基于应用层协议分析的应用层实时主动防御系统

谢柏林[1),2)] 余顺争[1)]

[1)](中山大学电子与通信工程系 广州 510006)

[2)](广东外语外贸大学思科信息学院 广州 510006)

摘 要 主动防御是当今网络安全研究领域的一个热点,现有的主动防御技术主要从网络层和传输层的角度来防御攻击.由于新出现的网络攻击主要发生在应用层,这类攻击在网络层和传输层的数据流与正常数据流没有显著区别,导致现有的主动防御技术无法有效应对这一类攻击,因此研究有效的应用层主动防御具有重要的意义.文中提出一种基于隐半马尔可夫模型的应用层风险实时评估方法,该方法通过分析网络中的实时数据流来评估应用层风险.基于上述风险实时评估方法和应用层协议分析,提出一种应用层实时主动防御系统,当系统发现用户的应用层协议行为存在风险时,该系统根据用户行为的风险值对其产生的数据包进行排队控制,自动纠正用户的异常行为,实现应用层主动防御.实验结果表明该系统具有良好的实时主动防御性能.

关键词 主动防御;应用层;风险评估;协议分析;隐半马尔可夫模型

中图法分类号 TP393 **DOI号**: 10.3724/SP.J.1016.2011.00452

Application Layer Real-time Proactive Defense System Based on Application Layer Protocol Analysis

XIE Bai-Lin[1),2)] YU Shun-Zheng[1)]

[1)](*Department of Electronics and Communication Engineering, Sun Yat-Sen University, Guangzhou* 510006)

[2)](*Cisco School of Informatics, Guangdong University of Foreign Studies, Guangzhou* 510006)

Abstract Proactive defense is a prevalent topic in current research field of network security. Existing proactive defense techniques mainly detect attacks from network layer and transport layer. Since most new attacks are based on application layer protocols and don't present significant difference in network traffic, it is difficult for existing proactive defense techniques to effectively detect such application layer attacks without special techniques. Therefore, the research on proactive defense of application layer becomes very important. This paper presents a risk real-time evaluation method for application layer based on hidden semi-Markov model. This method evaluates the application layer risk by analyzing network traffic. Based on this risk evaluation method and application layer protocol analysis, this paper presents a real-time proactive defense system for application layer. When user's behavior is at risk, the system queues the user's packets according to the risk indicator. By this means, the proposed system can automatically restrict each user's anomalous behavior, and achieve the application layer proactive defense. The final experiment results validate the performance of the system.

Keywords proactive defense; application layer; risk evaluation; protocol analysis; hidden semi-Markov model

收稿日期:2010-06-06;最终修改稿收到日期:2010-01-17.本课题得到国家自然科学基金-广东联合基金重点项目(U0735002)、国家自然科学基金(60970146,61070154)和国家"八六三"高技术研究发展计划项目基金(2007AA01Z449)资助.**谢柏林**,男,1982年生,博士研究生,主要研究方向为应用层异常检测、网络应用层行为分析.E-mail: xiebailin96@126.com. **余顺争**,男,1958年生,博士,教授,博士生导师,主要研究领域为信息安全、信号处理、无线网络.

1 引　言

主动防御是当今网络安全研究领域的一个热点.对网络的主动防御就是要能够有效地对付各种已知的网络攻击,特别是对付各种未知的、新颖的网络攻击.主动防御的战略目标是:"通过态势感知、风险评估、安全检测等手段来对当前安全情况进行判断,并依据判断结果来实施网络主动防御"[1].目前国内外学者主要从以下三个方面来研究主动防御:基于规则异常来研究主动防御;基于陷阱来研究主动防御,例如 Wang 等人[2]提出一种基于蜜罐网络的防火墙方案,该方案具有主动防御的功能;基于异常检测来研究主动防御.另外,姜伟等人[3]提出一种基于网络系统安全测评的主动防御模型以及基于该模型的一种最优主动防御策略选取算法;Deng 等人[4]提出一种基于云计算的分布式主动防御框架;Nguyen 等人[5]提出使用 k-NN 分类法(k-Nearest Neighbor classifier)来主动检测网络层 DDOS 攻击.现有的主动防御技术基本上都是通过分析网络层或传输层的数据来防御网络攻击,而忽略了应用层数据的信息.目前网络攻击越来越多地发生在应用层.根据市场研究公司 Gartner 最近预测:在未来的几年里 80% 的企业将成为应用层攻击的受害者.这些应用层攻击在网络层和传输层的数据流与正常用户产生的数据流通常没有显著的区别,例如应用层 DDOS 攻击[6].所以现有的主动防御技术不能有效防御这类攻击,因此研究有效的应用层主动防御具有重要的意义.

网络风险实时评估是实施网络主动防御的前提,要想实现对网络的主动防御,一般都需要有与其对应的网络风险实时评估方法,并且网络风险实时评估方法的性能直接影响主动防御的效果.目前国内外有不少学者正在研究如何对网络风险进行实时评估,他们提出了一些网络风险实时评估方法.例如 Arnes 等人[7]提出一种基于隐马尔可夫模型(Hidden Markov Model,HMM)的网络风险实时评估方法,该方法以入侵检测系统(IDS)的告警作为输入,采用 HMM 来评估整个网络所面临的风险程度,在该方法中 HMM 的状态数选为 4 个,分别为 Good、Probed、Attacked、Compromised,HMM 的模型参数则通过手工来设置.李伟明等人[8]对上述基于 HMM 的网络风险实时评估方法进行了一些改进和优化,缩小了 HMM 观测值矩阵的规模,并使用遗传算法来自动求解 HMM 的模型参数.陈秀真等人[9]提出一种层次化的网络风险实时评估方法,该方法基于 IDS 告警信息和网络性能指标来评估网络风险,并且分别从服务、主机及网络系统这三个层面来评估风险.上述几种网络风险实时评估方法均以 IDS 的告警作为输入,由于现有的 IDS 主要针对网络层和传输层上的已知攻击,因此上述几种评估方法只能评估已知攻击对网络造成的风险程度,而不能评估未知攻击、新颖攻击对网络造成的风险程度,也不能有效评估应用层攻击对网络造成的风险程度.另外王益丰等人[10]提出一种基于人工免疫的网络风险实时评估方法,该方法依据人体免疫系统中抗体浓度的变化与人体生病严重程度的关系来评估某种已知攻击对网络造成的风险程度.由于主动防御是要能够有效地对付各种网络攻击,特别是对付各种未知的、新颖的网络攻击,因此现有的网络风险实时评估方法不能满足主动防御的需求,更不能满足应用层主动防御的需求.

本文首次提出一种基于隐半马尔可夫模型(Hidden Semi-Markov Model,HSMM)[11-12]的应用层风险实时评估方法.该方法分为模型训练和风险评估两个阶段:在模型训练阶段,利用前后向算法训练得到正常用户在使用每种应用层协议时其行为的隐半马尔可夫模型;在风险评估阶段,在线统计每个观测序列相对于模型的平均对数或然概率,并根据平均对数或然概率值计算得到每个用户行为的风险值.本文提出的风险实时评估方法是基于异常检测来评估应用层风险,该方法能够评估应用层已知攻击对网络造成的风险程度,也能够评估应用层未知攻击和新颖攻击对网络造成的风险程度,因此该方法能够满足应用层主动防御的需求.本文提出的风险实时评估方法直接使用网络中的实时数据流来评估应用层风险,因此该方法具有很强的实时性.基于这种应用层风险实时评估方法,本文提出一种基于应用层协议分析的应用层实时主动防御系统.

本文第 2 节介绍系统的结构;第 3 节介绍系统中各个模块的原理,重点介绍应用层风险评估模块的原理;第 4 节给出实验结果;最后在第 5 节总结并讨论下一步研究工作.本文的创新点主要有以下两个:第一,提出一种基于隐半马尔可夫模型的应用层风险实时评估方法;第二,把这种应用层风险实时评估方法与现有的一些技术相结合,提出一种应用层实时主动防御系统.

2 系统结构

本文提出的应用层实时主动防御系统主要应用于网络边界的网关处，该系统由应用层协议识别模块、应用层协议关键词提取模块、应用层风险评估模块、控制模块四部分组成，该系统的结构如图 1 所示. 在网关处，过往的数据包依次经过应用层协议识别模块、应用层协议关键词提取模块、应用层风险评估模块、控制模块. 其中，应用层协议识别模块的功能是识别过往的数据包属于哪种应用层协议；应用层协议关键词提取模块的功能是提取数据包载荷中包含的应用层协议关键词；应用层风险评估模块的功能是对用户的应用层协议行为进行风险评估，并根据用户行为的风险值对其产生的数据包标记不同的 *mark* 值；控制模块的功能是根据数据包的 *mark* 值对数据包进行排队控制，在排队控制的过程中，让不同 *mark* 值的数据包进入不同的队列，而不同队列的带宽不同. 通过这种软控制方式可以自动纠正用户的异常行为，使得用户的行为与网络公众的共同行为特征相类似、且潜在行为无害，最终实现应用层主动防御.

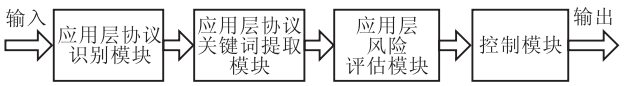

图 1 系统结构

3 各模块原理介绍

3.1 应用层协议识别模块

应用层协议识别模块采用开源软件 L7-filter[13] 的思想来识别应用层协议，即基于正则表达式(regular expression)来识别应用层协议. 在识别基于 TCP 的应用层协议时，首先缓存每个连接的前几个数据包，然后再重组这些数据包的应用层数据，最后对重组后的应用层数据进行协议正则表达式的匹配. 这样一旦识别出某个连接所属的应用层协议，则对于后续的数据包，只需检查其是否属于该连接，如果属于该连接，那么这些数据包就一定属于相同的应用层协议，而无需另外进行协议正则表达式的匹配. 在识别基于 UDP 的应用层协议时，直接对每个数据包的载荷进行协议正则表达式的匹配. 采用上述方法识别应用层协议，其准确度高、速度快[14].

3.2 应用层协议关键词提取模块

在识别出数据包所属应用层协议的基础上，通过应用层协议关键词提取模块来提取数据包载荷中所包含的应用层协议关键词以及记录这些关键词到达网关时的时间. 所谓的应用层协议关键词(keywords)是指能反映用户在使用该协议时其行为的词. 例如用户在使用 HTTP 协议时，为了描述用户的应用层行为，该协议的关键词可选为 GET、POST、HAED、100、200、304、404 等；而当用户在使用 SMTP(Simple Mail Transfer Protocol)协议时，该协议的关键词由 HELO、MAIL FROM、RCPT TO、QUIT、VFRY、DATA、REST、NOOP、EXPN、HELP 和一些状态码组成. 在协议识别的基础上提取关键词，是为了减少每次提取关键词时所需匹配的关键词的数目，从而提高关键词提取的速度.

3.3 应用层风险评估模块

应用层风险评估模块是从应用层协议的角度来评估每个用户在使用每种应用层协议时其行为所存在的风险程度. 从应用层协议的角度来看，一个用户在使用某种应用层协议时，其在一段时间里的行为反映到应用层协议上就是一系列应用层协议关键词之间的交互. 假设某种应用层协议具有 K 个关键词，即 $word_1, word_2, \cdots, word_K$，并分别数字化为 $1, 2, \cdots, K$，则某个用户在使用该协议时，其在一段时间里的行为反映到应用层协议上就是由这 K 个关键词组成的一个关键词序列. 例如图 2 为用 HTTP 协议关键词序列表示的单个用户与 Web 服务器之间的通信过程，其中该用户的 HTTP 协议关键词序列为 GET，304，GET，200，….

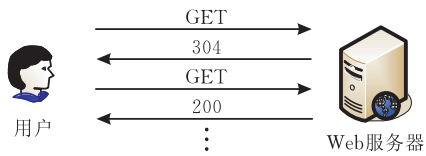

图 2 HTTP 协议关键词序列

大部分正常用户在使用某种应用层协议时，其产生的关键词序列的统计特征具有一定的相似性. 例如用户在使用 HTTP 协议时，由于正常用户的点击速度、浏览时间、浏览过程等具有一定的相似性，导致在正常用户产生的关键词序列中每个关键词出现的频率和关键词之间的时间间隔具有一定的相似性. 可以把这种统计特征作为正常用户的模型，而异常用户产生的关键词序列的统计特征与该统计特征具有比较大的差异. 例如当某个僵尸主机(Bot)发起 HTTP 请求泛洪攻击(HTTP request flood attack[15])时，由于其发送网页请求的频率远

大于正常用户的频率，从而导致在僵尸主机产生的关键词序列中，关键词“GET”出现的频率非常高，而其它关键词出现的频率很低，并且关键词之间的时间间隔很小.

正常用户在使用某种应用层协议时其行为会发生变化，这会导致其产生的关键词的统计分布会有所不同. 例如正常用户在使用 HTTP 协议时，当正常用户分别处于浏览网页、在线看电影、在线购物时，在其产生的关键词序列中每个关键词出现的频率和关键词之间的时间间隔是不一样的. 我们可以把正常用户在使用某种应用层协议时的不同行为表现称为状态. 假设正常用户在使用某种应用层协议时具有 M 个离散状态，分别表示为 $s_1, s_2, \cdots, s_M$，并记这些状态的集合为 S，其中根据以往的经验，M 取值一般在 4～10 之间. 用户状态的变化就表现为应用层协议关键词序列本身或者其统计特征的变化. 假设用户状态的变化过程可以看作是一个马尔可夫过程，即用户的当前状态只与前一个状态有关，则正常用户在使用某种应用层协议时其状态转移关系可以用一个具有 M 个状态的马尔可夫链来描述，令 $\boldsymbol{A}=\{a_{mn}; 1\leq m,n\leq M\}$ 为状态转移概率矩阵，它的元素 a_{mn} 表示正常用户在使用某种应用层协议时其状态从 s_m 跳转到 s_n 的概率，$s_m, s_n \in S$.

当某个用户在使用某种应用层协议时，令 o_t 表示应用层协议关键词提取模块检测到的该用户的第 t 个观测量，它包括到达网关的第 t 个关键词 w_t 和该关键词与前一个关键词之间的时间间隔 r_t，即 $o_t=(w_t, r_t)$. 其中，$w_t \in \{1,2,\cdots,K\}$；r_t 取的是离散化的整数值，即 $r_t \in \{0,1,2,3,\cdots\}$，在本系统中 r_t 的时间单位选为秒. 令 $O=o_1, o_2, \cdots, o_T=o_1^T$ 表示某个用户在使用某种应用层协议时产生的一个长度为 T 的二维观测序列，我们在统计用户观测序列时使用 IP 地址作为用户区分的标志. 由于关键词之间的时间间隔 r_t 主要与主机的请求、响应处理时间和网络的传输时延有关，而关键词 w_t 只与协议有关、与主机处理时间和网络时延无关，因此可以近似假定：在给定的用户状态下 w_t 和 r_t 相互统计独立. 令 $\boldsymbol{B}=\{b_m(k,g); 1\leq k\leq K, 0\leq g, 1\leq m\leq M\}$ 为观测值概率矩阵，它的元素 $b_m(k,g)$ 表示正常用户在使用某种应用层协议时其在状态 s_m 下产生观测量 $w_t=k$，$r_t=g$ 的概率. $b_m(k,g)$ 可表示为

$$\begin{aligned} b_m(k,g) &= Pr[w_t=k, r_t=g \mid q_t=s_m] \\ &= Pr[w_t=k \mid q_t=s_m] \times Pr[r_t=g \mid q_t=s_m] \\ &= b_m(k) \times b_m(g) \end{aligned} \tag{1}$$

其中，q_t 表示第 t 个观测量到达网关时用户所处的状态，且满足 $\sum_k b_m(k)=1$ 和 $\sum_g b_m(g)=1$.

正常用户在使用某种应用层协议时，在其产生的二维观测序列中，关键词、关键词之间的时间间隔与用户状态都不具有一一对应的关系. 例如正常用户在使用 HTTP 协议时，在其产生的二维观测序列中关键词“GET”和“200”并不能直接说明用户正处于什么状态. 所以正常用户在使用某种应用层协议时，其状态转移过程是一个隐马尔可夫过程.

令 $\boldsymbol{\pi}=\{\pi_m; 1\leq m\leq M\}$ 为初始状态概率矩阵，它的元素 π_m 表示正常用户在使用某种应用层协议时，当用户的第一个观测量到达网关时其状态为 s_m 的概率. 令 $\boldsymbol{P}=\{p_m(d); 1\leq m\leq M, 1\leq d\leq D\}$ 为状态持续时间概率矩阵，它的元素 $p_m(d)$ 表示正常用户在使用某种应用层协议时其在状态 s_m 下连续产生 d 个观测量的概率，其中 D 为状态逗留的最大时间，且满足 $\sum_d p_m(d)=1$. 根据网络应用的不同和网络负载的变化，$p_m(d)$ 可能是一个比较复杂的分布，不一定是指数分布或几何分布. 而在隐马尔可夫模型中，$p_m(d)$ 必须服从指数分布或几何分布. 因此正常用户在使用某种应用层协议时，其状态转移过程实际上可以看作是一个隐半马尔可夫过程. 令 $\lambda=\{\boldsymbol{A},\boldsymbol{B},\boldsymbol{\pi},\boldsymbol{P}\}$ 为隐半马尔可夫模型的参数集.

3.3.1　模型训练

在网关处采集正常用户在使用某种应用层协议时产生的大量观测序列作为模型训练集. 假设训练集为 Φ，包含 L 个不同观测序列，即 $\Phi=\{O^{(l)}; l=1,2,\cdots,L\}$，其中 $O^{(l)}=o_1^{T_l}$ 为第 l 个观测序列，T_l 为对应序列的长度，并且假定这些观测序列服从同一个隐半马尔可夫模型且相互独立. 我们运用文献[12]中提出的前后向算法，采用多序列来训练模型参数[11]，模型训练的步骤如下：

1. 对模型参数集合 λ 赋初值.

2. 运用前向变量递推算法，求每个观测序列 $O^{(l)}(1\leq l\leq L)$ 的前向变量集 $\{\alpha_t^{(l)}(m,d); 1\leq t\leq T_l\}$.

3. 根据式(2)计算训练集中每个观测序列相对于模型的或然概率 $P_l(1\leq l\leq L)$. 然后再根据式(3)计算该训练集相对于模型的总或然概率 $P(\Phi|\lambda)$.

$$P_l=P(O^{(l)}|\lambda)=\sum_{m=1}^{M}\sum_{d=1}^{D}\alpha_{T_l}^{(l)}(m,d) \tag{2}$$

$$P(\Phi|\lambda)=\prod_{l=1}^{L}P(O^{(l)}|\lambda)=\prod_{l=1}^{L}P_l \tag{3}$$

4. 运用后向变量递推算法，求每个观测序列 $O^{(l)}(1\leq l\leq L)$ 的后向变量集 $\{\beta_t^{(l)}(m,d); 1\leq t\leq T_l\}$.

5. 由前向变量集$\{\alpha_t^{(l)}(m,d);1\leq t\leq T_l\}$和后向变量集$\{\beta_t^{(l)}(m,d);1\leq t\leq T_l\}$计算状态跳转联合概率集$\{\zeta_t^{(l)}(m,n);1\leq t\leq T_l\}$、状态持续联合概率集$\{\eta_t^{(l)}(m,d);1\leq t\leq T_l\}$、状态和观测值联合概率集$\{\gamma_t^{(l)}(m);1\leq t\leq T_l\}$.

6. 运用式(4)～(8)更新模型参数集合λ的值. 在式(7)中,如果$w_t^{(l)}=k$,则$\delta(w_t^{(l)}-k)=1$,否则$\delta(w_t^{(l)}-k)=0$. 在式(8)中,如果$r_t^{(l)}=g$,则$\delta(r_t^{(l)}-g)=1$,否则$\delta(r_t^{(l)}-g)=0$.

$$\hat{a}_{mn}=\frac{\sum_{l=1}^{L}\frac{1}{P_l}\sum_{t=2}^{T_l}\zeta_t^{(l)}(m,n)}{\sum_{l=1}^{L}\frac{1}{P_l}\sum_{n=1}^{M}\sum_{t=2}^{T_l}\zeta_t^{(l)}(m,n)} \tag{4}$$

$$\hat{\pi}_m=\frac{\sum_{l=1}^{L}\frac{1}{P_l}\gamma_1^{(l)}(m)}{\sum_{l=1}^{L}\frac{1}{P_l}\sum_{m=1}^{M}\gamma_1^{(l)}(m)} \tag{5}$$

$$\hat{p}_m(d)=\frac{\sum_{l=1}^{L}\frac{1}{P_l}\sum_{t=1}^{T_l}\eta_t^{(l)}(m,d)}{\sum_{l=1}^{L}\frac{1}{P_l}\sum_{d=1}^{D}\sum_{t=1}^{T_l}\eta_t^{(l)}(m,d)} \tag{6}$$

$$\hat{b}_m(k)=\frac{\sum_{l=1}^{L}\frac{1}{P_l}\sum_{t=1}^{T_l}\gamma_t^{(l)}(m)\delta(w_t^{(l)}-k)}{\sum_{l=1}^{L}\frac{1}{P_l}\sum_{k=1}^{K}\sum_{t=1}^{T_l}\gamma_t^{(l)}(m)\delta(w_t^{(l)}-k)} \tag{7}$$

$$\hat{b}_m(g)=\frac{\sum_{l=1}^{L}\frac{1}{P_l}\sum_{t=1}^{T_l}\gamma_t^{(l)}(m)\delta(r_t^{(l)}-g)}{\sum_{l=1}^{L}\frac{1}{P_l}\sum_{g}\sum_{t=1}^{T_l}\gamma_t^{(l)}(m)\delta(r_t^{(l)}-g)} \tag{8}$$

7. 判断在步3中求出的$P(\Phi|\lambda)$是否收敛到一个固定的值. 如果已经收敛,则退出模型训练,得到模型参数集λ,否则重复步2～7,对模型参数继续进行迭代估计.

在得到该HSMM的参数集合λ后,计算训练集中每个观测序列$O^{(l)}$($1\leq l\leq L$)相对于模型的平均对数或然概率E_l,E_l的定义如式(9)所示.

$$E_l=\frac{1}{T_l}\ln(P(O^{(l)}|\lambda)) \tag{9}$$

最后计算训练集中所有观测序列平均对数或然概率的平均值μ及标准差σ,μ和σ的计算公式分别如式(10)、(11)所示.

$$\mu=\frac{\sum_{l=1}^{L}E_l}{L} \tag{10}$$

$$\sigma=\sqrt{\frac{\sum_{l=1}^{L}(E_l-\mu)^2}{L}} \tag{11}$$

3.3.2 风险评估

在训练得到正常用户某种应用层协议行为的隐半马尔可夫模型后,就可以利用该模型对用户在使用该协议时的行为进行风险评估,并根据用户行为的风险值对其产生的数据包标记不同的*mark*值,其中$mark\in\{0,1,2\}$. 如果$mark=0$,则表示用户行为正常;如果$mark=1$,则表示用户行为稍微异常;如果$mark=2$,则表示用户行为很不正常. 为了实现风险评估,首先需要定义两个门限值,一个是观测序列长度门限值H,另外一个是最大间隔时间ΔT. 在评估过程中,只有当用户的观测序列长度大于H时,才对该序列进行风险评估. 这是因为过短的序列不能很好地反映用户的行为,并且少量的异常短序列也不会对网络安全造成威胁. 虽然大量的异常短序列会对网络安全造成威胁,但使用一些传统的方法对这些短序列的汇聚流进行检测,就可以根据汇聚流量的行为表现把这种威胁检测出来,并通过限速、过滤或负载均衡等措施来化解它. 例如可以使用网关处关键词的速率来检测这种威胁. 在网关处,如果在超过ΔT的时间段里没有检测到某个用户的观测量,那么就重新统计该用户的观测序列,这样使得观测序列能更好地体现用户的当前行为. 根据实验结果,我们选取$H=20$,$\Delta T=1800$s. 当某个用户在使用某种应用层协议时,按照以下步骤对该用户在使用该协议时的行为进行风险评估:

1. 当应用层协议关键词提取模块检测到该用户的第一个关键词时,记录下该关键词w_1和该关键词到达网关时的时间ν_0,并令$t=1$,$r_1=0$. 然后把该数据包的*mark*值设置为零,执行步2.

2. 在当前时刻,如果应用层协议识别模块检测到该用户在使用该协议时产生的数据包,但应用层协议关键词提取模块未检测到关键词,则把该数据包的*mark*值设置成为与该用户上一个数据包的*mark*值相同,回到步2;如果应用层协议关键词提取模块检测到关键词,则跳转到步3.

3. 令$t=t+1$,并记录下该关键词w_t和该关键词到达网关时的时间ν,然后计算出该关键词与上一个关键词之间的时间间隔$r_t=\nu-\nu_0$,最后令$\nu_0=\nu$. 如果$r_t\geq\Delta T$,则把该数据包的*mark*值设置为零,并跳转到步1;如果$r_t<\Delta T$,则执行步4.

4. 由最新的观测量$o_t=(w_t,r_t)$和前一个前向变量$\alpha_{t-1}(m,d)$迭代计算出当前的前向变量$\alpha_t(m,d)$. 如果$t<H$,则把该数据包的*mark*值设置为零,跳转到步2;如果$t\geq H$,则根据式(2)和(9)计算出观测序列o_1^t的平均对数或然概率E,并执行步5.

5. 计算$\delta=\frac{|E-\mu|}{\sigma}$,其中$\mu$,$\sigma$分别为模型训练阶段得到的训练集中所有观测序列平均对数或然概率的平均值及标准差. 根据δ的取值来设置该数据包的*mark*值,跳转到步2.

在上述循环过程中，δ 的取值反映了用户在使用这种协议时其行为所存在的风险程度. 其中 δ 的取值越大表示用户的行为越异常，如果 δ 的取值为零，则表示用户的行为很正常. 我们把 δ 称为用户行为的风险值，并且把 δ 的取值范围划分为三个区间，δ 在这三个区间的取值分别表示用户行为正常、稍微异常、很不正常.

为了提高模型的准确度，我们在应用层风险评估模块中增加了模型参数在线更新的功能，即通过在线采集正常的观测序列，然后每隔一定的时间就重新训练模型参数，以便使模型能更好地刻画正常用户的行为特征. 在本系统中，我们每隔两个小时就重新训练模型参数. 应用层风险评估模块的结构如图 3 所示. 应用层风险评估模块是基于异常检测来评估应用层风险，该模块能够评估应用层未知攻击和新颖攻击对网络造成的风险程度.

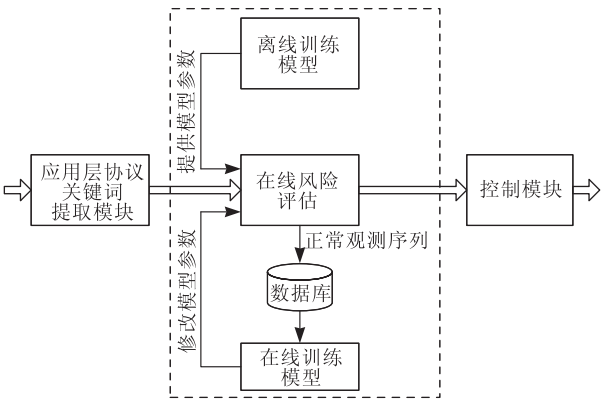

图 3　应用层风险评估模块的结构

3.4　控制模块

控制模块根据每个数据包的 *mark* 值对数据包进行排队控制，即按照 *mark* 值的不同而让数据包进入不同的队列进行排队控制. 在每个队列中采用随机及早检测方法(Random Early Detection，RED)[16]对数据包进行排队控制. 我们给每个队列分配不同的带宽，其中，$mark=0$ 的数据包所在队列(队列 1)的带宽最大，以保证该队列中的所有数据包都能顺利通过网关；$mark=1$ 的数据包所在队列(队列 2)的带宽比较小，在该队列中由于带宽比较小的原因，很多数据包将被迫延时等待；$mark=2$ 的数据包所在队列(队列 3)的带宽最小，在该队列中绝大部分的数据包都有可能被丢弃. 在同一队列中，每个数据包的优先级都相同，采用先入先出的排队方式. 控制模块的流程如图 4 所示.

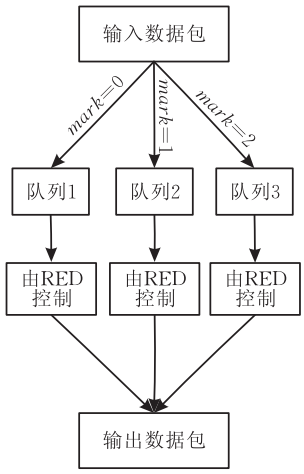

图 4　控制模块的流程图

4　实验测试及结果分析

本文提出的应用层实时主动防御系统是在 Linux 的 Netfilter 框架下实现的，该系统的一些模块通过扩展、移植一些 Linux 开源软件来实现. 例如：应用层协议识别模块是通过扩展、移植 L7-filter 源码来实现，控制模块是通过扩展、移植 Iproute2 里面的 TC 源码[17]来实现. 为了验证本系统的性能，我们对本系统进行了一些测试. 实验测试分为离线测试、在线测试和系统对比，其中离线测试是为了测试应用层风险评估模块对应用层攻击的识别能力，在线测试是为了测试整个系统的在线性能.

4.1　离线测试

应用层风险评估模块在很大程度上决定了本系统的性能，我们对该模块进行了离线测试，以测试本文提出的应用层风险实时评估方法对应用层攻击的识别能力. 我们使用 1999 DARPA 数据集[18]中的 HTTP、SMTP、FTP、Telnet 这 4 种应用层协议的数据对本文提出的应用层风险实时评估方法进行测试. 1999 DARPA 数据集是目前公开的唯一一个拥有完整数据包载荷的标准测试数据集，该数据集持续的时间为五周，其中 *week*1、*week*3 不包含任何攻击，我们使用的数据为该数据集中的 inside tcpdump data. 在 1999 DARPA 数据集中，与 HTTP 协议相关的攻击有 Apache2、Back；与 SMTP 协议相关的攻击有 Mailbomb；与 FTP 协议相关的攻击有 Ftpwrite、Guessftp；与 Telnet 协议相关的攻击有 Guesstelnet. 我们从 *week*1、*week*3 中提取出 5000 个 HTTP 观测序列、1000 个 SMTP 观测序列、1000 个 FTP 观测序列、1000 个 Telnet 观测序列，作为正

常用户分别在使用这 4 种协议时其行为模型的训练集.另外从 *week*2、*week*4、*week*5 中提取出 300 个 Apache2攻击产生的观测序列、12 个 Back 攻击产生的观测序列、200 个 Mailbomb 攻击产生的观测序列、11 个 Guessftp 攻击产生的观测序列、7 个 Ftpwrite攻击产生的观测序列、6 个 Guesstelnet 攻击产生的观测序列,作为模型的异常测试集.

在模型训练中,HTTP 协议关键词选为 GET、User-Agent、HEAD、POST、PUT、DELETE、TRACE 以及 HTTP/1.1 的 41 个响应码;SMTP 协议关键词选为 HELO、MAIL FROM、RCPT TO、DATA、REST、NOOP、QUIT、VRFY、EXPN、HELP 以及该协议的 20 个响应码;FTP 协议关键词选为 DELE、HELP、LIST、NOOP、PASS、PORT、CWD、QUIT、REST、RETR、SITE、SMNT、STOU、STRU、NLST、RNTO、RNFR、SYST、TYPE、USER 以及该协议常出现的 39 个响应码;Telnet 协议关键词选为 Abort、Break、Command、Data、DO、DON'T、Echo、Erase character、Erase Line、Go ahead、IAC、Interrupt Process、NOP、SB、SE、Suboption Begin、Suboption End、WILL、WON'T、login、password.另外令正常用户分别在使用 HTTP、SMTP、FTP、Telnet 这 4 种协议时其初始状态数都为 20 个,然后在模型训练中删除那些很少出现的状态,最后得到正常用户分别在使用这 4 种协议时其最终状态数.

模型训练结束后,我们在 HTTP、SMTP、FTP、Telnet 这 4 种协议的训练集和异常测试集中,分别计算出每个观测序列相对于各自模型的风险值.图 5 为 HTTP 观测序列风险值的直方图分布.Apache2攻击发送的 GET 请求中包含大量的关键词"User-Agent",导致在其产生的观测序列中关键词之间的时间间隔比较小,并且关键词"User-Agent"出现的频率比较大,因此其产生的观测序列的风险值比正常观测序列要大一些.Back 攻击发送的 GET 请求中包含大量的"/",这会造成服务器响应速度比较慢,并且很多 GET 请求没有对应的响应码,所以其产生的观测序列的风险值比正常观测序列要大很多.图 6 为 HTTP 正常观测序列风险值的门限值与检测率(Detection Ratio,DR)、误报率(False Positive Ratio,FPR)的关系.从图 6 可知,当门限值选为 2.04 时,模型对 Apache2 攻击产生的观测序列的识别率为 99.6%,误报率为 0.9%,模型对 Back 攻击产生的观测序列的识别率为 100%.

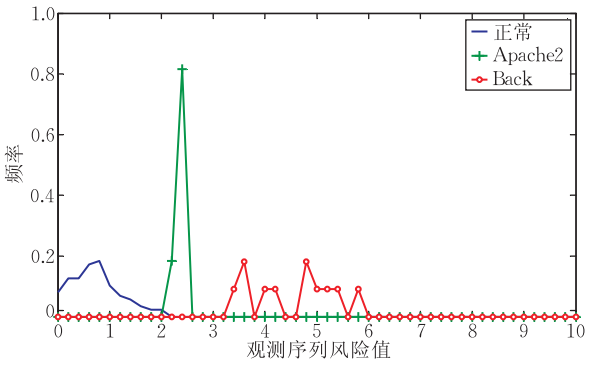

图 5 HTTP 风险值的直方图分布

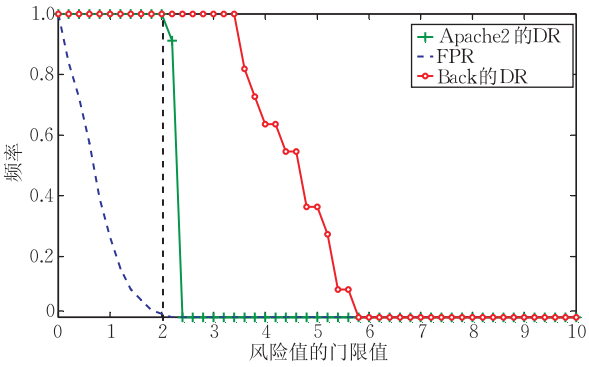

图 6 HTTP 风险值的门限值与检测率、误报率

SMTP 观测序列风险值的直方图分布如图 7 所示.Mailbomb 攻击是通过发送大量的垃圾邮件来攻击服务器,在其产生的观测序列中关键词之间的时间间隔比较小,在相同的时间段里其产生的观测序列比较长,因此其产生的观测序列的风险值比正常观测序列要大一些.图 8 为 SMTP 正常观测序列风险值的门限值与检测率、误报率的关系.从图 8 可知,当门限值选为 2.01 时,模型对 Mailbomb 攻击产生的观测序列的识别率为 99.5%,误报率为 1.1%.

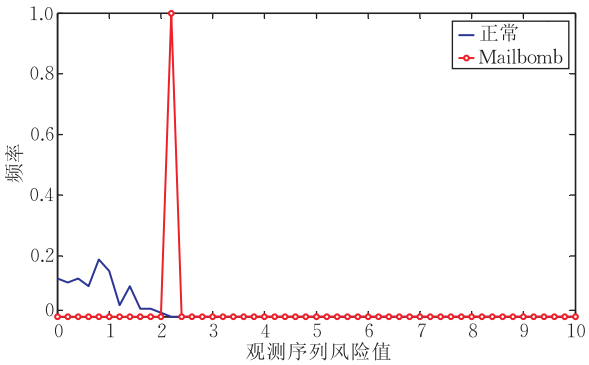

图 7 SMTP 风险值的直方图分布

FTP 观测序列风险值的直方图分布如图 9 所示.Guessftp 攻击是通过多次尝试不同的用户名和密码来试图登录到 FTP 服务器,在其产生的观测序列中关键词"USER"、"PASS"出现的次数比较多,

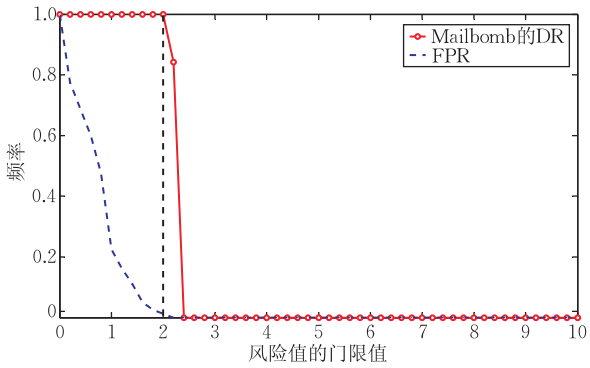

图 8　SMTP 风险值的门限值与检测率、误报率

因此其产生的观测序列的风险值比正常观测序列要大一些. Ftpwrite 攻击是通过在根目录下创建文件来实现，而在训练数据集中用户创建文件发生的次数很少，所以其产生的观测序列的风险值比正常观测序列要大很多. FTP 正常观测序列风险值的门限值与检测率、误报率的关系如图 10 所示. 从图 10 可知，当门限值选为 2.79 时，模型对 Guessftp 攻击产生的观测序列的识别率为 100%，误报率为 0.8%；模型对 Ftpwrite 攻击产生的观测序列的识别率为 100%.

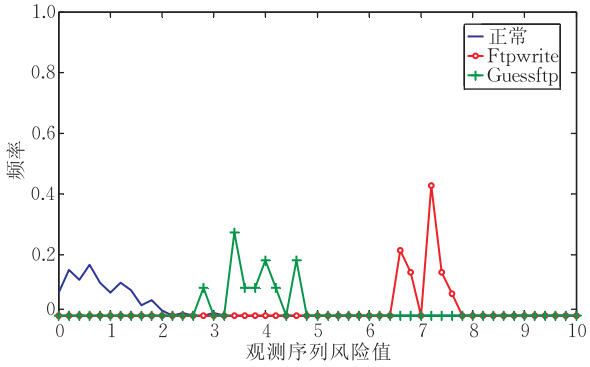

图 9　FTP 风险值的直方图分布

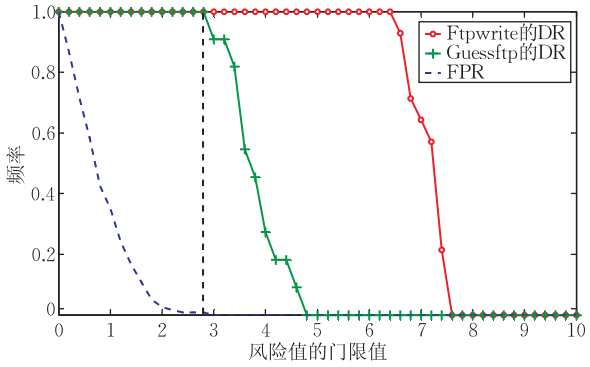

图 10　FTP 风险值的门限值与检测率、误报率

Telnet 观测序列风险值的直方图分布如图 11 所示. 在图 11 中，由于 Guesstelnet 攻击是通过多次尝试不同的用户名和密码来试图登录到 Telnet 服务器，所以在其产生的观测序列中关键词“login”、“password”出现的次数比较多，因此其产生的观测序列的风险值比正常观测序列要大一些. Telnet 正常观测序列风险值的门限值与检测率、误报率的关系如图 12 所示. 从图 12 可知，当门限值选为 2.38 时，模型对 Guesstelnet 攻击产生的观测序列的识别率为 100%，误报率为 0.1%.

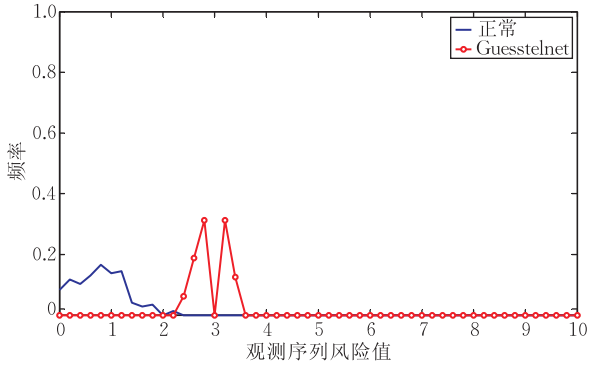

图 11　Telnet 风险值的直方图分布

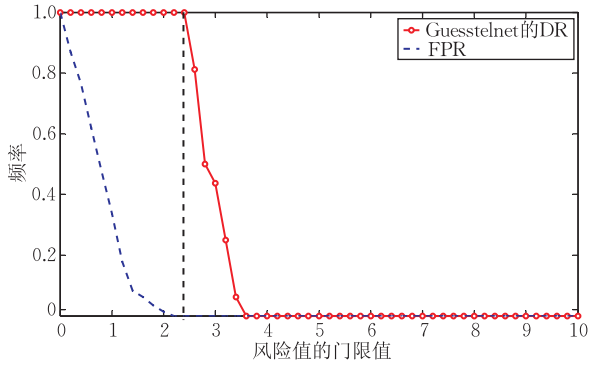

图 12　Telnet 风险值的门限值与检测率、误报率

从上面的测试实验可知，本文提出的应用层风险实时评估方法在检测应用层攻击时具有很高的检测率和较低的误报率. HTTP、SMTP、FTP、Telnet 这 4 种协议正常观测序列的风险值绝大部分都落在区间[0,2]，当我们把这 4 种协议正常观测序列风险值的门限值都选为 2 时，本文提出的风险实时评估方法对上述几种应用层攻击产生的观测序列的检测率和误报率如表 1 所示.

表 1　应用层风险实时评估方法的检测率与误报率

应用层攻击	检测率/%	误报率/%
Apache2	99.6	1.2
Back	100	1.2
Mailbomb	99.5	1.1
Ftpwrite	100	2.3
Guessftp	100	2.3
Guesstelnet	100	1.5

4.2 在线测试

为了测试本系统的在线性能，我们在中山大学一个实际网络的出入口对本系统进行了在线测试，测试实验的拓扑如图 13 所示.

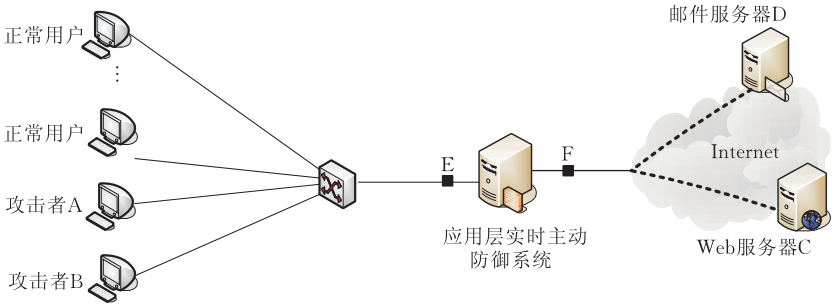

图 13 在线测试实验拓扑

在图 13 中，应用层实时主动防御系统所在那台主机作为网关，其配置为 CPU：Intel Core^2 Q9550（四核心，主频：2.83GHz），内存：4GB，硬盘：1T，操作系统：Fedora Core 10. 网关左边的主机数在 120～150 之间，其中两台主机（攻击者 A、攻击者 B）作为攻击源，以产生应用层攻击数据流. 在网关处，我们分配给队列 1 的带宽为 50Mbps，分配给队列 2 的带宽为 8Kpbs，分配给队列 3 的带宽为 1Kpbs，以便保证正常的数据流能顺利通过网关，而异常的数据流则能得到较好的控制.

在图 13 所示的网络中，我们使用所有流经网关的 HTTP、SMTP 的数据来测试本系统. 之所以只选择 HTTP 和 SMTP，这是因为在网络上很难找到利用其它应用层协议进行网络攻击的软件，即很难产生与其它应用层协议相关的真实攻击流.

为了测试本系统的在线性能，首先在网关处采集正常用户分别在使用 HTTP、SMTP 时产生的大量观测序列来训练模型. 在训练得到这两种模型后，我们分两种情况来测试本系统的在线性能：一种情况是网络中不存在应用层攻击；另外一种情况是网络中存在应用层攻击. 在测试过程中，HTTP、SMTP 正常观测序列风险值的门限值都选为 2. 整个在线测试实验的持续时间为 90min，其中在前 30min 的测试中网络不存在应用层攻击. 在第 30min 同时加入分别针对 HTTP、SMTP 的攻击，即让攻击者 A 利用一些攻击软件向 Web 服务器 C 发起 HTTP 请求泛洪攻击（其发送页面请求的速率为每秒钟 120 个）、攻击者 B 利用一些邮件群发软件向邮件服务器 D 发送比较多的垃圾邮件（每分钟发送 40 份垃圾邮件），并让这两种攻击持续 30min. 在最后 30min 的测试中网络不存在应用层攻击.

在测试过程中，我们在网络拓扑中的 E 点和 F 点对 HTTP、SMTP 的数据进行采集. 在线测试实验结束后，通过分析采集的数据计算本系统在上述 3 个不同测试阶段的检测率和误报率. 系统检测率定义为：系统检测到的应用层攻击数据包的个数/应用层攻击数据包的总个数；系统误报率定义为：系统把正常数据包误判为攻击数据包的个数/正常数据包的总个数. 本系统在上述 3 个不同测试阶段的检测率和误报率如表 2 所示.

表 2 系统的检测率与误报率

阶段	系统检测率/%	系统误报率/%
阶段 1（0m～30m）	N/A	0.5
阶段 2（30m～60m）	99.7	0.8
阶段 3（60m～90m）	N/A	0.6

从表 2 可知，当在线运行本系统时，本系统具有很高的检测率和较低的误报率. 由于分配给异常数据流的带宽很小，所以当攻击者 A、B 同时发起分别针对 HTTP、SMTP 的攻击时，在其产生的数据包中 99.6%的数据包被丢弃. 因此本系统具有很好的应用层实时主动防御效果.

为了测试本系统自身的抗攻击能力，我们在测试过程中每隔 3min 就记录一下本系统对主机 CPU、内存的使用情况. 本系统对主机 CPU、内存的使用曲线分别如图 14、图 15 所示.

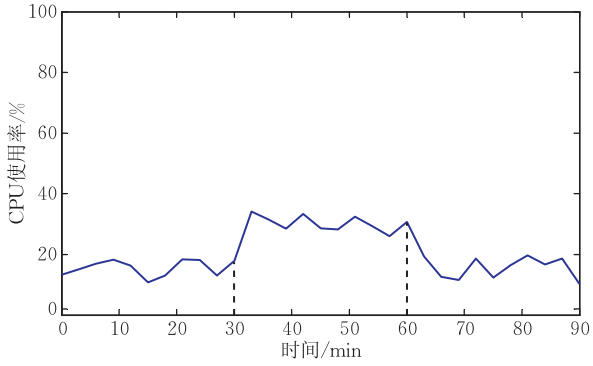

图 14 CPU 使用曲线

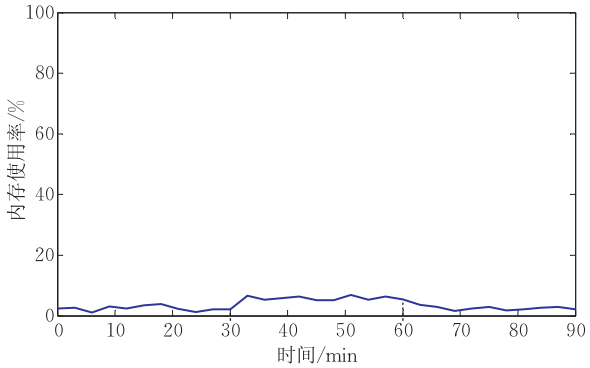

图 15　内存使用曲线

从图 14、图 15 可知，当网络中不存在应用层攻击时，本系统对 CPU 的消耗率在 11%～19%；对内存的消耗率在 1%～3.8%. 此时，网关处网络总流量的平均速度为 18Mbps，HTTP 流量的平均速度为 6Mbps，SMTP 流量的平均速度为 12Kbps. 当网络中存在应用层攻击时，本系统对 CPU 的消耗率在 28%～34%；对内存的消耗率在 5%～7%. 当攻击者 A、B 同时发起应用层攻击时，由于它们发送的数据包中含有大量的关键词，而本系统在检测到每个关键词后都要调用应用层风险评估模块对其进行风险评估，从而导致本系统对 CPU 的消耗显著增加. 本系统对内存的消耗比较小，即使当网络中存在应用层攻击时，本系统对内存的消耗率也只在 6% 左右.

4.3　系统对比

现有能识别、防御应用层攻击的系统主要有 PAYL(Payload-based Anomaly Detection)系统[19]，该系统是由哥伦比亚大学学者 Wang Ke 等人于 2004 年提出的. PAYL 系统是通过分析数据包载荷的字符统计分布来检测应用层攻击，该系统在计算数据包载荷的字符统计分布时采用 n-gram 分析法.

我们使用 1999 DARPA 数据集对本系统和 PAYL 系统进行了一些对比测试. 在系统误报率相同的条件下，本系统与 PAYL 系统对 1999 DARPA 数据集中一些应用层攻击的检测率如表 3 所示. 本系统主要针对观测序列较长的应用层攻击，在防御观测序列过短的应用层攻击时其效果不是很理想. 在 1999 DARPA 数据集中，由于 Eject 和 Netcat 攻击持续的时间很短，从而导致它们产生的观测序列过短，因此本系统在识别 1999 DARPA 数据集中 Eject 和 Netcat 攻击时其检测率比较低.

表 3　PAYL 系统与本系统的检测率

应用层攻击	PAYL 系统检测率/%	本系统检测率/%
Apache2	91	99.6
Back	95	99.2
Mailbomb	67	99.5
Guessftp	72	98
Guesstelnet	71	97
Eject	91	84
Ftpwrite	83	96
Netcat	92	81

虽然 PAYL 系统在识别 1999 DARPA 数据集中 Eject 和 Netcat 攻击时，其检测效果比我们的系统要好，但 PAYL 系统存在以下不足：首先，PAYL 系统是基于单个数据包来检测应用层攻击，而没有考虑数据包之间的联系，目前很多应用层攻击需要通过分析数据包之间的前后联系才能识别，例如应用层 DDOS 攻击、垃圾邮件等，因此 PAYL 系统很难识别出这类应用层攻击；其次，PAYL 系统在检测出应用层异常时采取的是阻止数据包或者通过报警提请人工干预，而异常检测通常存在误报率过高的问题，如果采用上述硬性控制措施，那么往往会对网络产生严重损害，或者需要人工从海量报警信息中分析出真正的危险并采取化解措施.

5　总结与展望

本文提出一种基于应用层协议分析的应用层实时主动防御系统，该系统由应用层协议识别模块、应用层协议关键词提取模块、应用层风险评估模块、控制模块四部分组成. 其中应用层风险评估模块是该系统的核心，这个模块中的应用层风险实时评估方法是本文的一个重要创新点. 我们对本文提出的主动防御系统进行了一些测试，实验结果表明：该系统在评估用户行为风险值时，具有很高的准确率和较低的误报率；当网络中存在应用层攻击时，该系统能主动化解网络中的攻击，实现应用层主动防御. 本文提出的主动防御系统是基于异常检测来实现应用层主动防御，因此该系统能够防御应用层上的未知攻击和新颖攻击.

目前该系统存在的一个主要问题是：运行该系统时其对 CPU 的消耗比较大，尤其当网络中存在应用层拒绝服务攻击时. 因此下一步的研究计划是：采用一些并行算法来评估应用层风险，以便降低该系统对 CPU 的消耗.

致 谢 真诚感谢审稿专家为本文提出的宝贵意见!

参 考 文 献

[1] Fang Bin-Xing. http://www.cert.org.cn/articles/news/common/2007051823317.shtml, 2007(in Chinese)
(方滨兴. 解读信息安全创新突破点. http://www.cert.org.cn/articles/news/common/2007051823317.shtml, 2007)

[2] Wang B, Zhu P, Wen Q et al. A honeynet-based firewall scheme with initiative security strategies//Proceedings of the 2009 International Symposium on Computer Network and Multimedia Technology. Wuhan, China, 2009: 1-4

[3] Jiang Wei, Fang Bin-Xing, Tian Zhi-Hong, Zhang Hong-Li. Evaluating network security and optimal active defense based on attack-defense game model. Chinese Journal of Computers, 2009, 32(4): 817-827(in Chinese)
(姜伟, 方滨兴, 田志宏, 张宏莉. 基于攻防博弈模型的网络安全测评和最优主动防御. 计算机学报, 2009, 32(4): 817-827)

[4] Deng Song, Lin Wei-Min, Zhang Tao, Yu Yong. Distributed proactive defense based on cloud computing//Proceedings of the 2010 International Conference on Intelligent Computing and Integrated Systems. Guilin, China, 2010: 95-98

[5] Nguyen H V, Choi Y. Proactive detection of DDoS attacks utilizing k-NN classifier in an anti-DDos framework. International Journal of Electrical, Computer, and Systems Engineering, 2010, 4(4): 247-252

[6] Ranjan S, Swaminathan R, Uysal M et al. DDoS-resilient scheduling to counter application layer attacks under imperfect detection//Proceedings of the 25th IEEE International Conference on Computer Communications. Barcelona, Spain, 2006: 1-13

[7] Arnes A, Valeur F, Vigna G et al. Using hidden Markov models to evaluate the risks of intrusions//Proceedings of the RAID'06. Hamburg, Germany, 2006: 145-164

[8] Li Wei-Ming, Lei Jie, Dong Jing, Li Zhi-Tang. An optimized method for real time network security quantification. Chinese Journal of Computers, 2009, 32(4): 793-804(in Chinese)
(李伟明, 雷杰, 董静, 李之棠. 一种优化的实时网络安全风险量化方法. 计算机学报, 2009, 32(4): 793-804)

[9] Chen Xiu-Zhen, Zheng Qing-Hua, Guan Xiao-Hong, Lin Chen-Guang. Quantitative hierarchical threat evaluation model for network security. Journal of Software, 2006, 17(4): 885-897(in Chinese)
(陈秀真, 郑庆华, 管晓宏, 林晨光. 层次化网络安全威胁态势量化评估方法. 软件学报, 2006, 17(4): 885-897)

[10] Wang Yi-Feng, Li Tao, Hu Xiao-Qin, Song Cheng. A real-time method of risk evaluation based on artificial immune system for network security. Acta Electronica Sinica, 2005, 33(5): 945-949(in Chinese)
(王益丰, 李涛, 胡晓勤, 宋程. 一种基于人工免疫的网络安全实时风险检测方法. 电子学报, 2005, 33(5): 945-949)

[11] Rabiner L R. A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 1989, 77(2): 257-286

[12] Yu S Z, Kobayashi H. An efficient forward-backward algorithm for an explicit-duration hidden Markov model. IEEE Signal Processing Letters, 2003, 10(1): 11-14

[13] L7-filter: http://l7-filter.sourceforge.net/

[14] Yu F, Chen Z, Diao Y et al. Fast and memory-efficient regular expression matching for deep packet inspection//Proceedings of the 2006 ACM/IEEE Symposium on Architecture for Networking and Communications Systems. San Jose, California, USA, 2006: 93-102

[15] Yatagai T, Isohara T, Sasase I. Detection of HTTP-GET flood attack based on analysis of page access behavior//Proceedings of the 2007 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing. Victoria, Canada, 2007: 232-235

[16] Floyd S. Random early detection gateways for congestion avoidance. IEEE/ACM Transactions on Networking, 1993, 1(4): 397-413

[17] TC: http://www.linuxfoundation.org/en/Net:Iproute2

[18] Mahoney M V, Chan P K. An analysis of the 1999 DARPA/Lincoln Laboratory evaluation data for network anomaly detection//Proceedings of the 6th International Symposium on Recent Advances in Intrusion Detection. Pittsburgh, PA, USA, 2003: 220-237

[19] Wang K, Stolfo S J. Anomalous payload-based network intrusion detection//Proceedings of the 7th International Symposium on Recent Advances in Intrusion Detection. Sophia Antipolis, France, 2004: 203-222

XIE Bai-Lin, born in 1982, Ph. D. candidate. His current research interests include application layer proactive defense and anomaly detection techniques.

YU Shun-Zheng, born in 1958, Ph. D., professor, Ph. D. supervisor. His main research interests include information security, signal processing, wireless network, etc.

Background

Network proactive defense means acting in anticipation to oppose attacks against the network. It can protect the network from unknown attacks. Existing proactive defense techniques are deployed on network layer and transport layer. Today the new network attacks often occur at application layer. These attacks may not generate abnormal network traffic and present significant malicious activities on the network layer and transport layer, such as application layer DDOS attack and e-mail spam. Therefore, it is difficult for existing proactive defense techniques to effectively detect such application layer attacks without special techniques.

Network risk real-time evaluation is the key to the proactive defense. Existing risk real-time evaluation methods usually evaluate the network risk using IDS (Intrusion Detection System) alerts. These methods can only evaluate the risk from network layer and transport layer, they can't evaluate the application layer risk. In order to perform the application layer proactive defense, we need a risk real-time evaluation method for application layer. This paper presents a risk real-time evaluation method for application layer based on hidden semi-Markov model, this method evaluates the application layer risk by analyzing network traffic. Based on this risk evaluation method and application layer protocol analysis, this paper presents a real-time proactive defense system for application layer. When user's behavior is at risk, the system queues the user's packets according to the risk indicator. By this means, the proposed system can automatically restrict each user's anomalous behavior, and achieve the application layer proactive defense.

This work is supported by the National High Technology Research and Development Program of China under grant No. 2007AA01Z449, the Key Program of NSFC-Guangdong Joint Funds under grant No. U0735002, the National Natural Science Foundation of China under grant Nos. 60970146, 61070154.