基于最大频繁序列模式挖掘的 App-DDoS 攻击的异常检测

李锦玲* 汪斌强
(国家数字交换系统工程技术研究中心 郑州 450002)

摘 要：为了动态、准确、高效地描述用户的访问行为，实现对不同应用层分布式拒绝服务(Application-layer Distributed Denial of Service, App-DDoS)攻击行为的透明检测，该文提出基于最大频繁序列模式挖掘的 ADA_MFSP(App-DDoS Detection Algorithm based on Maximal Frequent Sequential Pattern mining)检测模型。该模型在对正常 Web 访问序列数据库(Web Access Sequence Database, WASD)及待检测 WASD 进行最大频繁序列模式挖掘的基础上，引入序列比对平均异常度，联合浏览时间平均异常度、请求循环平均异常度等有效检测属性，最终实现攻击行为的异常检测。实验证明：ADA_MFSP 模型不仅能有效检测各类 App-DDoS 攻击，且有良好的检测灵敏度。

关键词：应用层分布式拒绝服务攻击；检测模型；频繁序列模式挖掘；异常度
中图分类号：TP393.08 **文献标识码**： A **文章编号**：1009-5896(2013)07-1739-07
DOI: 10.3724/SP.J.1146.2012.01372

Detecting App-DDoS Attacks Based on Maximal Frequent Sequential Pattern Mining

Li Jin-ling Wang Bin-qiang
(*National Digital Switching System Engineering & Technological R&D Center*, *Zhengzhou* 450002, *China*)

Abstract: In order to describe the user's access behavior dynamically, efficiently and accurately, a novel detection model for Application-layer Distributed Denial of Service (App-DDoS) attack based on maximal frequent sequential pattern mining is proposed, named App-DDoS Detection Algorithm based on Maximal Frequent Sequential Pattern mining (ADA_MFSP). After mining maximal frequent sequential patterns of trained and detected Web Access Sequence Database (WASD), the model introduces sequence alignment, view time and request circulation abnormality to describe the behaviour of App-DDoS attacks, finally achieves the purpose of attack detection. It is proved with experiments that the ADA_MFSP model can not only detect kinds of App-DDoS attacks, but also has good detection sensitivity.

Key words: Application-layer Distributed Denial of Service (App-DDoS) attack; Detection model; Frequent sequential pattern mining; Abnormality

1 引言

随着互联网的高速发展和广泛普及，相应的网络安全问题也越来越引人关注，在众多的网络攻击中，应用层分布式拒绝服务(Application-layer Distributed Denial of Service, App-DDoS)攻击以其强大破坏力和越发隐蔽性成为网络安全的头号问题[1,2]。由于 Web 日志最直接、最完整地记录了用户的访问行为和攻击者的攻击行为，Web 日志分析成为 App-DDoS 攻击异常检测的研究热点[3]。如何提出有效的日志分析方法，准确描述用户的访问模型，找出异常检测的有效特征，成为 Web 日志分析应用于 App-DDoS 攻击检测的主要问题。文献[4]提出的基于 Session 异常度的检测方法通过日志分析建立 Session 异常度模型，定义详细的异常属性达到 App-DDoS 攻击的检测目的，但该方法主要针对洪泛攻击，且需联合多个异常属性才能达到检测目的。文献[5,6]提出的基于隐半马尔可夫链(HsMM)的检测方法，在日志分析的基础上采用 HsMM 描述用户的访问行为，并用与大多数正常用户访问行为的偏离作为异常程度的测量，达到 App-DDoS 攻击检测的目的。该方法最大的贡献是将用户的访问行为进行了数学建模，但其仅给出了访问行为的轮廓，且对 Session Flood 攻击检测性能欠佳[7]。

为了动态、准确、高效地描述用户的访问行为，

2012-10-26 收到，2013-02-18 改回
国家科技支撑计划(2011BAH19B01)和国家高技术研究发展计划(2011AA01A103)资助课题
*通信作者：李锦玲 zifenglingsworld@163.com

实现对不同 App-DDoS 攻击行为的透明检测，本文提出了基于最大频繁序列模式挖掘的检测模型 ADA_MFSP(App-DDoS Detection Algorithm based on Maximal Frequent Sequential Pattern mining)。该模型首先采用 SDD-UDDAG(Up and Down Directed Acyclic Graph approach according to the Descending of Support Degree)挖掘算法挖掘出正常 Web 访问序列数据库(Web Access Sequence Database, WASD)及待检测 WASD 的最大频繁序列模式全集，通过动态规划算法实现频繁序列模式异常比对异常度的计算，联合浏览时间异常度及请求循环异常度，进而实现准确透明的攻击检测。

本文的主要贡献如下：(1)提出了适于 App-DDoS 攻击在线检测的 SDD-UDDAG 挖掘及更新算法、滑动窗口设定方法，可动态快速地描述用户的访问行为。(2)引入了基于 Web 日志 URL 片段的序列比对异常度，联合基于访问时间片段得到的浏览时间异常度、请求循环异常度，实现对各类 App-DDoS 攻击的透明高效检测。

2 ADA_MFSP 异常检测模型

ADA_MFSP 检测算法部署于服务器端，流程图如图 1 所示，具体分为训练过程和检测过程。训练过程中，将正常 WASD 分为两部分，一部分用来建立最大频繁序列模式数据库 P''，一部分用作验证序列获得正常情况下序列平均比对异常度 f_{sa}、浏览时间平均异常度 f_{ta} 及请求循环平均异常度 f_{ca} 的时间分布图，设定检测阈值。检测过程中，对当前服务器的 Web 日志进行单点、基于滑动窗口的间隔采集，在此基础上应用 SDD-UDDAG 挖掘、更新算法挖掘出待检测窗口 w_d 内的最大频繁序列模式集 P_d''。将 P_d'' 中的最大频繁序列模式依次按动态规划算法进行序列比对异常度 f_s 的计算，对所得 f_s 进行加权相加得到有效检测属性 f_{sa}，联合预设阈值做出异常判定。若判定异常，报警输出，若判定正常，则可能为 Session Flood 攻击，进一步计算序列浏览时间平均异常度 f_{ta} 及请求循环平均异常度 f_{ca}，联合相应阈值做出异常判定，若为正常，进行训练数据库的动态更新，若为异常，报警输出。

2.1 滑动窗口大小设定算法

为了准确动态地进行异常检测及增量更新，最大程度地减轻服务器的计算压力，本文采用滑动窗口作为 Web 日志的采集间隔及 ADA_MFSP 的检测间隔，其大小设定如下：

输入：检测窗口初值 w_0，训练 WASD，待检测 WASD

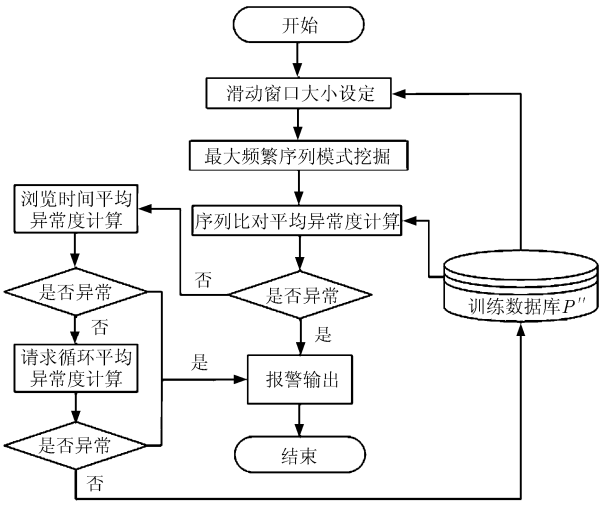


图 1 ADA_MFSP 检测模型

输出：最佳检测窗口大小

方法：

(1)设训练 WASD 的训练时间为 t，共包含 N_{sum} 条事务序列，计算得 w_0 窗口内包含的平均训练事务序列数为 $N_{w_0}=\lceil N_{\text{sum}}\times w_0/t\rceil$。

(2)在 w_0 的检测窗口内，待检测 WASD 共包含 N_{dw_0} 条事务序列，令 $\rho=(N_{dw_0}-N_{w_0})/N_{w_0}$，设定窗口 $w_1=\lceil w_0\times(1-\rho)\rceil$，窗口内包含的事务序列个数记为 N_{dw_1}。

(3)完成 w_1 窗口内的检测后，令 $w_2'=w_1$，若 w_2' 内包含 $N_{dw_2'}$ 条事务序列，计算 $\rho=(N_{dw_2'}-N_{dw_1})/N_{dw_1}$，设定窗口 $w_2=\lceil w_1\times(1-\rho)\rceil$。

(4)同理设定后续检测窗口的大小。

2.2 SDD-UDDAG 挖掘、更新算法

对 Web 日志经过预处理[8]，保留其 URL 地址及访问时间片段，形成 Web 访问序列数据库 WASD[9]，如表 1 所示。不失一般性，将 WASD 中的 URL 地址用字母表示，由于频繁序列挖掘无需访问时间片段，表中未标出。

为了满足频繁序列模式挖掘算法应用于 App-DDoS 攻击检测的需要[10]，本文提出基于 SDD-UDDAG 的频繁序列模式挖掘算法，该算法支持待挖掘序列模式的双向增长，在 k 次迭代后，序列模式至多能增长至 2^{k-1}，即至少迭代 $\lfloor\log_2 k\rfloor+1$ 次便可

表 1 Web 访问序列数据库 WASD

用户 (User)	Web 访问序列	频繁项序列	编号后的频繁项序列化
U1	abchea	abcea	54235
U2	bacegc	bacec	45232
U3	baef	baef	4531
U4	febad	feba	1345

得到k阶频繁序列模式，相比 PrefixSpan 等其他挖掘算法迭代次数明显下降[11]，效率明显提高。算法表述如下：

算法：SDD-UDDAG 挖掘算法

输入：WASD, minsup

输出：WASD 的最大频繁序列模式集 P''

方法：

(1)根据 minsup 得到 WASD 的一阶频繁序列集$(x_1, x_2, \cdots, x_m)$，按支持度升序编号为 1～m。

(2)按编号降序进行挖掘，首先建立频繁序列$i(1 \le i \le m)$的前缀/后缀投影序列库，并记录相应的元组序列号，保留其中编号为i或小于i的频繁项，记为 Pre(^iWD)/Suf(^iWD)，对 Pre(^iWD)/Suf(^iWD)分别进行一阶至高阶频繁序列的挖掘。

(3)根据 Pre(^iWD)$_1$/Suf(^iWD)$_1$的频繁序列，建立i的 UP-DAG, DOWN-DAG 图。

(4)对 UP-DAG 的任意节点v，求其有效下节点集$VDVS_v$=$\{v' \mid (v' \in V_D) \wedge (op(v).<i>.op(v') \in P_i)\}$，联合$VDVS_v$最终建立$i$的 UDDAG 图，求得$P_i$。

(5)保留P_i中的最大频繁模式集，记为P_i''。

(6)挖掘结束后，形成 WASD 的最大频繁序列模式集$P'' = P_1'' \cup \cdots \cup P_i'' \cup \cdots \cup P_m''$。

应用上述算法挖掘表 1 中 WASD 的完全最大频繁序列模式集：

(1)令 minsup 为 2，则该 WASD 的一阶频繁模式集为{f:2,c:3,e:4,b:4,a:5}，按支持度升序编号为{f→1,c→2,e→3,b→4,a→5}。

(2)对频繁项 5 进行挖掘，首先建立 Pre(^5WD)/Suf(^5WD)并记录相应的元组序列号，得到 Pre(^5WD):U1)<5 4 2 3>, U2)<4>, U3)<4>, U4)<1 3 4>;Suf(^5WD):U1)<4 2 3 5>, U2)<2 3 2>, U3)<3 1>。分别求得 Pre(^5WD)/Suf(^5WD)中的频繁序列，建立频繁项 5 的 UP-DAG, DOWN-DAG，如图 2(a), 2(b)所示。

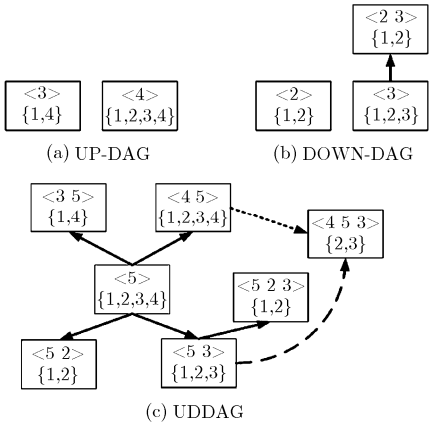


图 2 频繁项 5 的 UP-DAG，DOWN-DAG 及 UDDAG

(3)对 UP-DAG 中的节点<4>，求得其$VDVS$ = {3}，得最终的 5-UDDAG 如图 2(c)所示。

(4)由 5-UDDAG 可得P_5={<3 5>,<4 5>,<5 2>, <5 3>,<5 2 3>,<4 5 3>}，保留其最大频繁模式P_5'' = {<3 5>,<5 2 3>,<4 5 3>}，用于后续异常比对及检测。

(5)同理，可得P_4''={<4 2 3>},P_3''={<2 3>}, P_2'' ={<2>},P_1''={<1>}由于$P_2'' \subseteq P_3'' \subseteq P_4''$，故保留$P_4''$即可。

(6)该 WASD 的最大频繁模式集为{<3 5>, <5 2 3>,<4 5 3>,<4 2 3>,<1>}。

SDD-UDDAG 更新算法具有更新速度快且计算简单的优势，具体实施为：若待检测 WASD 的一阶频繁项中含有未包含于训练 WASD 一阶频繁序列集$(x_1, x_2, \cdots, x_m)$的频繁项，则将这些频繁项按支持度升序编号为$m+1, m+2, \cdots$，其中 m 为训练 WASD 一阶频繁序列的最大编号，然后依据 SDD-UDDAG 挖掘步骤进行待检测 WASD 完全频繁序列模式全集P_d''的挖掘。

2.3 序列比对平均异常度 f_{sa}

两个序列的比对是指这两个序列中各字符间的一一对应关系，或字符的对比排列。本文中序列比对的根本任务是发现P_d''与P''中最大频繁序列模式之间的差异，以此达到 App-DDoS 攻击的检测目的。由于动态规划算法能计算出双序列比对的最大 SP 函数值[12]，故引入动态规划(dynamic programming)寻优策略进行序列比对优化。

设序列 s, t 的长度分别为 g 和 h，若已知序列 0:s:i($0 \le i \le g$) 和 0:t:j($0 \le j \le h$) 所有较短子序列的最优比对，则有

$$S(_0{:}\,s:_{i,0}:t:_j) = \max\begin{cases} S(_0{:}\,s:_{(i-1),0}:t:_{(j-1)}) + p(s_i, t_j) \\ S(_0{:}\,s:_{(i-1),0}:t:_j) + p(s_i, -) \\ S(_0{:}\,s:_{i,0}:t:_{(j-1)}) + p(-, t_j) \end{cases} \tag{1}$$

其中$S(_0{:}\,s:_{i,0}:t:_j)$为序列 0:s:i 与序列 0:t:j 比对得分，记为$S_{i,j}$，$p(s_i, t_j)$为字符s_i与字符t_j的比对得分，$p(s_i, -)$为字符s_i与空位符的比对得分。

动态规划算法计算过程如下：

输入：拟比对的序列

输出：最佳比对路径

方法：

(1)首先进行序列初始化及$S_{i,j}$初值的设定，完成后从$S_{0,0}$开始计算，在计算$S_{i,j}$后，保存其计算路径。上述计算过程到$S_{g,h}$结束。

(2)最优路径求解：与计算过程相反，从$S_{g,h}$开

始，按计算过程保存的路径反向前推，至 $S_{0,0}$ 结束，反推经过的路径即为最优路径，对应于两个序列的最优比对。

根据 SDD-UDDAG 挖掘过程，可知 $P'' = P_1'' \cup \cdots \cup P_i'' \cup \cdots \cup P_m''$，其中 $P_i''(1 \le i \le m)$ 为包含编号为 i 或小于 i 的频繁项，同理，$P_d'' = P_{d1}'' \cup \cdots \cup P_{di}'' \cup \cdots \cup P_{dn}''$，其中 $P_{di}''(1 \le di \le n)$ 为包含编号为 di 或小于 di 的频繁项。若 $di > m$，将 P_{di}'' 同 P_m'' 按动态规划算法进行比对，若 $di \le m$，将 P_{di}'' 同 P_i'' 进行序列比对，其中 $di = i$。

考虑异常检测的需要，避免两两比对出现混乱，本文采用星形比对结构进行 P_d'' 与 P'' 之间的序列比对异常度计算。选择 $P_{di}''(1 \le di \le n)$ 包含的序列依次为核心序列，与 P'' $(di = i)$ 或 P_m'' 中的序列按动态规划算法进行两两比对，选择最大 SP 函数值进行序列比对异常度 f_s 的计算：

$$f_s = 1 - \frac{\max(SP)}{l} \tag{2}$$

其中 l 为核心序列的长度。

引入影响因子 $\alpha =$ 包含 p_j 的元组总数 $\Big/\sum_{p_j}$ 包含 p_j 的元组总数，$\gamma = f_s^{p_j} \Big/ \sum_{p_j} f_s^{p_j}$，则可得检测窗口 w_d 内最大频繁序列模式的序列比对平均异常度为

$$f_{sa} = \frac{1}{|P_d''|} \sum_{p_j} \frac{1}{2} \times f_s^{p_j} \times (\alpha + \gamma) \tag{3}$$

其中 $p_j \in P_d''$，$|P_d''|$ 为检测窗口 w_d 内最大频繁序列模式的个数。

2.4 序列浏览时间平均异常度 f_{ta}

对 Web 日志进行预处理形成 WASD 时，保留其访问时间片段，并定义序列的浏览时间为尾项访问时间与首项访问时间之差。

定义 1 给定待检测最大频繁序列模式，除去编号大于 m 的频繁项，根据访问时间及 UDDAG 图，分别求得其在 P_d'' 与 P'' 中浏览时间的平均值 t_d，t_t，定义该最大频繁序列模式的浏览时间异常度为

$$f_t = 1 - t_d / t_t \tag{4}$$

若 $f_t \le 0$，令 $f_t = 0$。检测窗口 w_d 内最大频繁序列模式的浏览时间平均异常度为

$$f_{ta} = \frac{1}{|P_d''|} \sum_{p_j} \frac{1}{2} \times f_t^{p_j} \times (\alpha + \gamma) \tag{5}$$

2.5 序列请求循环平均异常度 f_{ca}

根据最大频繁序列模式的首项访问时间及 UDDAG 图，描绘出该频繁序列的时间分布图，以检测窗口 w_d 对时间分布图进行截取，选择该频繁序列最密集出现的时间段，并计算其出现次数。

定义 2 给定待检测最大频繁序列模式，除去编号大于 m 的频繁项，若已知 w_d 时间内该频繁序列在 P''，P_d'' 出现的最大次数分别为 c_t，c_d，定义请求循环异常度为

$$f_c = c_d / c_t \tag{6}$$

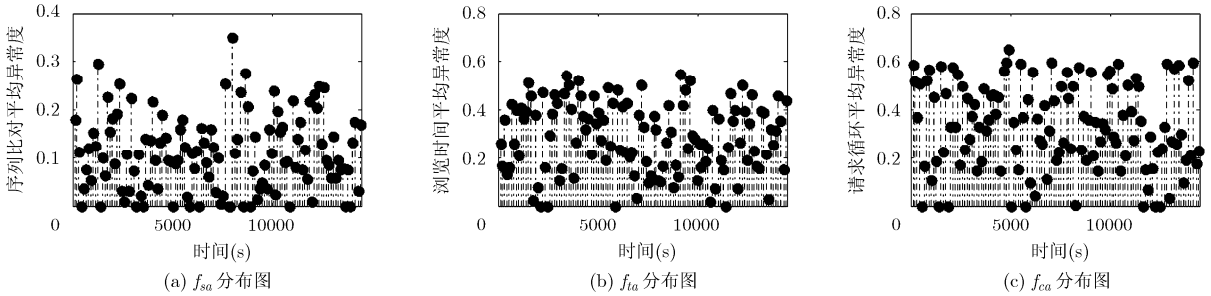
若 $f_c > 1$，令 $f_c = 1$。检测窗口 w_d 内最大频繁序列模式的请求循环平均异常度为

$$f_{ca} = \frac{1}{|P_d''|} \sum_{p_j} \frac{1}{2} \times f_c^{p_j} \times (\alpha + \gamma) \tag{7}$$

3 仿真实验

本仿真实验中，训练阶段采用真实日志 EPA-HTTP 作为训练集，训练时间 t=16 h，共包含 5034 条事务序列，应用 SDD-UDDAG 挖掘算法建立最大频繁序列模式数据库 P''。将后续 t=4 h 的正常序列集作为检测 WASD，共包含 1292 条事务序列，依式(3)，式(5)，式(7)及 P'' 计算得到 f_{sa}，f_{ta}，f_{ca} 的时间分布图，如图 3 所示。根据图 3 中训练异常度分布，设定 f_{sa}，f_{ta}，f_{ca} 的检测阈值分别为 0.4, 0.6, 0.7。

根据 App-DDoS 攻击原理，利用后续 4 h 的正常检测 WASD 分别模拟资源消耗型攻击，Forged-URL Flood, Single-URL Flood, Multi-URL Flood, Random-URL Flood 及 Session Flood 攻击，并将其按比例注入到检测 WASD 中。以设定的滑动窗口为采集间隔对注入攻击后的 WASD 进行采集、预处理，应用 ADA_MFSP 检测模型计算各类攻击的有



(a) f_{sa} 分布图 (b) f_{ta} 分布图 (c) f_{ca} 分布图

图 3 f_{sa}，f_{ta}，f_{ca} 的时间分布图

效异常度，在与预设阈值相比较的基础上最终实现 App-DDoS 攻击的异常检测。

(1)资源消耗型攻击。根据 Web 日志片段判断出图像、视频或数据库查询等消耗资源的 URL 请求地址，采用这些 URL 地址随机生成资源消耗型攻击事务序列，应用 ADA_MFSP 检测模型计算得到 f_{sa} 的分布图如图 4(a)所示。由图 4(a)可得，当攻击序列比例大于 37%时，可实现无漏报检测，当攻击序列比例大于 84%时，f_{sa} 趋近于 1。

(2)Forged-URL Flood 攻击。随机伪造长度大于 800 byte 的 URL 地址形成攻击事务序列，应用 ADA_MFSP 检测模型计算得到 f_{sa} 的分布图如图 4(b)所示。由图 4(b)可得，当攻击序列比例大于 26%时，可实现无漏报检测，当攻击序列比例大于 69%时，f_{sa} 趋近于 1。

(3)由于 Single-URL Flood, Multi-URL Flood, Random-URL Flood 发起攻击原理相同，且 Random-URL Flood 的隐蔽性更好，因此通过模拟 Random-URL Flood 攻击测试模型检测性能。采用训练 WASD 中的 URL 地址随机生成攻击事务序列，应用 ADA_MFSP 检测模型计算得到 f_{sa} 的分布图如图 4(c)所示。由图 4(c)可得，当攻击序列比例大于 42%时，可实现无漏报检测，当攻击序列比例大于 94%时，f_{sa} 趋近于 1。

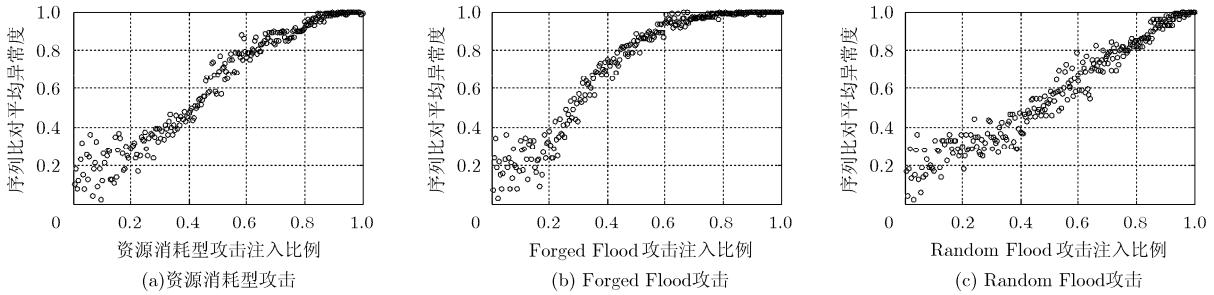
(4)Session Flood 攻击。随机采用训练 WASD 中的事务序列生成攻击事务序列，以高请求速率、正常请求速率分别进行攻击，应用 ADA_MFSP 检测模型计算得到高请求速率 f_{sa}, f_{ta}，正常请求速率 f_{ca} 的分布图如图 5 所示。由图 5 可知，当高请求速率 Session Flood 攻击的注入比例增大时，f_{sa} 反而减小，不能作为异常检测的有效特征，而当攻击序列比例大于 67%时，$f_{ta} > 0.6$，可实现攻击的无漏报检测。同样，当正常请求速率 Session Flood 攻击的注入比例增大时，f_{ta} 亦不能作为异常检测的有效特征，而当攻击序列比例大于 70%时，$f_{ca} > 0.7$，可实现攻击的无漏报检测。

4　性能分析

4.1 实验结果分析

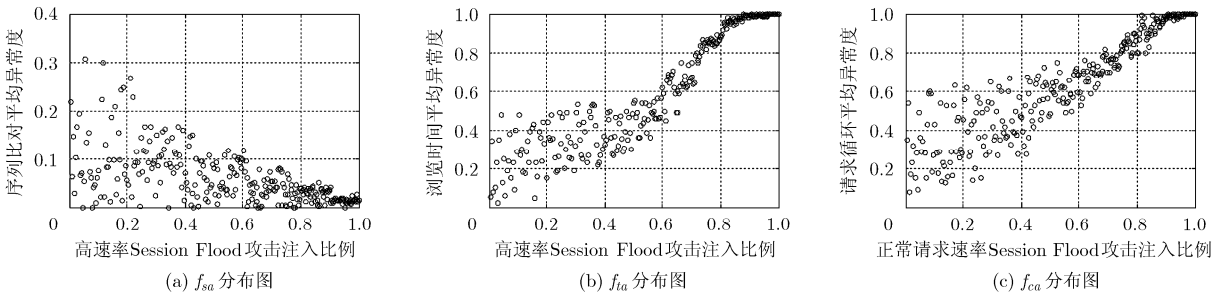
(1) f_{sa} 对除 Session Flood 以外的攻击都有很好的检测效果，尤其是对 App-DDoS 攻击检测盲区的资源消耗型攻击有很好的检测效果，在联合 f_{ta}, f_{ca} 之后，对相对隐蔽的 Session Flood 攻击也有令人满意的检测效果。

(2)由图 4 可知，f_{sa} 对于 Forged-URL Flood 攻击的检测最为敏感，攻击流量比例仅为 26%时便可实现无漏报检测，这是由于 Forged-URL Flood 的频繁项都系伪造，与训练 WASD 频繁项的相似度几乎为零，所以 f_{sa} 序列比对异常度取值基本都趋近于 1，少量的攻击流量便可达到检测目的。对于资源消耗型攻击及 Random-URL Flood 攻击，f_{sa} 的检测性能相当，这是由于两类攻击都是由 URL 地址随机生成攻击事务序列，只是生成攻击事务序列的 URL 范围不同。



(a)资源消耗型攻击　(b) Forged Flood攻击　(c) Random Flood攻击

图 4　f_{sa} 分布图



(a) f_{sa} 分布图　(b) f_{ta} 分布图　(c) f_{ca} 分布图

图 5 Session Flood 攻击的异常度分布图

(3)由图 5 可知，单纯依靠 f_{sa} 无法达到检测 Session Flood 攻击的目的，甚至会造成严重的误判。f_{ta} , f_{ca} 作为有效特征时，随着异常流比例的增加，检测率振荡提高，在异常流比例高于正常流时，可较快实现无漏报检测。

4.2 算法复杂度分析

ADA_MFSP 检测模型的时间复杂度主要来自于 SDD-UDDAG 挖掘算法及异常度求解算法。SDD-UDDAG 算法的时间复杂度为 $O(L\cdot u\cdot d\cdot\log_2 k)$，其中 L 为一阶频繁模式的总数，k 为最大频繁序列的长度，u 为 UDDAG 树中包含的频繁项个数，d 为 DOWNDAG 树中包含的频繁项个数，$u\cdot d$ 为 UP-DAG 树联接 DOWN-DAG 树形成 UDDAG 树时最坏情况下的计算复杂度。序列比对异常度求解的时间复杂度为 $O(m\cdot n)$，其中 m,n 为参与比对的序列长度，浏览时间异常度及请求循环异常度求解的时间复杂度均为 $O(1)$，故 ADA_ MFSP 检测模型的时间复杂度为 $O(L\cdot u\cdot d\cdot\log_2 k+m\cdot n)$。由实验情况可知，无论训练阶段还是检测阶段 u,d,m,n 的取值都较小，而应用本文提出的 SDD-UDDAG 更新算法，L 的取值在检测阶段也远远小于训练阶段，所以本算法可应用于在线检测。

4.3 与其他算法的对比

(1)由实验结果可知，ADA_MFSP 检测模型对各类 App-DDoS 攻击都有较好的检测效果，实现了透明检测。而文献[4]提出的基于 Session 检测方法未考虑到资源消耗型攻击的检测，文献[5]提出的基于 HsMM 的方法无法抵御 Session Flood 这种较隐蔽的攻击。

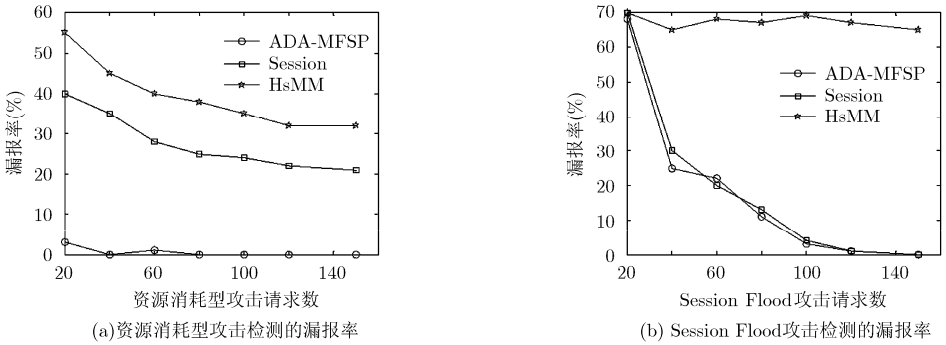
(2)文献[7]提出的基于会话异常度模型的方法需联合页面转移异常度等 6 个异常属性才能达到对 App-DDoS 洪泛攻击的检测，而 ADA_MFSP 检测模型仅需 f_{sa} , f_{ta} , f_{ca} 3 个异常度便可实现对各类 App-DDoS 攻击的检测。

(3)检测灵敏度分析。与文献[5]，文献[7]的工作相比较，分析资源消耗型、Session Flood 攻击的漏报率随攻击请求数的变化关系，如图 6 所示。Forged-URL Flood, Random-URL Flood, Single-URL Flood, Multi-URL Flood 与 Random-URL Flood 也可做类似分析。

由图 6 可知，对于资源消耗型攻击，当攻击请求数为 20 时，ADA_MFSP 的漏报率便可降至 4%，相对于 Session 模型 40%及 HsMM 模型 55%的漏报率，ADA_MFSP 模型的检测灵敏度显然具有明显优势。而对于 Session Flood 攻击，ADA_MFSP 模型同 Session 模型都采用浏览时间异常度及请求循环异常度进行检测，故检测灵敏度相当，优于 HsMM 模型。

5 结论

针对当前基于 Web 日志分析的检测方法无法实现对各类 App-DDoS 攻击的准确透明检测，本文提出了基于最大频繁序列模式挖掘的检测模型 ADA_MFSP。模型首先采用 SDD-UDDAG 挖掘、更新算法挖掘出正常 WASD 与待检测 WASD 的最大频繁序列模式集 P'' , P_d''，在滑动窗口实时设定的基础上采用动态规划算法求得 P'' , P_d'' 之间的序列比对平均异常度，联合浏览时间平均异常度、请求循环平均异常度达到对各类 App-DDoS 攻击的异常检测。由于 SDD-UDDAG 更新算法良好的时间性能及检测灵敏度，使得 ADA_MFSP 检测模型可实现对各类 App-DDoS 攻击的透明在线检测。



(a)资源消耗型攻击检测的漏报率　(b) Session Flood攻击检测的漏报率

图 6 漏报率比较

参 考 文 献

[1] Durcekova V, Schwartz L, and Shahmehri N. Sophisticated denial of service attacks aimed at application layer[C]. ELEKTRO, Rajeck Teplice, 2012: 55–60.

[2] Renuka Devi S and Yogesh P. A hybrid approach to counter application layer DDoS attacks[J]. *International Journal on Cryptography and Information Security*, 2012, 2(2): 45–52.

[3] Zade R and Patil H. A survey on various defense mechanisms against application layer distributed denial of service attack [J]. *International Journal on Computer Science and Engineering*, 2011, 3(11): 3558–3563.

[4] Ranjan S, Swaninathan R, and Uysal M. DDoS-shield: DDoS-resilient scheduling to counter application layer attacks[J]. *IEEE/ACM Transactions on Networking*, 2009, 17(1): 26–39.

[5] Xie Yi and Yu Shun-zheng. Monitoring the application-layer DDoS attacks for popular websites[J]. *IEEE/ACM Transactions on Networking*, 2009, 17(1): 15–25.

[6] Xie Yi and Yu Shun-zheng. A large-scale hidden semi-markov model for anomaly detection on user browsing behaviors[J]. *IEEE/ACM Transactions on Networking*, 2009, 17(1): 54–65.

[7] 肖军, 云晓春, 张永铮. 基于会话异常度模型的应用层分布式拒绝服务攻击过滤[J]. 计算机学报, 2010, 33(9): 1713–1724.
Xiao Jun, Yun Xiao-chun, and Zhang Yong-zheng. Defend against application-layer distributed denial of service attacks based on session suspicion probability model[J]. *Chinese Journal of Computers*, 2010, 33(9): 1713–1724.

[8] Duan Jian-li and Liu Shu-xia. Research on Web log mining analysis[C]. International Symposium on Instrumentation & Measurement, Sensor Network and Automation, Sanya China, 2012: 515–519.

[9] 阮幼林. 频繁模式挖掘算法及在入侵检测中的应用研究[D]. [博士论文], 华中科技大学, 2004.
Ruan You-lin. Research of mining algorithms of frequent patterns and their applications in intrusion detection[D]. [Ph.D. dissertation], Huazhong University of Science and Technology, 2004.

[10] Singh D K, Sharma V, and Sharma S. Graph-based approach for mining frequent sequential access patterns of Web pages [J]. *International Journal of Computer Applications*, 2012, 40(10): 33–37.

[11] Chen Jin-lin. An updown directed acyclic graph approach for sequential pattern mining[J]. *IEEE Transactions on Knowledge and Data Engineering*, 2010, 22(7): 913–928.

[12] Chordia S and Adhiya P. Grouping Web access sequences using sequence alignment method[J]. *Indian Journal of Computer Science and Engineering*, 2011, 2(3): 308–314.

李锦玲：　女，1986 年生，硕士生，研究方向为网络安全、异常流量检测.

汪斌强：　男，1963 年生，教授，主要研究方向为宽带信息网络、路由与交换技术.