Snort 规则集的优化方法

董明明，巩青歌
（武警工程学院，陕西 西安 710086）

摘 要：高速网络的发展使得提高检测速度成为入侵检测系统面临的关键问题。通过对 Snort 规则集优化方法的分析与比较，提出将活跃规则集划分与多子集划分相结合的方法，先从整体上优先选择要匹配的规则集，然后进行并行匹配，以提升入侵检测中规则匹配的效率。

关键词：Snort ； 入侵检测 ； 规则匹配

Research and Improvement on Optimizing Snort Rule Sets

DONG Ming-ming, GONG Qing-ge
(Engineering College of China Armed Police Force, Xi'an, Shanxi 710086,China)

Abstract: The development of high speed network makes that how to improve the detection rate a key problem for the intrusion detection system. This paper proposed combining dividing active rule sets and dividing multi-subsets by analyzing and comparing several methods of optimizing the Snort rule sets, first choosing the rule sets for matching, and then parallel matching them to improve the matching rate effectively.

Key words: Snort ; IDS ; Rule matching

1 引言

随着计算机和网络通信技术的广泛应用，网络安全威胁的日益增长，网络安全问题已经越来越引起人们的重视。网络入侵检测系统（NIDS）提供一种积极主动的安全防护，提供了对内部攻击、外部攻击和误操作的实时保护，成为继防火墙之后安全防护的第二道闸门。Snort 是一个开源的 NIDS，能完成实时流量分析和对网络上的 IP 包登录进行测试、协议分析、内容查找／匹配和探测多种供给和嗅探等功能，是目前入侵检测系统中一个重要的研究方向。

规则匹配一直是入侵检测系统的核心问题，是 Snort 系统主要的性能瓶颈，约占整个系统运行时间的 30%。规则匹配算法也一直是研究的热点问题，单模式匹配算法（BM 算法），多模式匹配算法（Aho-Corasic 算法并行搜索模式）以及 Wu-Manber 算法的提出，还有各种改进算法不断推出，但是随着网络流量的增加，网络速度的提升，仅仅通过研究算法本身的特性去提高检测效率已经远远不能满足网络安全防护的需求。基于此，本文从 Snort 规则集入手，通过对现有规则集优化方法的分析，提出了具有更高效率的优化方法。

2 Snort 规则匹配原理

Snort 规则包含两个逻辑部分内容：规则头和规则选项。规则头包含执行的动作、使用的协议、源（目的）IP 地址和掩码、端口信息及数据流向等。规则选项 OTN(Optional Tree Node) 包含一些报警信息 (msg)、匹配内容 (content) 项及用于确定是否触发规则相应动作而需检查的数据包的详细信息。

Snort 将所有已知的攻击以规则的形式存放在规则库中，系统初始化并解析规则时，根据规则所用协议分别分成 TCP、UDP、ICMP 和 IP 四个不同的规则树。每个规则树包含独立的三维链表：RuleTreeNode（规则头）、OptTreeNode（规则选项）和函数指针（指向规则行为）。当 Snort 发送一个数据包到规则检测引擎时，首先分析该数据包使用哪个协议以决定将与之对应的规则树进行匹配；然后与 RTN 节点依次进行匹配，当与一个链表头节点相匹配时，向下与逐个 OTN 节点进行匹配。每个 OTN 节点包含一条规则所对应的全部选项，同时包含一组函数指针，用来实现对这些选项的匹配操作。当数据包与某个 OTN 节点相匹配时，即判断此数据包为攻击数据包并将触发规则中定义的规则行为。

3 现有优化方法分析

标准的 Snort 系统采用单进程方式运行，对截获的每个数据包都需逐个串行地进行检测，这种策略在检测规则集数目较少时是比较有效的，当面对规则库里大量的规则该策略已无法满足用户的需求。为提高规则匹配速度，Snort 也采用了二维列表递归检索 (RTN 和 OTN) 以及函数指针列表等方法，许多优化方法从构造检测引擎方面来改变规则集的结构，提出各种搜索策略。

Snort 自 2.0 版本后引入了快速检测引擎(FPDE) [1]，提出了一种新的创建规则索引的思路(见图 1)。Snort 初始化时，会根据配置文件的要求加载相应的规则并生成规则树。随后，Snort 对规则进行三层分类，分类层次从上至下依次为：协议映射类(PORT_RULE_MAP)、源／目标端口集合类(PORT_GROUPs)、内容集合类(PORT_GROUP)。这一思路大大提高了系统的性能，但这种方法与以前版本相比只是一种静态的改进。

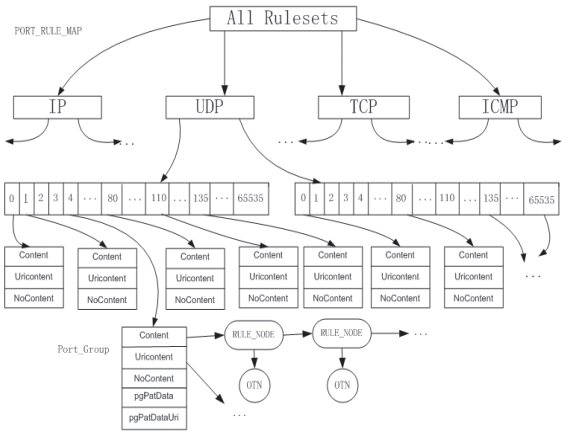

图 1 Snort 规则所引和规则树

在搜索过程中，增加优先级节点的方法在一定程度上提高了检测速度 [2]。在生成规则树时，根据规则的优先级，在协议下增加一条优先级选项链 Priority，并将程序做了相应改动。通过分析其实验数据，虽然该处理过程发生在系统生成规则树时，也就是系统初始化时，不影响系统检测数据包的时间，但在提高检测速度上效果不是很明显。

考虑到尽管 Snort 的规则数目越来越多，但在一定时间内只有一小部分规则是活跃的，大部分规则不会被触发，于是提出了一种新的规则分类组织排序方法，把每个端口下的规则集分成活跃规则集与不活跃规则集 [3]。该方法虽然使 Snort 的性能有了改进，在一定程度上提高了系统规则匹配的效率，但对于规则集的匹配仍需要逐一进行，试验也表明这种方法对检测速度没有大幅度的提升。而在此基础上，将 Snort 规则划分为多个子集，使得每一个数据包匹配时仅有一个规则集与之匹配，从而只需要在这个规则集内对这个包进行匹配，这样可以弥补对 Snort 的每一个数据报都要对所有规则进行匹配的缺点，能够更有效地提高规则匹配速率。

4 改进方法

在 Snort 的 2.2 版本的规则库中 [4]，有 292 条独立的头规则，2107 个固定长度的标识，还有 233 个常规表达式。如今计算机应用的普及使得网络入侵攻击手段不断更新，为了有效应对各种攻击，要求 Snort 规则库必须不断增加新规则以增强检测的效能。在 Snort 的实现中，相对耗时的操作主要有：从网络传输介质上捕获数据包，分析数据结构，规则匹配以及对每一个数据包进行的校验。其中，规则匹配是 Snort 的核心运算模块，如果能提高它的运算速度，将有效提高 Snort 的整体性能。

4.1 活跃规则集的建立

对于一个端口下的全部规则集合 $\{R_1,R_2,\cdots,R_n\}$，取一段时间 t，假设网络流量为每秒 p 个数据包，每条规则对应的攻击出现的概率分别为 $P_1,P_2,\cdots,P_n$，则每秒钟与规则 Rk 对应攻击数据包的数量是 $p*P_k$。启动 Snort 一段时间，可根据网络流量调整时间的长短，如一周或一个月，假设匹配频度不为零的规则序号依次为 1,2,…,i，M_n 代表对应规则的匹配频度，通过一定的算法，每隔一段时间比较各规则的匹配程度，进行活跃规则集与不活跃规则集的重新划分，实现和保持活跃规则集的动态建立。

4.2 规则树的建立

在 RuleTreeNode 下增加一个 OptGroupNode 选项节点，选项链由两个节点组成，包括三个内容：flags 的值(可赋值为 0 或 1)，一个向右的指针(指向下一个 OptGroupNode 选项节点)以及一个向下的指针(指向选项链)。把每个端口下的规则分成两组，一组为活跃规则集(flags 值为 0)，另一组为不活跃规则集(flags 值为 1)。

在 Snort 的规则解析函数 ParseRulesFile() 中为每一个 OTN 生成一个指向该选项节点的指针 OptIndexNode，并将它与规则头相连，形成“选项索引链表”，如图 2 所示。

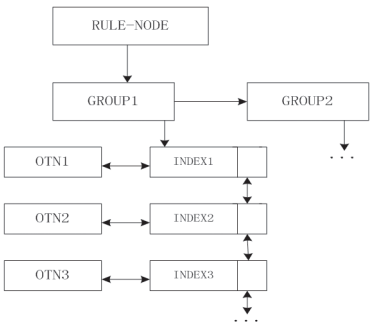

图 2 增加 OptIndexNode 索引后的规则树示意图

图 2 中，GROUP1 代表此规则集为活跃规则集，GROUP2 代表此规则集为不活跃规则集，匹配时先匹配 GROUP1，然后对其中的每个 OTN 节点进行匹配，没有相匹配的时候再匹配 GROUP2。

根据快速检测引擎 FPDE 的设计思想，将 GROUP1 和 GROUP2 下的各匹配规则划分为多个子集并对其建立索引节点。(GROUP2 下子集的划分同 GROUP1，在图中没有画出)。INDEXM(M=1,2,…) 节点分别代表将活跃规则集基于某种惟一的参数(如源端口号、目的端口号和规则内容选项等)来划分为多个子集的索引节点。将规则按照目的端口号进行升序排序，在目的端口号相同时再按源端口号排序，应用统计学中聚类分析方法确定每个规则

子集下包含的规则数，这样划分是为了尽量创建最小、最有效率的规则子集；并且所创建的规则子集是独立的，使得每一个被检测的数据包，只需要对一个规则集进行匹配。然后将多个规则头建立一个索引，索引值为每一组的最小端口号。

在进行数据包匹配时，先利用索引值进行搜索，在匹配索引值以后，检测算法对每个规则进行并行匹配。对于端口号为确定值的规则类型（TCP/UDP 协议类型），直接使用其值；对于端口号为不确定值的情况，如果端口号位 ANY 或者为一个范围，设其值为零；对于不含端口值的规则类型（ICMP 或 IP 协议类型），采用如下处理方法：对于 ICMP 协议规则，如果规则中指定了 ICMP 类型值和代码制，则源端口值指定为 ICMP 类型值，否则，规定为任意 ANY(0)。对于 IP 协议类型，源端口号指定为任意值 ANY(0)，如果规则中指定了传输协议 ID 值，则目的端口值指定为传输协议 ID 值，否则指定为任意值 ANY(0)。

在构造检测引擎时，将源端口号和目的端口号均为任意值 ANY(0) 的情况单独列出，并作为第一个索引值，因为这一列规则是所有数据包都要检测的。以上划分使得不同类型的数据包根据其自身特点被归到一个字规则集中，规则检测引擎在活跃规则集检测中只需检测较小的规则集，并且不会出现重复检测的现象。

对于将规则集进行活跃与不活跃的划分，实验已经证明，使规则匹配时间缩短[5]。在此基础上提出对活跃规则集与不活跃规则集进行多子集划分的方法，减少了匹配操作，使规则匹配速度更快，系统性能得到更大的提高。

5 总结

本文从提高规则匹配速度的另一个角度——规则集入手，通过对规则集的划分来改进匹配的效率，在活跃规则集与不活跃规则集划分的基础上，结合快速检测引擎（FPDE）的构造思想，引入多子集划分的思路，建立了改进后的检测方案，使得规则检测减少了检测操作，另外匹配算法可以并行地进行匹配，更有效地提高入侵检测系统的匹配速度，更好地保障网络安全。

注：本论文提出的改进方案将应用于上级交给的安全项目当中。

参考文献：

[1]ArboledaAF.Snortdevelopmentdiagrams[EB/OL].[2005-04-14]. http://afrodita.unicauca.end.co/cbedon/sonrt/snortdevdiagrams.pdf.

[2] 王志强，王猛，提高 Snort 规则匹配速度方法的研究与实现．计算机安全 2008.8:20-22.

[3] 张雅羚，谢绍春，唐来风，基于活跃规则集的 Snort 高校规则匹配方法．计算机工程与应用，2008,44(24):124-127.

[4]Michael Attig and John Lockwood:Snort Intrusion Filter for TCP,COMPUTER SOCIETY,2005.

[5]Snort 2.0: Rule Optimizer. Sourcefire, Inc. http://www.snort.org.2004.4.

作者简介：**董明明**（1986-），女，硕士研究生，主要研究方向：网络安全、入侵检测；**巩青歌**，女，副教授，硕士生导师，主要研究方向：数据挖掘、网络安全。

收稿日期：2009-01-03

（上接第 34 页）

讲，攻击分析和特征提取可以借鉴其化领域中处理数据信息的一些成熟的理念、方法和技术，如机器学习、人工智能、数据挖掘等，以加强基于数据可视化、协议还原、攻击工具自动识别、入侵场景自动分析、攻击特征自动提取等技术，提供更多深层的安全信息。

蜜罐技术是一种新兴的技术，还处于发展阶段。由于它有着其他技术无可取代的优点，成为一个完整防护体系不可或缺的一部分，相信会得到研究人员的广泛关注，也相信随着技术的不断完善，蜜罐技术将来会得到更广泛的应用，发挥更大的作用。

参考文献：

[1] 熊华．网络安全——取证与蜜罐 [M]．北京人民邮电出版社 .2003.

[2] 肖军弼，刘广祎．分布式蜜罐系统的设计与实现 [J]．计算机工程与设计 .2007.

[3] 程杰仁，殷建平．蜜罐及蜜网技术研究进展 [J]．计算机研究与发展．2008.

[4] Honey Project. Know your enemy:honeynets. Http://www.honeynet.org/.2003.

[5] Lance Spitzner. The Value of Honeypots. http: //www.spitzner.net/.2003.

作者简介：**陈超**（1986-），男，硕士研究生，主要研究方向为指挥自动化；**妙全兴**（1965-），男，副教授，研究生导师，主要研究方向：计算机网络、信息安全。

收稿日期：2009-03-13