

# 提高 snort 性能的方法

snort 是一个开源的轻量级入侵检测系统, 由 Martin Roesch 等人开发。它结构灵活, 功能强大, 易于配置, 可在多个平台上运行, 在 Internet 上广为流行, 也给很多商业入侵检测系统的开发提供了思路。它是基于网络的入侵检测系统, 所以网络流量的大小对它影响很大。现在网络通信能力飞速发展, 如何尽可能提高 snort 的检测性能便成为大家关心的问题。希望以下建议能够给大家提供一点帮助。

1、根据自己的网络环境, 调整 snort 的规则。在任何平台上, 规则都是影响 snort 性能的重要因素, 花几天时间来优化你的规则是非常值得的。

2、使用 FAST 警报模式和二进制日志方式。这在 snort 的用户手册和 FAQ 里面都已经明确: 二进制方式和 ASCII 方式的性能相差极大。

3、在探测器上使用高质量的网卡。对于 100M 的环境, 推荐使用 Intel PRO/100 网卡。对于千兆网络环境, 推荐使用 INTEL 千兆服务器网卡。

4、使用高性能的磁盘系统。高性能的磁盘系统可以减少 snort 花在磁盘 I/O 上的时间, 同理, 也可以用 Barracuda 来减少磁盘 I/O 的问题。在内存如此廉价的今天, 使用内存文件系统的代价也不是不可接受的。

5、使用高性能的处理器。一个强劲

的处理器当然是 snort 性能的前提, 而且它是越来越便宜了。曾经有人提到, 在 linux 系统下, 仔细配置规则, PIII 1G 的处理器可以轻松处理 100M 网络的数据。

6、重新编译 snort。从性能角度考虑, 最好不要使用 snort 网站提供的二进制代码, 它在你的计算机上不一定表现良好。

7、重新编译内核, 将那些你不需要的功能去掉。预编译好的 RedHat 内核中有很多的功能, 使得用户易于使用, 但这是以降低系统性能为代价的。根据系统的配置和要求, 重新编译 linux 内核会对性能有帮助。

(楚风)

2.3 版本的 squid 的配置文件 squid.conf 需要如下条目:

http\_port 3128

httpd\_accel\_host virtual

httpd\_accel\_port 80

httpd\_accel\_with\_proxy on


httpd\_accel\_uses\_host\_header on

2.4 版本的 squid 需要增加如下选项:

httpd\_accel\_single\_host off

中文文章可查看以下网址: <http://www.neweasier.com/article/2002-08-02/1028302489.html>

## 7、如何能够实现对一个数据报 DROP 同时 LOG?

 LOG 不会终结一个数据报的处理, 通过一个 LOG 规则后, 系统会继续匹配下一条规则。如果需要同时 DROP 和 LOG, 可以建立定制一个规则链, 命令如下:

```
iptables -N logdrop
```


```
iptables -A logdrop -j LOG
```

```
iptables -A logdrop -j DROP
```


这样, 对需要 LOG 和 DROP 的数据报只要简单的使用

“-j logdrop”就可以了。

## 8、为什么连接管理中 UNREPLIED 连接的超时时间非常长?

 如果你查看 /proc/net/ip\_conntrack, 会发现其中 UNREPLIED 连接有一个非常长的超时时间 (最大到 5 天), 为什么会如此浪费连接管理的条目呢? 很简单: UNREPLIED 条目是临时条目, 如果我们用完了连接管理的条目 (最大连接数在 /proc/sys/net/ipv4/ip\_conntrack\_max 中设定), 我们就删除旧的 UNREPLIED 连接。也就是说: 与其放着一个空的条目, 不如保留一些信息在其中, 直到我们真正需要使用这个条目。

## 9、为什么 iptables 中没有实现一个 “-C(-check)” 选项?

 实现一个 “-C(-check)” 选项基本上是不可能的。在传统的无状态防火墙中, 数据报的处理取决于报头提供的信息, 但是有连接管理后, 数据报的处理不仅与报头有关, 还与载荷甚至当前连接中以前的数据报有关。

(mboy)