文章编号: 1003-5850(2008)11-0011-03

具有适应性的 Snort 规则树构建方法

The Method for Constructing Snort Rules Tree with Adaptability

苗宝秋 孙 敏

(山西大学计算机与信息技术学院 太原 030006)

【摘 要】合理构建规则树可以提高 Snort 的匹配效率。当前常见的规则树构建方法不能很好反映实际工作环境的特点, 从而造成某些重要的规则属性得不到优先匹配。主要是基于对网络实际数据的统计来构建规则树, 提出了一种属性重要性的测度方法, 使得规则树能够适应实际的网络工作环境, 从而提高了规则匹配的速度。

【关键词】入侵检测, 规则树, 适应性算法, 属性重要性测度

中图分类号: TP393.08 文献标识码: A

ABSTRACT Constructing Snort rules tree reasonably can improve the efficiency of rules matching. But in current, the process of constructing rules tree can't reflect the characteristics of practical work environment well, and then some important attributes can't be matched firstly. This paper proposes a method to measure significance of attributes, and the process of constructing rules tree is based on statistics of the real data packets, so it can increase the speed of rules matching.

KEYWORDS intrusion detection, rules tree, adaptive algorithms, measure significance of attributes

Snort 是一个开放源码的、轻量级的误用网络入侵检测系统(Network Intrusion Detection System, NIDS)。基于规则的模式匹配是 Snort 的核心, 即针对每一种已知的入侵行为, 都归纳出它的特征值并按照一定的规范书写成检测规则, 从而形成规则库; 将捕获的数据包与规则库中的规则进行匹配。随着规则库中规则数目的增加, 规则结构也越来越复杂, 这就对规则匹配引擎的性能提出更高的要求。

1 Snort 规则

1.1 规则的构成

Snort 规则库中的每一条规则都是一条攻击标识, 逻辑上可分为两部分: 规则头(括号左边的内容)和规则体(括号内的内容)。规则头规定了该规则被触发时的动作, 以及该规则适用的一组协议字段的值; 规则体是在规则头的基础上作进一步的分析, 规则体由若干个被分号隔开的片断组成, 每个片断定义了一个选项和相应的选项值, 其中最重要最有用的是那些分析数据包内容的 Content 选项[1]。如下面的一条用以检测尼姆达蠕虫的规则:

alert tcp EXTERNAL NET any→HOME NET 139 (msg: "NETBIOS nimda.nws"; content: "|00|.|00|N|00|W|00|S";)

这条规则被触发时将执行 alert 动作, 它将和有如下属性值的数据包匹配: 传输层协议是 TCP, 源 IP 地址包含在全局变量 EXTERNAL NET 中, 目标 IP 地址包含在全局变量 HOME NET 中, 目标端口的值是 139, 并且在净荷数据中包含字符串"|00|.|00|N|00|W|00|S"。

1.2 规则树

最简单的规则匹配策略是让捕获到的每一个数据包与规则库中的全部规则逐一匹配, 这就意味着要对待检测数据包的同一个属性作多次重复匹配, 大大降低了检测的效率。为了使规则匹配尽可能的并行化, 对数据包同一个属性的匹配次数尽可能的少, Snort 组织规则库是按照规则的处理动作(Alert, Dynamic, Activation, Pass, Log)来划分成 5 个链表, 其中每个链表又按照协议类型(TCP, UDP, IP, ICMP)分成 4 个二维链表, 所有的规则都会被分配到这些二维链表中。每个二维链表又由规则树节点(RuleTreeNode, RTN)和选项树节点(OptionTreeNode, OTN)组成, 如图 1 所示。

每一个规则树节点 RTN 代表了一组规则的共有

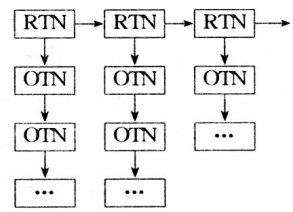

图 1 Snort规则树

* 2008-05-29 收到, 2008-09-21 改回

** 基金项目: 山西省高校科技开发项目(20051202)。

*** 苗宝秋, 男, 1977 年生, 硕士, 研究方向: 网络安全。

属性值, 选项树节点 OTN 是规则中更详细的匹配内容。程序先进行的就是选择规则树的工作, 即在由 RTN 构成的链中确定一个匹配的 RTN, 当选定一个之后, 就从该节点向下对 OTN 进行匹配。

早期 Snort 版本中, 如果 RTN 匹配成功, 则逐条检测规则子集中的每一条规则, 规则选项也是依次、独立地被检测。由于规则子集中的规则被线性检测, 某些属性被多次重复匹配, 因此降低了检测的速度。

Snort2.0 改进了检测引擎机制, 采用了基于集合的检测方法。选取唯一能够代表规则集合的参数作为分类依据, 如 ICMP 的 type 参数、TCP/UDP 的端口号, 将所有具有相同内容的规则放在一起。这样做的目的就是为了划分出更小的规则子集并且提高规则匹配的并行度[2]。但从 Snort 整个检测的结构和过程来看, 规则树的构建依赖于专家的人为指定, 致使规则树过于简单, 造成某些 RTN 下的 OTN 链比较庞大, 是影响 Snort 检测效率的重要因素。数据挖掘的方法也曾应用在该领域, 使得规则树搜索过程得到了优化[3], 但也很难反映 Snort 所处的工作环境特点。

2 适应性算法构建规则树

2.1 规则树构建算法

Snort 规则树的构建应该适应其所在的工作环境, 包括实际的网络数据和不断更新的规则集的特点, 本文针对此问题提出一种具有适应性的规则树构建算法, 其基本策略是:

① 所有的规则置于一个集合中, 作为规则树的根节点;

② 从待检测的数据中读取一定的数据样本, 通过统计对规则的不同属性的重要性作出测度;

③ 选取重要性最大的属性对规则集进行划分, 该属性值相同的规则划为一类作为子节点;

④ 对每一个子节点再用同样的方法, 从尚未用过的属性中选取出属性重要性最大的属性, 作更进一步的划分;

⑤ 如此递归执行, 直到每一个子节点满足下列条件之一为止: a. 没有剩余的属性可以用来作进一步划分; b. 剩余属性的重要性已小于或等于阈值 Smin。

显然, 该算法的关键之处在于如何对属性重要性作出测度, 才能使得构建出来的规则树对其所在的工作环境具有适应性。

2.2 属性重要性的测度

采用统计的方法, 利用两个因子来对属性的重要性进行测度。

2.2.1 因子一: 该属性排除规则的能力

通过实验统计发现: 对于一个给定的数据包样本和一个给定的规则集, 让样本中的每一个数据包和规则集中的所有规则进行一次匹配, 通过匹配不同的属性所能排除掉(即匹配不成功)的规则条数的平均值是不同的。本文选取 Snort2.1.3 的规则集共 2 059 条规则, 数据包样本来自本地实验室的网络环境, 实验结果如图 2 所示。

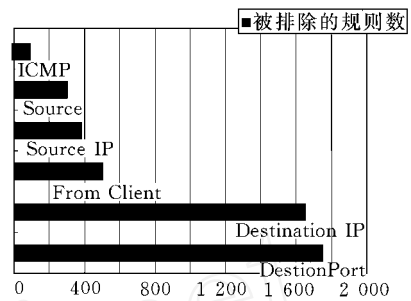

图 2 匹配不同属性所能排除的平均规则条数

这里基于这样一个认识: 选取某一属性对规则集作一次划分, 该属性能排除规则条数的期望值越大, 则下一次划分所考虑的规则集越小, 所以通过属性能排除的规则条数可以测度属性重要性[4]。本文对数据作归一化处理: 设通过匹配属性 A_i 能排除的规则数为 m_i, 所考虑的规则集的规则总数为 m, 则 $N_i = m_i / m$ 作为测度 A_i 属性重要性的第一个因子。

2.2.2 因子二: 该属性与其他属性的相关度

为了反映属性之间的相关关系, 这里提出属性的相关度矩阵:

	$A_1, A_2, \cdots, A_n$	$T_i = \sum_{j=1, j\neq i}^{n} a_{ij}$	$R_i = 1/T_i$
A_1	-, a_{12}, …, a_{1n}	T_1	R_1
A_2	a_{21}, -, …, a_{2n}	T_2	R_2
…	…, …, -, …	…	…
A_n	a_{n1}, a_{n2}, …, -	T_n	R_n

矩阵元素 a_{ij} $(i \neq j)$ 是这样得到的: 对于一个给定的规则集和给定的数据包样本, 假设通过匹配属性 A_i 能排除的规则数为 m_i, 而在这 m_i 条规则中能够被属性 A_j 排除的规则条数为 m_j, 则 $a_{ij} = m_j / m_i$。它反映的是属性 A_i 与属性 A_j 的信息重复量。按行求和得到 $T_i = \sum_{j=1, j\neq i}^{n} a_{ij}$ 表示属性 A_i 与其他所有属性的信息重复量, 将其求倒数得到 $R_i = 1/T_i$, 作为测度属性重要性的第二个因子。

最后, 对属性重要性的测度结果为 $I_i = N_i * R_i$, 这种测度方法从两个方面所映了当前工作环境的特征, 使得构建出来的规则树具有对工作环境的适应性。

2.3 算法实例

基于上述理论,以从实际实验室的网络环境中捕获的数据作为样本,对 Snort2.1.3 规则集作递归的划分,得到的规则树如图 3 所示。

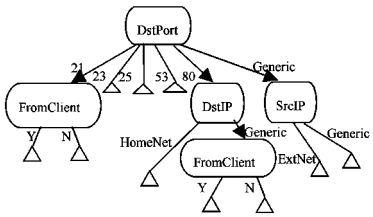

图 3 具有适应性的规则树

用此规则树来指导规则属性的匹配:优先匹配目标端口,再根据不同的规则子集优先选取不同的属性。对于某些数据包,不需要与所有的规则属性一一匹配,仅匹配一至两个就能得到分类结果,这样就提高了入侵检测速度。

3 实验评测

为了对改进后的检测模型做出量化的评测,本文选取 1999 DARPA 第四周的数据进行实验,经过反复比对,取 $S_{min}=1.2*10^{-3}$,结果显示,以每秒钟处理的数据包个数为指标,改进后的模型性能得到了不同程度的提高,结果如图 4 所示。

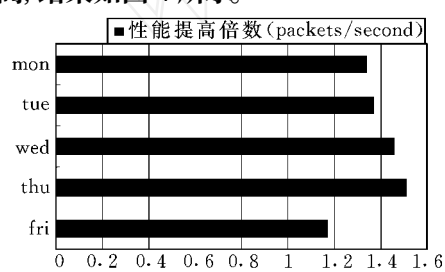

图 4 不同数据集评测的性能提高倍数

综上所述,用适应性算法优先选取最重要属性,将减少属性匹配的次数,从而提高分类的速度,提高 Snort 的检测效率。规则属性的个数越多、规则库中的规则数量越多,采用上述方法,匹配速度提高的越明显。

4 结束语

本文应用基于统计的适应性算法,针对给定的规则集和数据样本,对规则的属性重要性做出了测度,来指导 Snort 规则树的构建。选取对分类最有利的属性优先匹配,可以较好的提高分类速度。

参考文献

[1] Brain C, Jay B 著. 宋劲松译. Snort2.0 入侵检测[M]. 北京:国防工业出版社, 2004

[2] SNORT2.0 Rule Optimizer[EB/OL]. http://www.sourcefire.com/products/library.html#wp, 2007-09-10.

[3] Christopher K, Thomas T. Using Decision Trees to Improve Signature-based Intrusion Detection[A]// Vigna. Recent Advances in Intrusion Detection. Sixth Symposium on Recent Advances in Intrusion Detection[C]. Pittsburgh, PA, USA, September 8-10, 2003. Berlin: Springer, 2003: 173-191.

[4] Sushant S, Farnam J, Jignesh M. Patel. WIND: Workload-Aware INtrusion Detection [A] // Zamboni. Recent Advances in Intrusion Detection. 9th International Symposium Symposium on Recent Advances in Intrusion Detection [C]. Hamburg, Germany, September 20-22, 2006. Berlin: Springer, 2006: 290-309.

(上接第 8 页)

按钮可控制入射点在三棱镜左侧表面上任意移动。通过“$n1$ 增加”、“$n1$ 减小”、“$n2$ 增加”、“$n2$ 减小”四个交互按钮可控制三棱镜及三棱镜所在媒质的绝对折射率的大小。从模拟效果来看,通过对参数的交互控制,该仿真模型能很好地模拟出复色光通过三棱镜时的色散现象,具有很好的仿真效果。仿真模拟效果如图 3 所示。

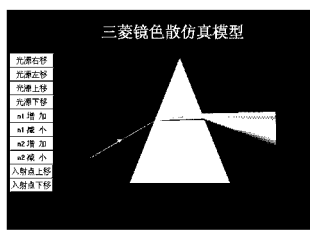

图 3 三棱镜色散仿真模型的模拟效果

5 结束语

三棱镜色散仿真模型利用数字化技术实现了对实物原型色散现象的仿真,利用该模型进行色散实验,可以不受实验环境条件的限制,并且可以任意设定三棱镜及所处媒质的折射率、复色光的入射角度和入射点的位置,从而拓展了实验的范围和空间。该仿真模型可以用于虚拟实验室进行远程网络虚拟实验。

参考文献

[1] 北京洪恩教育科技有限公司. Authorware 多媒体开发[M]. 北京:方圆电子音像出版社, 2007.

[2] 电脑报. Authorware 7.0[M]. 汕头:汕头大学出版社, 2006.

[3] 毕广吉. Authorware 多媒体开发程序设计[M]. 北京:人民邮电出版社, 2004