第16卷 第6期
2006年6月

计算机技术与发展
COMPUTER TECHNOLOGY AND DEVELOPMENT

Vol.16 No.6
Jun. 2006

基于异常的入侵检测技术在Snort系统中的应用

宋连涛,庄卫华
(河海大学 计算机与信息科学学院,江苏 南京 210098)

摘 要:分析了开放源代码的入侵检测系统——Snort的结构特征及其优缺点,并针对基于滥用的检测技术的缺点,研究了基于有效载荷,以网络请求类型、长度和位频率为建模对象的异常检测技术。设计了包含数据报处理、统计处理两层结构的系统模型。利用Snort系统的灵活性,以插件的形式应用到Snort系统中,从而提高其对未知入侵的识别能力。

关键词:网络安全;入侵检测;Snort;有效载荷;异常检测
中图分类号:TP393.08 **文献标识码**:A **文章编号**:1673-629X(2006)06-0136-03

Application of Anomaly Detection Technology in Snort System

SONG Lian-tao, ZHUANG Wei-hua
(College of Computer and Information Engineering, Hohai University, Nanjing 210098, China)

Abstract: Snort intrusion detection system is an open source IDS. This paper analyzes the structure, advantages and disadvantages of Snort, and then puts forward an anomaly detection technology based on network payload, request type, request length, and byte frequency to remedy the disadvantages of abuse detection technology used by Snort. This paper designs a system model with two levels of packages process level and statistical process level and plugs it into the Snort to enhance its ability of intrusion detection.

Key words: network security; intrusion detection; Snort; payload; anomaly detection

0 引 言

随着Internet网络安全问题的不断暴露和当前黑客攻击技术的日益进步,入侵检测技术成为网络安全研究的热点。入侵检测按其检测技术主要分为基于异常的入侵检测和基于滥用的入侵检测两种[1]。基于滥用的检测技术与防毒软件的工作原理相同,都是根据已知的入侵特征建立规则库,因此不能检测未知的入侵,且需要不断升级规则库。而基于异常的检测技术则可以弥补这个不足,但目前其技术还不够成熟。Snort系统是一个开放源代码的入侵检测系统,具有系统尺寸小、易于安装、便于配置、功能强大、使用灵活等诸多优点。

1 Snort系统的结构

整个Snort系统架构的着眼点在于强调性能、简洁和灵活性3个方面。总体上Snort系统由3个子系统构成[2,3],如图1所示:数据包解析器、检测引擎和日志/报警子系统。

Snort系统利用数据包解析器获取网络流中的数据信息,然后将数据提交检测引擎进行检测,最后根据检测结果决定是否调用日志/报警子系统进行警报。其工作原理是典型的基于滥用的检测,检测引擎的规则是根据现有入侵的特征人工进行定义的。这样,当面对新的入侵时就束手无策了,直到新的入侵的特征被提取,建立新的规则,Snort才能发挥作用。文中利用基于异常的检测技术来弥补这个缺陷,提高Snort应对新入侵的能力。

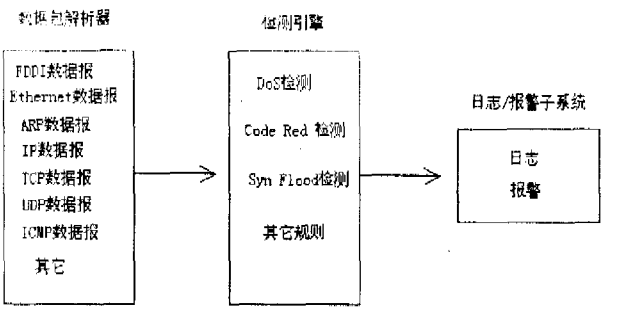

图1 Snort体系结构

2 基于异常的检测技术

与Snort的基于滥用的检测技术相反,基于异常的检测技术(Anomaly Detection)是对系统正常情况建模。检测分为培训(Training Phase)和检测(Detection Phase)两个阶段,在培训阶段以最近的历史数据为基础,建立被监视目标在正常情况下行为和状态的统计特征;在检测阶段,通过检测这些统计特征的当前值是否显著偏离了其相应的正常值来发现入侵。其关键是如何选取数据对正常情况建模。

收稿日期:2005-10-11
作者简介:宋连涛(1980-),男,吉林珲春人,硕士研究生,研究方向为网络安全;庄卫华,副教授,硕士生导师,研究方向为计算机应用技术。

网络入侵的形式多种多样,其中一些攻击利用协议的漏洞(如拒绝服务攻击利用TCP的3次握手),另外一些攻击对系统进行扫描、探测,一般可以通过分析网络数据报头,监测网络交通的连接尝试和会话行为进行检测,而最重要的攻击,如蠕虫攻击,其网络连接是正常的,只是包含异常的有效载荷来利用系统的漏洞,这些则可以通过检测数据报的有效载荷来发现。

目前有很多基于异常的入侵检测系统,如SPADE,NIDES,PHAD,ALAD[4]。这些系统提取的特征和计算正常模型的算法各不相同。大部分从数据报头提取特征,例如:SPADE,ALAD,NIDES对源、目的IP地址及端口号和TCP连接状态的分布建模。有的系统利用有效载荷的特征,只是利用方式很有限。经过大量实验,笔者选择对有效载荷的所有字节建模,然后利用统计学原理计算背离程度,这样检测效果会大大提高。

网络有效载荷是一个位流(Stream of Byte),不像网络数据报头,有效载荷没有确定的格式,每个字符都可能在载荷位流的任何位置出现,很难从中得到有用的信息,因此需要将有效载荷进行分类,这里选取服务类型作为分类标准。每种类型的服务都有各自的协议,因此载荷类型各不相同,例如:FTP命令传输的数据是加密的,各位应该均匀分布,而Telnet服务传输的数据则主要是可打印字符。

对于同一种服务,有效载荷的长度变化也很大,常用的TCP数据报的有效载荷长度从0到1460不等。不同的长度范围有着不同的载荷类型。根据以上分析,用有效载荷的以下特征对系统建模。

① 载荷类型(服务类型);

② 载荷长度;

③ 载荷的位分布。

总异常值(Anomaly Score) $AS = a * AS_{type} + b * AS_{len} + c * AS_{pd}$ ($a + b + c = 1$),其中,AS_{type}是载荷类型异常值,AS_{len}是载荷长度异常值,AS_{pd}是载荷位分布异常值。a,b,c是参数,以区分各个异常在总异常值中所占比重。

(1)载荷类型。

一般的服务由于长时间地被广泛使用,安全性很强,所以基于缓存区溢出或者非法输入的攻击都是利用那些很少被用户使用的服务。因此这种类型的服务请求含有异常的可能性高于其他请求,必须设置较高的异常值,定义p[type]为服务type发生的概率,p[type]∈[0,1],AS_{type}随着p[type]的增大而减小。文中取 $AS_{type} = -\log_2(p[type])$来计算载荷类型异常值,实验表明p[type]的最小值大约为 $3.05 * 10^{-5}$,即在一周的测试中,大约发生10次左右的请求。因此 $AS_{type\,max} = -\log_2(3.05 * 10^{-5}) = 15$,$AS_{type} \in [0,15]$。

(2)载荷长度。

载荷长度可以很好地标识出请求内容是否正确。请求一般包含协议信息和用户提供的输入,协议信息的长度很小,用户提供的请求输入一般也很少,而包含异常的请求长度就会大大增加。在此,借鉴Krugel的方法[5],取 $AS_{len} = 1.5^{(l-u)/(2.5*\sigma)}$,$u$为载荷长度的平均值,$\sigma$为标准差,均在培训期间计算得出。

(3)载荷的位分布。

系统以有效载荷中的字符为研究对象,进行建模。虽然各个ASCII字符的出现频率是不同的,但却有规律可循,尤其是将256个字符的相对频率按降序排列后,规律会更加明显。将排序的字符的相对频率定义为字符分布。对于字符串"Aaaza",其对应的ASCII值为"65 97 97 122 97",因此其字符分布如图2所示。

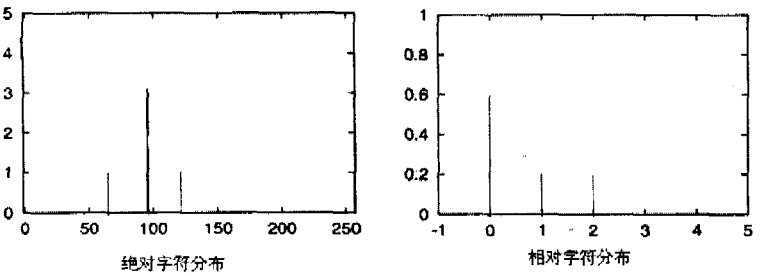

图2　字符分布

对于正常的有效载荷而言,递减排序的相对频率会沿x轴正方向缓慢下降。而在异常的负载中,如利用缓存区溢出获取shell的攻击中,由于要将生成shell代码的字符加入其中,因此频率会下降得很剧烈,或者几乎不下降。文中将正常有效载荷的理想的字符分布定义为载荷分布(Payload Distribution,PD),载荷分布是个离散分布,定义PD(n)表示第n大的频率($n=0$为最大),图2中,PD(0)=0.6,PD(1)=0.2,PD(2)=0.2。载荷分布在培训阶段计算得出,近似等于所有字符分布的平均值。具体求法如下:$pd(n)_i$($i = 0,1,2,\cdots,N$)为每个请求的字符分布,则 $PD(0) = [pd(0)_1 + pd(0)_2 + \cdots + pd(0)_N]/N$,$PD(1) = [pd(1)_1 + pd(1)_2 + pd(1)_3 + \cdots + pd(1)_N]/N$,……。对于检测阶段的每个请求,假设检测到的每个字符分布是从载荷分布中取出的一个样本,可以利用统计学中Pearson的χ^2统计量来决定检测样本与有效载荷分布之间的距离。当一个新的请求到达时,各个字符的绝对出现次数可以确定,将这些值按降序排列,然后进行如下测试,计算其与正常情况的距离:

① O_i为观测到的字符i的个数,E_i为期望的个数,$E_i = l * PD(i)$(l为载荷长度)。

② 计算 $\chi^2 = \sum[(O_i - E_i)^2/E_i]$($i \in [0,255]$)。

③ 查表获得偏移精度。根据查表数据与χ^2的关系,可以直接使用χ^2。

由于在相对分布相同的情况下,较长的载荷会得出较大的结果,由此将l做除数,使得AS_{pd}独立于载荷长度,同时乘上常数15以调整结果,使得AS_{pd}属于区间[0,15],得出:$AS_{pd} = \chi^2 * 15/l$。

(4)根据以上 3 个参数,确定异常值。

异常值 (Anomaly Score) $AS = 0.3 * AS_{type} + 0.3 * AS_{len} + 0.4 * AS_{pd}$ ($AS \in [0,15]$)

其中系数可据具体的网络环境自行调节,以区分 3 个参数在检测中的重要性。将计算的异常值与管理员设定的阈值进行比较,当超过阈值时,进行报警。

3 模型框架

系统将异常检测设计为一个两层的模型,包含数据报处理层和统计处理层,这样就不必为每种服务都编写全部代码。

(1)数据报处理层。

数据报处理层读取网络数据报,为统计处理层提供输入数据。由于是对有效载荷进行分析,所以不能直接在数据报层次上进行分析,因为攻击者可能将攻击分几个数据报发送。系统将以服务请求(A Server Request)为单位处理数据。对于 HTTP 请求,可能是 GET,HEAD,POST 包含 URL 或者用户使用的浏览器类型等信息的参数。DNS 请求可能只包含一个单独的含有要解析的名字的数据报。数据报处理层从数据报中提取服务请求,并把它们传入统计处理层。数据报处理分两个阶段进行:

① 第一阶段从网络数据中提取基本的 IP 和 TCP/UDP 数据报,将完整的 UDP 数据报或者是 TCP 流的有效载荷传给第二个阶段。Snort 系统已经完成这一功能。

② 第二个阶段从输入中提取各个完整的服务请求。通常,服务请求的终止可以通过观察请求终止字符或者字符序列(例如:两个 CTRL - LF 是 HTTP 的请求终止字符序列)或者通过检查请求头部的长度获得。请求的类型也可以通过一个字符序列或者一个头部值获得。

(2)统计处理层。

统计处理层根据请求的有效载荷和请求的类型对数据进行统计异常检测。根据公式:异常值 (AS) = $0.3 * AS_{type} + 0.3 * AS_{len} + 0.4 * AS_{pd}$,计算异常值。插入本模块后 Snort 系统的体系结构如图 3 所示。

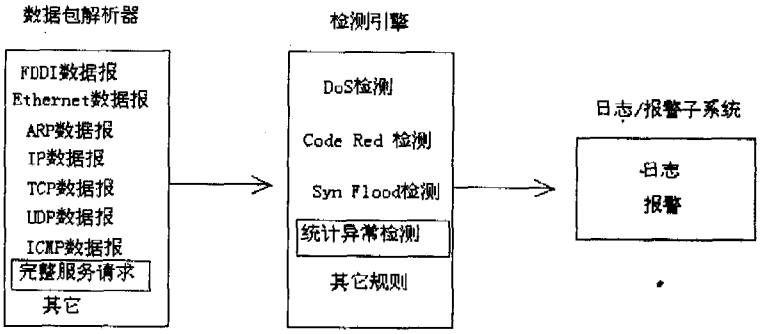

图 3 Snort 体系结构

4 结 论

系统在 Redhat9.0(Linux 2.4.20 - 8 内核)平台上,利用 Snort2.3.3 进行测试,结果如图 4 所示。

图 4 左是正常的 DNS 请求的字符分布,其字符频率平缓下降,而图 4 右是含有入侵的字符分布,可以发现其字符频率下降异常。

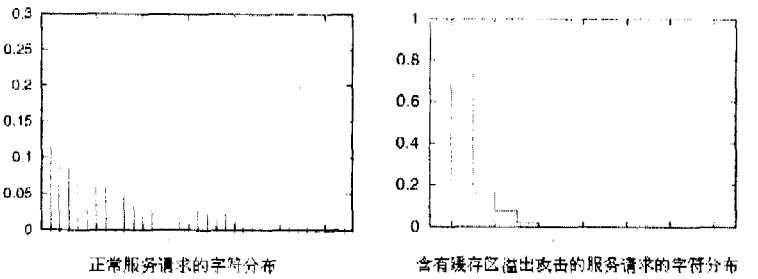

图 4 测试结果

利用 MIT 林肯实验室(MIT Lincoln Lab)的测试数据对本系统进行测试,结果在去掉 Snort 系统中关于 Code Red II 的相关检测规则后,Snort 只能检测到 95 个入侵中的 90 个入侵;将本系统加入 Snort 后,另外 5 个 Code Red II 入侵也可以被检测到。这表明本系统能够提高 Snort 对未知入侵的识别能力,可以有效地保障网络的安全。

参考文献:

[1] Denning D. An intrusion - detection model[A]. In IEEE Symposium on Security and Privacy[C]. Oakland, USA: IEEE,1986.118 - 131

[2] 唐正军. 黑客入侵防护系统源代码分析[M]. 北京:机械工业出版社,2002.

[3] 张 翔,张吉才,王 韬,等. 开放源代码入侵检测系统——Snort 的研究[J]. 计算机应用,2002,22(11):96 - 97.

[4] Wang Ke, Stolfo S J. Anomalous Payload - based Network Intrusion Detection[Z]. RAID, SpringerLink,2004.

[5] Kruegel C, Toth T, Kirda E. Service Specific Anomaly Detection for Network Intrusion Detection[A]. In Symposium on Applied Computing(SAC)[C]. Spain: ACM,2002.

(上接第 129 页)

速算法[J]. 上海交通大学学报,2002,36(4):555 - 558.

[2] 梁 旭,张 楠,黄 明,等. 一种新的高效关联规则数据挖掘算法[J]. 大连铁道学院学报,2001,22(1):60 - 63.

[3] 窦祥国,胡学钢. 关联规则的评价方法研究[J]. 安徽技术师范学院学报,2005,19(4):44 - 47.

[4] 陈文庆,许 棠. 关联规则挖掘 Apriori 算法的改进与实现[J]. 微机发展,2005,15(8):155 - 157.

[5] 吴芬兰,胡朝举,高 雅,等. 关联规则挖掘算法的改进[J]. 微机发展,2005,15(8):151 - 152.