应用层洪泛攻击的异常检测*)

谢 逸 余顺争
(中山大学电子与通信工程系 广州 510275)

摘 要 从近年的发展趋势看，分布式拒绝服务攻击已经从原来的低层逐渐向应用层发展，它比传统的攻击更加有效且更具隐蔽性。为检测利用合法应用层 HTTP 请求发动的洪泛攻击，本文把应用层洪泛攻击视为一种异常的用户访问行为，从用户浏览行为的角度实现攻击检测。基于实际网络流的试验表明，该模型可以有效测量 Web 用户的访问行为正常度并实现应用层的 DDoS 洪泛攻击检测。

关键词 应用层，洪泛攻击，异常检测

A Model for Detecting Application Layer Flooding Attacks

XIE Yi YU Shun-Zheng
(Department of Electrical and Communication Engineering, Sun Yat-Sen University, Guangzhou 510275)

Abstract Distributed Denial of Service (DDoS) attacks are typically carried out at the network layer. However, there is evidence to suggest that application layer DDoS attacks can be more effective than the traditional ones. A sophisticated attack using legitimate application layer HTTP requests from legitimately connected network machines to overwhelm Web server is discussed. A counter-mechanism based on Web user browsing behavior is proposed to protect the servers from these attacks. In contrast to prior works, Hidden semi-Markov Model is explored to describe the browsing behaviors of Web users and to implement the anomaly detection for the application layer flooding attacks. By conducting an experiment with a real traffic data, the model shows that it is effective in measuring the user behaviors and detecting the application layer flooding attacks.

Keywords Application layer, Flooding, Anomaly detection

1 引言

分布式拒绝服务(Distributed Denial of Service, DDoS)攻击是现有网络安全问题中的一个严重威胁[1]。DDoS 攻击可以在没有任何先兆的情况下轻易地耗尽攻击目标的计算资源和通信资源。由于 DDoS 攻击的严重性，许多学者提出了基于统计的防御机制，实现 DDoS 攻击的检测与防御。Douligeris[1]总结了近年在 DDoS 攻击检测领域的主要成果。现有基于统计的方法主要利用了分组头信息的属性，例如：IP 地址、TTL(time-to-live)和协议类型等。通过测量到达分组头信息的属性，把那些被认为异常的分组抛弃，从而实现攻击防御。这一类方法通常隐含了一个前提条件：正常分组与攻击分组在分组头信息或流特性存在某些统计差异，因此可以基于这些统计特性检测异常流。

传统的 DDoS 洪泛攻击通常发生在网络层，例如：SYN/ACK 洪泛攻击、UDP 洪泛攻击和 ICMP 洪泛攻击等。然而，随着低层防御措施的加强，攻击者逐渐把其攻击策略向应用层迁移。为了躲避检测，攻击者从原来单纯的带宽洪泛攻击改变为模拟网络中的突发流(flash crowds)，或模仿合法 Web 用户的浏览行为。这些基于应用层的攻击以高层服务器资源作为攻击目标，例如：sockets、磁盘带宽、数据库带宽和进程等。由于这一类洪泛攻击是基于应用层，本文称它为应用层洪泛攻击(Application layer Flooding, App-Flooding)。文[2,3]指出，随着互联网上 Web 应用的复杂度不断地增加，网络带宽的不断增长，服务器资源(例如 CPU 和 I/O 带宽)将成为网络中的主要瓶颈。由于一个高层的 Web 请求所能实现的计算复杂度远比一个低层分组要大得多，因此对于应用层攻击，低流量的攻击请求就足够耗尽目标的 CPU 资源，使其无法正常处理其它合法的 Web 请求。另一方面，与基于低层的 DDoS 攻击不同，App-Flooding 攻击并不依赖于不完善的低层协议或操作系统，它们通过合法的网络链接和请求来实现 DDoS 攻击的效果。2004 年的蠕虫病毒“Mydoom”就是一种典型的 App-Flooding 攻击。

由于恶意请求与合法请求的差异仅在于本身的目的而不在于请求的内容，因此现有的检测系统很难区分出攻击者发出的攻击请求与正常用户的 Web 请求。而且，由于攻击请求可能来自不同地理位置的合法计算机，因此传统针对虚假 IP 地址的 DDoS 检测方案难以实现有效的过滤。基于密码的方案可以非常有效地阻断 App-Flooding 攻击源，但是密码的验证需要建立一个新的连接并允许一些未经过鉴权的用户直接访问服务器的 socket 缓存和工作进程，这很容易促使攻击者针对鉴权机制发动另一次攻击。因此 App-Flooding 的防御是现代网络安全中的一个新挑战。

考虑到基于高层的 App-Flooding 攻击信号可以认为是一种异常的 Web 用户行为，本文将从用户行为的角度实现 App-Flooding 的攻击检测。我们将设计一种新的模型，用于描述 Web 用户的访问行为特征。文[4,6]有许多关于 Web 用户行为的研究。然而现有关于 Web 应用的研究并没有广泛应用于网络安全，而且由于这些行为模型的计算复杂度很高，难以用于实时的在线异常检测。因此，本文使用隐半马

*)本文由国家自然科学基金(90304011)、广东省自然科学基金(04009747)及高等学校博士学科点专项科研基金(20040558043)资助。谢 逸 博士研究生；余顺争 教授，博士生导师。

尔可夫模型(Hidden semi-Markov Model，HsMM)[7,8]来描述Web用户的浏览行为，并应用该模型对模仿正常用户HTTP请求的App-Flooding攻击进行异常检测。

2 模型与算法

2.1 隐半马尔可夫模型

HsMM是一种带状态驻留的隐马尔可夫模型(Hidden Markov Model，HMM)[7]。它是随机有限状态机，其参数包括：(1)$S=\{1,\cdots,N\}$是隐状态集合，N是状态数。(2)π是初始状态概率矩阵，$\pi_m\equiv(\Pr[q_1=m]$，q_t表示t时刻的状态，$m\in S$，初始状态概率分布满足$\sum_m\pi_m=1$，$m\in S$。(3)A是状态转移矩阵，$a_{mn}\equiv\Pr[q_t=n\mid q_{t-1}=m]$，$m,n\in S$，且满足$\sum_n a_{mn}=1$，$m\in S$。(4)$P$是状态驻留矩阵，$p_m(d)\equiv\Pr[\tau_t=d\mid q_t=m]$，$\tau_t$表示当前状态$q_t$还将持续的次数，$d\in\{1,\cdots,D\}$，$D$是状态驻留的最大次数，且满足$\sum_d p_m(d)=1$，$d\in\{1,\cdots,D\}$且$d\geqslant 1$。

由于模型的隐状态是不可观测的，只能从观测符号$O=(o_1,\cdots,o_T)$序列中获取信息，其中o_t表示模型在t时刻的输出，T表示观测序列的长度。对于一个指定的状态m，模型的输出分布表示为$b_m(k)\equiv\Pr[o_t=k\mid q_t=m]$，其中$m\in S$且$k\in V=\{1,\cdots,K\}$。$K$表示模型输出符号的个数。输出概率满足$\sum_k b_m(k)=1$，$m\in S$，$k\in V$。因此，HsMM的模型参数$\lambda$包括初始状态分布，状态转移概率分布，输出概率分布和状态驻留次数概率分布。为了简化，本文使用$\lambda=(\{\pi_m\},\{a_{mn}\},\{b_m(k)\},\{p_m(d)\})$表示HsMM的参数。通过模型的学习、评估和解码[7]，HsMM可以用于分类和模式识别。此外，近10年的研究[6]表明网络流具有自相似性。Yu等[9]证明了HsMM比传统的HMM更能用于描述网络流随时间变化的二阶自相似性和长相关性。因此，HsMM适合用于描述Web用户浏览行为并执行异常检测。HsMM算法在文[7,8]中有详细的论述。

2.2 访问行为的数学描述

从客户的角度看，一次合法的Web浏览行为包含访问期间发送的多个请求。用户可能采用"闭合循环"的方式发送请求，也就是客户发送一组请求后就处于暂停状态，等待服务器返回的响应，直到所有的响应都到齐了，才进行下一轮的请求。用户也可以采用"管道"形式发送请求，也就是他们连续发送多个Web请求而不需要停下来等待服务器的响应。对于Web应用来说，一次Web页面的点击行为通常包含一个对主文档的请求和多个对该页面内嵌对象的请求(例如：图片，广告条)。通常情况下，主文档一般是动态页面需要涉及到数据库操作，而内嵌对象则是静态文件。从Web请求的传输过程看，客户的Web请求首先被路由到代理服务器，如果代理服务器所保存的相关对象是有效的，它将直接响应客户的请求，否则，代理服务器把该用户的Web请求转发到目的服务器。如果客户请求的仅是静态页面，例如：HTML页面或静止图像，Web服务器搜索该对象并返回响应。如果客户所请求的内容包含了动态Web内容，服务器将执行脚本程序，例如PHP，JSP或者Javascript。这一类的请求通常包含了一次或多次的数据库查询，服务器根据查询的结果生成响应页面返回给客户。

在上述Web访问行为中，有两点需要注意。首先，由于Internet中的代理和缓存有可能直接响应客户的Web请求，因此并不是所有的客户请求都可以到达目标Web服务器。这将导致即使是完全相同的访问行为(请求内容)，但由于客户的位置不同，或由于请求所经过的路由不同，Web服务器上日志文件所记录的内容存在差异。其次，由于客户在进行Web访问时，既可以按Web页面上提供的连接进行，也可以直接在浏览器的地址栏中直接输入URL地址，或者使用"前进"或"后退"按钮，因此Web服务器的日志文件所放映的用户访问行为可能显得杂乱无章。

考虑到以上特点，我们采用以下的数学方法描述Web用户的访问行为：

1)使用变量q_t表示用户的点击行为，于是用户的访问行为可以看为一个随机过程$\{q_t:t=1,\cdots,T\}$，q_t表示用户第t次请求所属的Web页面。q_{t-1}到q_t的跳转可以认为是用户从一个Web页面到另外一个Web页面的点击行为，用概率$a_{q_{t-1}q_t}$表示q_{t-1}到q_t的转移概率。

2)使用另外一个变量o_t表示用户在第t个请求所索取的对象的编号，例如：一个HTML页面或内嵌图像文件。于是用户的HTTP请求序列可以使用另外一个随机过程$\{o_t:t=1,\cdots,T\}$表示。

3)使用第3个随机变量τ_t表示当用户点击一个连接(请求页面q_t)后，浏览器发出的所有HTTP请求中能够到达目标Web服务器的请求的个数。

在上述3个随机变量中，$\{o_t\}$可以通过分析Web服务器的日志文件得到，而$\{q_t\}$与$\{\tau_t\}$无法观测得到。考虑到用户的访问通常是从一个页面到另一个页面，我们假设用户的点击行为$\{q_t\}$是一个Markov过程。由于用户的真实点击行为在服务器端是不可观测得到的，而且用户的一次点击行为有可能促使浏览器产生多个HTTP请求(包括对HTML文档和内嵌对象的请求)，我们进一步使用一个半Markov过程来描述Web用户的点击行为。因此，Web用户的访问可以使用以下过程描述：每一个被点击的Web页面视为一个Markov状态(隐状态)，用户对该页面的HTTP请求及对相关内嵌对象的请求是Markov隐状态的观测值，一次点击行为所产生的且最后能到达目标Web服务器的HTTP请求的个数是半Markov状态的驻留次数。于是，可以使用HsMM[7,8]描述Web访问过程。根据HsMM算法，可以估计模型的参数λ，用户观测序列相对于给定参数的模型的或然概率由下式计算得到：

$$\Pr[o_1^{T(1)}\mid\lambda]=\sum_{m,d}\Pr[o_1^{T(1)},(q_{T(1)},\tau_{T(1)}=(m,d)\mid\lambda] \quad (1)$$

$$\Pr[O\mid\lambda]=\Pr[O^{(1)},\cdots,O^{(L)}\mid\lambda]=\prod_{l=1}^{L}\Pr[O_l^{T(1)}\mid\lambda] \quad (2)$$

2.3 App-Flooding攻击检测

利用正常用户发出的HTTP请求序列建立HsMM模型用于描述Web用户访问行为轮廓。通过模型可以计算每个正常用户请求序列的或然概率，这些或然概率构成了初始或然概率分布(Original Likelihood Distribution，OLD)。实验(第4节)表明大部分正常用户相对于模型的或然概率都非常相似。根据异常检测原理，可以根据实测用户请求序列相对于代表正常行为的模型的或然概率与OLD的距离来判断其异常性。偏差越小，说明观测序列的正常性越高。据我们搜集到的资料看，目前存在的App-Flooding攻击工具通常是利用目标Web服务器上特定页面(例如：主页或其它高频访问页面)来实现HTTP洪水攻击。虽然每一个用于攻击的HTTP请求都是合法的(包括：合法的IP分组，正常的TCP连接，流特征，及HTTP请求结构)，但是它所表现出来的访问行为却与正常用户存在明显的差异。因此，这将导致用于攻

击的 HTTP 请求序列相对于模型的或然概率偏离 OLD，利用这种偏离可以检测出构成异常浏览行为的 HTTP 请求并实现过滤。模型在实际网络环境下的应用包括以下环节。

选取一组正常用户的请求序列作为训练数据建立模型并计算正常用户的或然概率分布 OLD。异常检测方案设计如图 1 所示。基于用户行为的异常请求过滤器位于 Internet 与目标 Web 服务器之间，拦截发往 Web 服务器的 HTTP 请求，然后依据计算得到的或然概率决定是接受还是拒绝(丢弃)该请求。具体的过滤策略如下：首先，定义 T_d 为采集的 HTTP 请求序列长度，对用户 l 的一个请求序列，使用训练得到的模型 λ 计算它的或然概率 lkh_l。然后比较 lkh_l 和 OLD 的偏差，如果偏差大于预定的极限值 lkh_{thresh}，发出该请求序列的用户将被认为是异常用户，于是在资源缺乏时，该用户发出的所有当前及后续请求将被过滤器所丢弃。否则，用户的 HTTP 请求将通过过滤器到达服务策略模块，按照每个请求序列的或然概率值，服务策略模块在内存中把用户的请求进行排队等待服务器的响应，由队列控制服务器响应客户请求的优先度。

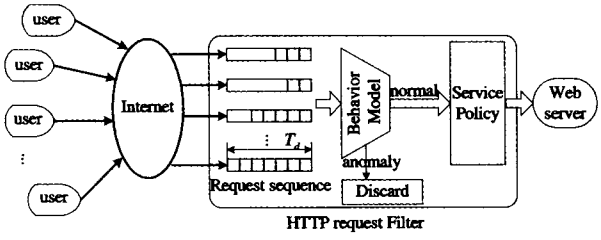


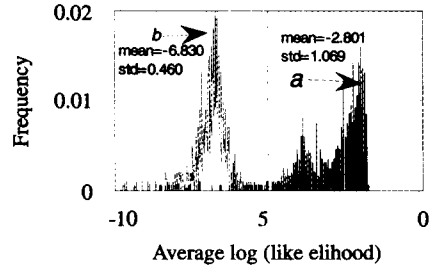
图 1　基于行为模型的请求过滤

在下一节中，我们将使用检测率(Detection Rate，DR)和误检率(False Negative Ratio，FNR)来衡量过滤器的性能。一般情况下，由于计算能力的限制，过滤器很难快速地适应变化的正常用户行为。因此，我们允许过滤器的判断策略(即或然概率的判断极限值)可以在一个很短的期间内维持不变。本文把这一段时间称为一个时隙。当一个时隙结束，模型将自动使用实时数据进行模型参数更新。

3　实验结果分析

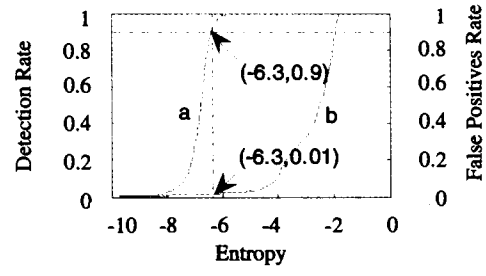
我们使用校园网的 Web 服务器日志文件及仿真的 App-Flooding 攻击来验证所提出的模型与算法。从 Web 日志文件中分离出不同用户的 HTTP 请求序列，并保持每一个请求序列的时间次序。为验证本文模型对 App-Flooding 攻击检测的有效性，我们按照“Mydoom”蠕虫病毒的工作原理模拟了 App-Flooding 攻击。从收集到的资料看，“Mydoom”在发动 DDoS 攻击时，每一个傀儡机每秒可以同时向目标主机发出 64 个 GET 请求。仿真中，我们假设有 200 个潜在的攻击节点。每个节点按照 20ms 的平均间隔(约 50 个请求/秒)向目标服务器发送攻击请求。

图 2 是正常用户和攻击节点的请求序列相对模型的或然概率分布，可以看到它们的分布特征存在明显的差异：大部分正常用户的对数或然概率大于－6，而攻击节点的对数或然概率基本都集中在－6 以下。可见，基于用户访问行为的 App-Flooding 攻击检测模型可以有效地分辨出异常的攻击请求。图 3 是用于异常判断的对数或然概率极限值与检测率，误检率的关系，从图中可见，对于本文所采用的数据，当以－6.3 作为异常判断的极限值时，检测系统的检测率可以达到 90%，而误检率仅为 1%。



a: 正常用户; *b*: 异常用户

图 2　或然概率分布



***a*. DR vs. Entropy; *b*. FPR vs. Entropy**

图 3　或然概率、检测率与误检率

结论　针对 App-Flooding 攻击检测，本文使用 HsMM 描述 Web 用户的浏览行为。通过用户请求序列相对于模型的或然概率实现 App-Flooding 攻击的检测与过滤。仿真试验表明，该模型可以有效用于识别正常用户和恶意攻击者所发出的 HTTP 请求，并实现过滤。而且，当模型的对数或然概率决策极限取为－6.3 时，检测率可以达到 90%，误检率为 1%。

参 考 文 献

1　Douligeris C, Mitrokotsa A. DDoS attacks and defense mechanisms: classification and state-of-the-art [J]. Computer Networks: The International Journal of Computer and Telecommunications Networking, 2004, 44(5): 643～666

2　Amza C, Cecchet E, Chanda A, et al. Bottleneck characterization of dynamic web site benchmarks[R]: [Technical Report TR-02-391]. Rice University, February 2002. online: http://rubis.objectweb.org/download/ dyna_bottleneck.pdf

3　Ranjan S, Karrer R, Knightly E. Wide area redirection of dynamic content by internet data centers[A]. In: the Proceeding of INFOCOM 2004. Volume 2, March 2004. 816～826

4　Chatterjee P, Joffman D, Novak T. Modeling the clickstream: Implications for Web-based advertising efforts[J]. Marketing Science, 2003, 22: 520～541

5　Bürklen S, Marrón P J, Fritsch S, et al. User Centric Walk: An Integrated Approach for Modeling the Browsing Behavior of Users on the Web[A]. In: the Proceedings of the 38th Annual Simulation Symposium (ANSS'05), April 2005. 149～159

6　Dill S, Kumar R, Mccurley K S. Self-Similarity in the Web[J]. ACM Transactions on Internet Technology, 2002, 2(3): 205～223

7　Rabiner L R. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition[A]. Proceeding of IEEE, February 1989, 77(2): 257～286

8　Yu S Z, Kobayashi H. An Efficient Forward-Backward Algorithm for an Explicit Duration Hidden Markov Model[J]. IEEE Signal Processing Letters, 2003, 10(1): 11～14

9　Yu S Z, Liu Z, Squillante M S, et al. A Hidden Semi-Markov Model for Web Workload Self-Similarity[A]. In: Proceedings of The 21st IEEE International Performance, Computing, and Communications Conference (IPCCC 2002). Phoenix, AZ. April 2002. 65～72