = any)的$HOME_NET 的 ICMP 包就发送一个警告；如果数据大小（dsize）是 0，而 ICMP 类型（itype）是 8（就是 echo（request））。（还可以查看 http://www.snort.org/pub-bin/sigs.cgi? sid=469）除“警告”外的其他动作包括，但不止是，登录和通过。这种情况下的协议是 ICMP，但是也支持 IP， TCP 和 UDP。变量需要在配置文件中定义，并添加美元符号前缀。

SID469 不是很好的规则。它会产生很多的假阳性，因为它相当“宽容”。除了 NMAP 外的其他应用发送没有任何负载的 echo request 包，而且没有其他标准可以是这条规则“更严格”或者更详细。下面是一条较好的规则：

alert udp $EXTERNAL_NET any -> $SQL_SERVERS any (msg:"MS-SQL probe response overflow attempt"; content:"|05|"; depth:1; byte_test:2, >, 512, 1; content:"|3B|"; distance:0; isdataat:512, relative; content:!"|3B|"; within:512; reference:bugtraq, 9407; reference:cve, 2003-0903; reference:url, www.microsoft.com/technet/security/bulletin/MS04-003.mspx; classtype:attempted-user; sid:2329; rev:6;)

由于空间限制，我不能解释这条规则的每个细节，但是你只要看看就可以发现它很详细。你还可能注意到这些例子包括对外部资源的参考。查看 Snort“etc”目录下的“reference.config”文件中的 url，或者查看 Snort 网站上的 SID 中的规则文件，包括这些参考资料的连接。

Snort 2.1.0 版本引入了 PCRE（Perl Compatible Regular Expressions）、thresholding 和 suppression，所有的这些对适当的调制都很严格。PCRE 允许在规则中使用 Regular Expression，所以你搜索的很详细。三种类型的 thresholding 允许限制“噪杂”规则以各种方式发送的警告的数量，而且可以写入自定义规则，或者放置在单独的配置文件中，例如 threshold.conf。特别是，你可以使用外部文件 “调制”官方

Snort.org 规则。而不需要实际修改这些规则，这就使更新更加简单了。Suppression 命令允许在哪些设备允许或者不允许触发警告上更加细致具体。例如，如果你在 10.1.1.54 上有网络管理工作站，使用“public”社区字符串选择 SNMP 设备（这样很不好，对吧？）。你可以使用下面的规则来抑制这个情况：suppress gen_id 1， sig_id 1411，track by_src， ip 10.1.1.54

我们只是提到了表面。更全面的信息请查看 Snort.org 上的 Snort 用户手动指南，并从 Snort.org 和 BleedingSnort.com 上获取目前的规则。“readme”文件中包括 Snort 的源代码，这些源代码可以下载、检查来学习更多。如果你不想下载源代码，可以访问 Snort CVS Repository 网页上的 Snort 源代码、文档和废弃的规则。注意 CVS 库中的规则没有像 2005 年四月那样做维护，虽然有一些这方面的讨论，也可能会做出一些改变。可以阅读 Snort.org 上的文档或者加入 Snort 的 Sigs（Special Interest Group）邮件表。

（作者：JP Vossen 译者：Tina Guo 来源：TechTarget 中国）

Snort：如何配置 Snort 变量

Snort 有很多配置变量和选择，但是最重要的两个是$HOME_NET 和$EXTERNAL_NET。$HOME_NET 是详细说明网络或者你想要保护的网络的变量，而$EXTERNAL_NET 是你连接的外部的、不受信任的网络。这些变量实际上用于所有的规则，来指定信息包的来源和目的地的标准。

两种变量的默认值是“any”，意思就是字面上的意思。设置$HOME_NET 不需要费脑子。把这个变量配置到网络或者你想保护的网络上，就像在 snort.conf 配置文件中演示的那样。

$EXTERNAL_NET 更加灵活，而且有两种管理配置这个变量的方法。第一个是把它设置为默认的“any”，因为这样可以找出多数的事件。第二个方法是选择!$HOME_NET，照字面意思就是不属于$HOME_NET 的任何东西，因为这种设置可以减少假阳性，而且可以使 Snort 的运行更有效。尽管如此，它可能错失内部道内部（例如$HOME_NET 到$HOME_NET）的 攻击。因为 60%到 80%的攻击是内部的，这取决于你阅读的调查，我推荐把 $EXTERNAL_NET 设置为 any。还有，如果$HOME_NET 设置为 any，就不要把$EXTERNAL_NET 设置为!$HOME_NET，因为这样就使$EXTERNAL_NET 不做设置。

还有一些其它变量可以详细说明 DNS、SMTP、HTTP、SQL、Telnet、 SNMP、AIM 服务器、HTTP、SHELLCODE 以及 Oracle 端口。所有服务器变量的默认值是把他们设置为 $HOME_NET，这也是正确设置的另一个原因。这样工作良好，但是你可能发现你可以通过更详细的设置这些变量降低假阳性。如果使用的是不标准的端口，就应该验证端口变量，但是也可以不做处理。

（作者：JP Vossen 译者：Tina Guo 来源：TechTarget 中国）

Snort：在哪里查找 Snort IDS 规则

一旦安装、配置了 Snort，并已经开始工作，要考虑的下一件事情是规则。Snort 规则定义了用于查找网络上潜在恶意流量的方式和标准。没有这些 IDS 规则，Snort 仅仅只是另一个嗅探器。为了帮助你开始，这里提供四个可以查找到 Snort 规则的位置。

1．从 Snort.org 的下载官方规则快照。

在 2005 年三月七日，Sourcefire 变更了 Snort 规则的许可和分布。对于其他，Sourefire 创建了 VRT 认证规则，它是由 Sourefire 漏洞研究小组测试和证明的。为了开始使用这些规则和社区规则，可以查看 Snort.org 的常见疑难问答，然后下载从 Snort.org 选择的规则。如果从运行的 Snort 引擎上选择了正确的快照，就像在下在页上解释的一样，这些 IDS 规则保证可以工作。如果选择错了，Snort 可能不能启动。验证你使用的 Snort 版本（实际上，可以获取最新版本），然后再试一下。我强烈推荐，从 VRT 规则开始，并向他学习。

2．使用 Bleeding Snort 规则

如果你喜欢 Bleeding，使用 Bleeding Snort 规则可以达到两个目标（如果你的计算可以使你依靠 Bleeding，就是三个）。首先，这个站点是最新的有实验依据的规则和思想的交换处。而这些规则可能会是假阳性，而且有时可能不会向预期那样工作，他们需要经常跟新。第二，他们完全以激活大量可靠而准确的规则为目标，而当这些规则最终用于正式的 Snort.org 社区的规则库中时，就可以提供长期的价值。Bleeding Snort 规则本质上是测试或者 beta 规则，而且更适合已经测试了环境的人，以及喜欢 bleeding 的人或者必须更新 IDS 规则的人。

3．订阅 Snort-Sigs 列表

官方的 Snort-Sigs 邮件列表关注“Snort 规则的讨论和开发”。订阅信息和 Web 档案可以在 Snort.org 获得。因为他们讨论过的变化，大部分来自 Snort 社区的新规则都可以通过 Bleeding Snort 解决。

4. 自定义并共享规则

如果发现漏了什么，不要被编写规则吓怕了。你可以很快速很简单的编写规则。（学习如何编写，可以阅读最近的文章修改和编写自定义 Snort 规则）。还有，不要忘了通过 Bleeding Snort 规则和 Snort 社区共享这些规则。你可能会对面对相似问题的人提供帮助。

小心下载 Snort 规则的位置

最后，虽然可以在除上面提到的下载位置，在互联网上还可能找到 Snort 规则，但是我建议除非你真正的理解你要做的事情，否则要避免使用。规则语句已经发展到可以提供更好的方式来完成特定的目标，例如“建立”关键词，可以替换在很多环境中查看 TCP 的以前的方式。在 John Doe 的 任意网站上发现的这些 McRules 可能或者不能工作，所以为了安全最好避免使用。

2005 年四月，Snort.org 上有 3166 条激活的 VRT 规则和 33 条社区规则，而 BleedingSnort.com 的激活规则有 775 条，而且总是有增加。

（作者：JP Vossen 译者：Tina Guo 来源：TechTarget 中国）

Snort：如何自动更新 Snort 规则

如果使用 Snort，就可能发现有些默认规则没有用。你可能还发现需要的一些规则默认没有激活，而你可能需要修改一些规则。因为 Snort 需要像杀毒软件一样定期更新规则，每次下载新规则的时候都手动创建所有的变更不太实际。还有有一个免费的工具，叫做 Oinkmaster。它可以帮助维护 Snort 规则，并在 Unix 和 Windows 上运行。（它是用 Perl 语言编写的，所以在 Windows 上运行需要 ActivePerl。）

Oinkmaster 是 Snort 社区中的实际规则更新工具。它是经过几年的 beta 测试版本后，在 2004 年五月发布的产品版本。它是由配置文件驱动的，而在配置文件中可以正确定义它应该做什么。首先使用 HTTP、 HTTPS、 FTP、文件或者 SCP 的方法获取最新的规则。从 Snort.org 获取 VRT 规则需要 Oinkcode。然后确定更新或者跳过哪些文件、修改那些特征 ID（SID）、激活那些 SID、不激活哪些。你还可以“包括”一些文件到任意的深度，这可以允许使用标准的方法。Oinkmaster 的配置文件备有证明文件，而且包含有用的默认设置，但是仍然必须要查看并确保下载了正在运行的 Snort 版本的正确的规则快照，而且增加了 Oinkcode。

当使用 Oinkmaster 直接更新传感器的时候，较好地方法是使用一个配置文件来更新分段传输目录，并报告变更细节。一旦查看并通过了变更，就可以使用另一个配置文件来跟新传感器（查看 oinkmaster.sourceforge.net 上的 Oinkmaster 的常见疑难问答的第三个问题。）

激活或者不激活 SID 很容易；你要做的就是在“激活的 SID”或者“没激活的 SID”中增加 SID。修改一条 SID 很困难，而且包括了创建一个 Perl Regular Expression 来描述变更。还有模板功能可以执行多个相似的变更。因为 Snort 规则还可以使用 Regular Expressions。它在其他程序领域也很有用，如果对它不熟悉，值得学一下。

1.1 版本是在 2004 年十月发布的，而且增加了两个报告格式，可以更清楚地显示当规则变化（在变更日志中查看 -s 和 -m）时哪里不一样了。1.2 版本是在 2005 年四月发布的，可以允许在 oinkmaster.conf 中存在多个-u ...[命令行]或则和多个“url = ...”[语句]，使下载 VRT、社区和 Bleeding Snort 规则更便利、更迅速。Oinkmaster 的档案编写的很好，使其成为学习这种工具的很好的开始。如果你使用 Snort，就多花点时间吧——你不会后悔的，我保证。

（作者：JP Vossen 译者：Tina Guo 来源：TechTarget 中国）

Snort：如何破译 Snort VRT 规则的 Oinkcode

Sourcefire 已经变更了 Snort 规则的许可和分布，而且创建了 VRT 认证规则。VRT 认证规则是由 Sourcefire 漏洞研究小组（Sourcefire Vulnerability Research Team）测试和认证的。

作为终端用户，可以通过以下三种方法中的任意一种获取这些规则：

- ✓ 向 Soucefire 支付订阅费用，在发布规则后就可以尽快获得
- ✓ 免费注册，在付费订阅发布的五天后获取规则
- ✓ 在 Snort 引擎发布的时候获取通用的规则（也就是在引擎更新之间没有更新，不推荐使用这种方式）

如果你是“Snort 综合者”，还要考虑一些问题。无论如何，都需要阅读 Snort.org 上的常见疑难问答。

如果你的企业是 Snort 终端用户，你当然想要免费注册并产生 Oinkcode，这样就可以下载 VRT 规则。一旦你已经阅读并且理解了这些疑难的答案，就可以在 Snort.org 创建上账户并登录。在账户设置页面，可以找到一个产生 Oinkcode 按钮，如果使用 Oinkmaster，还可以找到配置 oinkmaster.conf 的说明。但是即使没有使用，你建立的 URL（例如 http://www.snort.org/pub-bin/oinkmaster.cgi/5a081649c06a277e1022e1284bdc8fabda70e2a4/snortrules-snapshot-2.3.tar.gz）在浏览器中可能被加为书签，或者使用 wget 或其它的下载工具下载这些规则。

你的 VRT 认证规则的 Oinkcode 可能是像这样的：

http://www.snort.org/pub-bin/oinkmaster.cgi//

Snort2.3 的例子：

http://www.snort.org/pub-bin/oinkmaster.cgi/5a081649c06a277e1022e1284bdc8fabda70e2a4/snortrules-snapshot-2.3.tar.gz

还可以从 Snort.org 获得 GPL 社区规则（免费，不需要注册或者 Oinkcode，非常有限。

（作者：JP Vossen 译者：Tina Guo 来源：TechTarget 中国）

Snort：使用 IDS 规则测试 Snort

新的 Snort 系统的运行是不是太安静了？不论你是使用新的 Snort，还是在新的平台上配置的——你所担心的低噪音等级。它可能会是严谨的调制（或者过分调制）的系统，或者你可能在一个平静的网段上拥有 IDS，还好还有一些在线测试 Snort 的方法，确保它在你的环境中正常运行。

可以从命令行中用嗅探的方式开始运行，这样可以确认网卡运行正常，span 端口被激活了（查阅如何使用交换器和分段处理网络设计），而且 Snort 实际可以查看到流量。在你使用多个网络适配器的情况下（请查阅如何如何确定 IDS 的网络适配器），你需要确定 Snort 使用那一个。为了在 Linux/Unix 中找到适配器的名称，可以使用 ifconfig；在 Windows 中，使用 Snort -W。然后使用 snort -vi（适配器名称）；例如，在 Linux 中的 snort -vi eth1 或者 Windows 中的 snort -vi 2，告知 Snort 嗅探哪个网络适配器。如果所有的都可以正常工作，信息包标头流量的信息（和 tcpdump/windump 相似）屏幕翻动的速度比阅读的速度快。按 CTRL-C，停止捕获或者查看信息包数据，例如分析的信息包数据、协议的崩溃、分裂以及其他。还要使用 -d (dump)和 -q (quiet)试验，查看他们如何影响发动流量。

你可以使用一些简单的测试规则手动检查 Snort。为了测试工作，你需要再设置中增加一条或者多条规则。最简单的方法是在 snort.conf 文件的末尾增加，虽然你也可以创建 test.rules 文件，并包括到 snort.conf 中。你必须有从定义为$EXTERNAL_NET 的网络向定义为$HOME_NET（查阅如何配置 Snort 变量）的网络发动信息包的能力。

- alert ip any any -> any any (msg:"Got an IP Packet"; classtype:not-suspicious; sid:2000000; rev:1;)
- alert icmp any any -> any any (msg:"Got an ICMP Packet"; classtype:not-suspicious; sid:2000001; rev:1;)

- alert icmp any any -> any any (msg:"ICMP Large ICMP Packet"; dsize:>800; reference:arachnids, 246; classtype:bad-unknown; sid:2000499; rev:4;)

前面两条 snort 规则应该产生对查看任何 IP 或者 ICMP 信息包的警告。因为他们可以触发网络上的几乎每个单独的信息包，没有你想要在网络上运行超负载的产品的规则。如果需要就在较小的或者测试网段上运行。最后一条规则是 SID（规则）499（注意 Snort.org 的“官方”规则的 SID 是 1-1000000。查看 Snort.org 的 Snort 用户手册）的复本，它已经修改的更松散，这样可以为我们测试的目的增加警告的产生。正常情况下，你可能需要避免松散的规则，因为可能导致假阳性。还有，原始的规则已经不被重视，而且放入了 deleted.rule 文件中。为了使用上面提到的规则，ping -s 1024 {target host} (Linux)或者 ping -l 1024 (target host) (Windows)。如果这些测试都没有起作用，那么 Snort 可能没有工作和/或信息包没有通过。不要忘了在结束的时候删除测试规则。

最后，Snort 的测试交换机（-T）可以允许简单的测试配置中提出的变化。可以运行这样的命令 snort -c /etc/snort/snort.conf -T，并阅读发送流量查看配置是否起作用了。Snort 还可以在起作用的时候返回 0 代码，而在失败的时候返回其他（通常是 1）。这可以通过运行下面两条命令中的一条表示出来：snort -c /etc/snort/snort.conf -T & echo "Return code: $? " (Linux)或者 snort -c ./Snort.conf -T & echo Return code: %ERRORLEVEL% (Windows)。因为总是可以运行不止一条的 Snort 复本，你可以使其中一个保持运行，使用另一个改变配置并测试，然后停止生产过程并快速重启，在测试后安装变化。

在线测试 Snort 还要注意：有些旧规则使用 TCP header flag 查看信息包是购使已经建立的 TCP 会话。新的规则使用已经创建的关键词（查阅在哪里查找 Snort IDS 规则）。在这两种情况中，你不能简单地使用 Netcat 把预期的 TCp 信息包负载放置在外面，而期待 Snort 可以查看到它——负载必须要显示为已经常见的 TCp 会话的一部分，在合适的方位，这需要在 Snort 触发警告之前。“已经常见的”关键词对于减少假阳性很重要，但是

在测试 Snort 时，可能会混乱，这也是我们使用 ICMP 或者上面说到的自定义规则的原因。

(作者：JP Vossen 译者：Tina Guo 来源：TechTarget 中国)