基于 AGD 的恶意域名检测

臧小东[1,2,3]，龚俭[1,2,3]，胡晓艳[1,2,3]

（1. 东南大学网络空间安全学院，江苏 南京 211189；
2. 东南大学江苏省计算机网络重点实验室，江苏 南京 211189；
3. 东南大学教育部计算机网络和信息集成重点实验室，江苏 南京 211189）

摘 要：提出了一种聚类和分类算法相结合的恶意域名检测思路，首先通过聚类关联，辨识出同一域名生成算法（DGA, domain generation algorithm）或其变体生成的域名，然后分别提取每一个聚类集合中算法生成域名（AGD, algorithmically generated domain）的 TTL、解析 IP 分布、归属、whois 的更新、完整性及域名的活动历史特征等，利用 SVM 分类器过滤出其中的恶意域名。实验表明，该算法在不需要客户端查询记录信息的情况下即可实现准确率为 98.4%、假阳性为 0.9%的恶意域名检测。

关键词：网络安全监测；域名生成算法；命令与控制服务器；算法生成域名

中图分类号：TP393

文献标识码：A

doi: 10.11959/j.issn.1000−436x.2018116

Detecting malicious domain names based on AGD

ZANG Xiaodong[1,2,3], GONG Jian[1,2,3], HU Xiaoyan[1,2,3]

1. School of Cyber Science and Engineering, Southeast University, Nanjing 211189, China
2. Jiangsu Provincial Key Laboratory of Computer Network Technology, Southeast University, Nanjing 211189, China
3. Key Laboratory of Computer Network and Information Integration of Ministry of Education, Southeast University, Nanjing 211189, China

Abstract: A new malicious domain name detection algorithm was proposed. More specifically, the domain names in a cluster belonging to a DGA (domain generation algorithm) or its variants was identified firstly by using cluster correlation. Then, these AGD (algorithmically generated domain) names' TTL, the distribution and attribution of their resolved IP addresses, their whois features and their historical information were extracted and further applied SVM algorithm to identify the malicious domain names. Experimental results demonstrate that it achieves an accuracy rate of 98.4% and the false positive of 0.9% without any client query records.

Key words: network security monitoring, domain generation algorithm, command and control server, algorithmically generated domain

1 引言

作为互联网的重要基础设施，DNS 承载着域名与 IP 的双重映射，电子邮件、网站服务、微博等多种应用都与之相关。与此同时，各种恶意活动也随之诞生，如僵尸网络[1]、垃圾邮件、钓鱼网站等利用 DNS 解析域名获取回连服务器的 IP 地址，隐藏和躲避僵尸代理背后的命令与控制（C&C, command and control）服务器，然后回连该服务器，接收指令和回传窃取的信息。由此可见，域名对僵尸网络等恶意活动的构建和发动攻击起着重要的作用。因此，有效检测和识别出这些恶意活动使用的域名，对发现和防范其传播具有重要意义。

早期的恶意代码通常在程序中预置 IP 地址作为其 C&C 的地址，这种现象经常导致单点故障

收稿日期：2017−10−09；**修回日期**：2018−06−21

基金项目：国家自然科学基金资助项目（No.61602114）

Foundation Item: The National Natural Science Foundation of China (No.61602114)

问题。随后攻击者采用 IP-Flux 技术频繁切换 IP 地址躲避拦截，但该技术使用的域名固定，难逃域名黑名单的封堵。为了进一步提高逃逸和生存能力，最近攻击者采用域名生成算法生成域名，该算法使用一个特定参数（日期、时间、知名网站的内容）作为种子初始化一个随机算法，生成大量域名，并选择部分进行注册，如 Conficker-A/B/C 僵尸、Kraken botnet 等[2]。本文将不同 DGA 生成的域名统称为 AGD。为了有效区分这些 AGD，研究人员开始逆向分析恶意代码[3-4]，记录和封堵当前和后续生成的域名，但是当攻击者采用加密、周期性更新代码等混淆技术时，这种方法费时费力。

近年来，研究人员致力于通过分析 DNS 的活动行为进行恶意服务检测，算法的关键在于挖掘出合法活动区别于恶意活动的特征。相关研究存在 2 种思路：1) 围绕本地缓存名字服务器获取的下层 DNS 流量开展，这种思路认为同一恶意代码感染的僵尸主机具有相似的访问模式，研究人员分别从解析成功和解析失败的域名出发，分析客户端用户对域名的查询行为[5-10]并进行辨识；2) 分析上层 DNS 流量，借用下层类似模型分析 DNS 活动，测量缓存名字服务器中的 IP 地址集的相似性、空间分布等特征[11-13]。研究发现，上述研究成果多集中在恶意域名的检测方面，而对生成的 AGD 属于哪一种 DGA 的研究较少；另外发现，上述算法各有所长且有着不同的适用范围，如文献[7]提出的 Pleiades 系统对具有 Fast-Flux 特性的域名具有较低的识别率；而文献[11]提出的 Kopis 仅识别某个特定顶级域的恶意域名；尽管文献[14]对生成的 AGD 属于哪一种 DGA 进行了研究，但该算法没有区分 AGD 中的恶意域名，实测过程中发现该算法存在较多误报。

为了弥补上述研究中存在的不足，并考虑到相关检测信息收集的难度和成本，本文提出了一种新的检测思路。具体地，通过对流经中国教育科研网江苏省网边界的上层 DNS 解析流量的实际观察，本文发现同一 DGA 生成的大量域名映射到相同的 IP 集合，本文认为攻击者拥有一个相对稳定的 IP 集合，映射到这些 IP 集合上的域名有可能是一种 DGA 生成的，为此，本文将解析到相同 IP 集合的域名聚类，这些 IP 很有可能是恶意代码的 C&C 服务器。另外发现，不同 DGA 生成的域名在字面组成上具有可区分性，根据这种现象，本文提取域名的字面特征，将字面特征相近的域名聚类。然后对 2 种聚类集合进行关联性分析，进一步过滤掉噪声，使每一个聚类集合中的域名更贴近同一 DGA 或其变体生成的域名。然后针对每一个聚类集合，挖掘出域名解析 IP 的使用位置、归属特征、whois 特征、TTL 及域名活动历时等特征，并在此基础上利用 SVM 算法过滤掉合法的域名。

本文的贡献如下所示。

1) 通过聚类关联有效识别同一 DGA 或其变体生成的域名。尽管使用聚类算法对域名分簇已十分常见，但将不同聚类算法关联起来，通过计算 2 种不同聚类算法输出结果的 Jaccard index 得分来识别同一 DGA 或其变体生成的域名在现有文献中暂未发现。与文献[14]相比，本文算法在识别同一 DGA 生成的域名方面较全面、准确，还可以区分出其中的恶意域名。

2) 提出了新的恶意域名检测测度，如域名解析 IP 使用位置、归属、whois 的完整性及逆向 IP 查询得到的二级域名数。实验表明，利用本文提出的测度结合 TTL、域名解析 IP 的分布及域名活动的历史信息等旧的测度可以有效地辨识出 DGA 生成的恶意域名及 Fast-Flux[15-17]域名，并有着较高的检测精度。与文献[6]相比，本文算法不需要客户端的 IP 地址参与，不会因 DNS 缓存机制屏蔽用户的查询行为导致检测率低。

2 相关工作

越来越多的攻击者利用 DNS 这种互联网的重要基础设施作为纽带从事非法活动，如新型僵尸网络等利用 DNS 服务隐藏其命令与控制服务器，提高生存性，其行为可分为感染阶段、C&C 服务器招募阶段、攻击阶段及维护阶段，如图 1 所示。由此可见，识别出这些恶意代码使用的域名，可以实现事前检测和拦截恶意服务。近年来，研究人员通过分析 DNS 的活动行为检测恶意服务。由于在实际监测过程中观察到 DNS 流量的覆盖范围不同，本文将 DNS 的解析流量分为上层流量和下层流量，如图 2 所示。其中，下层流量指实际所有 DNS 解析流量（包括访问缓存的流量），上层流量指访问 DNS 服务器的 DNS 解析流量。

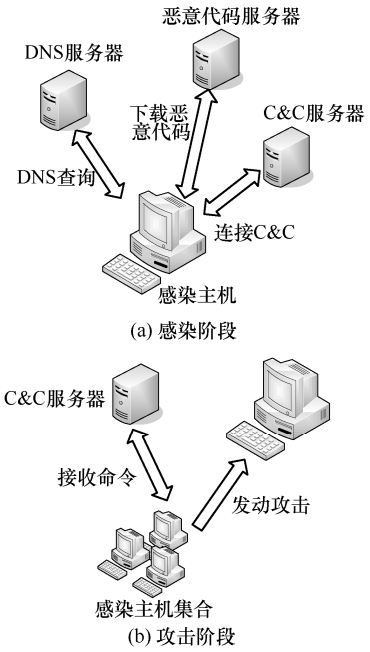

图 1　僵尸网络

由于在接入网边界采集的 DNS 解析流量中不包含访问本地缓存的 DNS 解析流量，因此从接入网边界观察到的是上层 DNS 解析流量。研究思路分为 2 种，一种是围绕本地缓存名字服务器获取的下层 DNS 流量开展[5-10]，分析查询失败的域名，从中识别出 AGD。而文献[10]能从正确解析的域名出发提出动态的域名打分系统 Notos，该系统使用被动 DNS 的查询数据、网络及域名的 zone 特征，通过计算域名的得分情况来确定该域名的合法性。

另一种思路围绕上层 DNS 流量开展，借用下层 DNS 活动中的类似模型辨识出其中的恶意域名[11-13]。相关研究工作较少，代表性的方法有文献[11]提出的 Kopis 检测系统，该系统通过分析顶级域名服务器和权威服务器的数据以及全球域名的查询和响应行为特征进行识别，但是该系统仅可识别某个特定顶级域的域名。由于从顶级域名服务器处观察到的本地缓存名字服务器数量庞大且地域分布广泛，因此会降低从本地缓存名字服务器视角观察 IP 地址集相似性及空间分布特征的上层 DNS 流量检测算法的准确率。文献[12]基于域名的依赖性和使用位置区分域名的恶意性，实测过程中发现存在大量误报。近年来，深度学习越来越受研究人员的关注，它通过组合低层特征形成更加抽象的高层表示属性类别或特征用来发现数据的分布。文献[18]提出一种基于域名历史数据的异常域名检测算法，能够有效识别出具有一定生存时间的可疑域名，且该方法在历史数据足够多的情况下比较高效。文献[19-20]提出了用模糊模式识别技术来识别被管理网络中的僵尸，该技术通过 DNS 检测和 TCP 检测等步骤识别恶意域名和 IP，但它使用的简易成员函数导致误报率高、实用性差。

分析发现，已有工作主要集中在恶意域名检测方面，而对生成的 AGD 属于哪种 DGA 研究较少。尽管文献[14]对这一方面进行了研究，实测发现该算法误报多且没有区分 AGD 的恶意性。

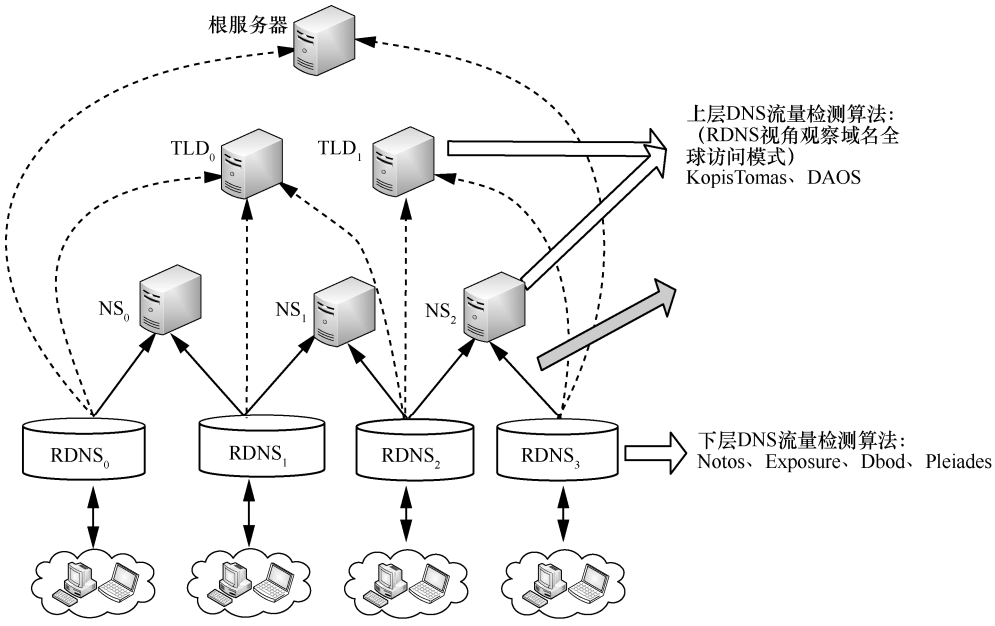

图 2　DNS 流量层次分类

另外，本文的研究对象是上层 DNS 流量经过基于词素算法[21]得到的 AGD，而文献[14]中研究对象涉及的域名包括正常域名和 DGA 生成的域名。由于 DNS 缓存机制的存在，它会屏蔽终端用户查询相关域名的 IP，即上层 DNS 流量无法看到终端用户信息，从而导致从终端用户角度出发分析客户端访问模式的相似性算法失效。本文算法不需要客户端查询记录，可以有效区分域名的恶意性并具有较高的检测精度。考虑到本文训练数据仅 1 万条而测试数据有 100 万条，相比之下训练样本较少，而 SVM 具有在训练过程中不需要太多的训练样本，即可实现较高精度的分类且能够处理高维的特征向量特点。鉴于其在本文的适用性，本文利用它过滤掉正常域名，并将其与 C4.5、*K*NN 等分类算法对比，发现 SVM 算法具有较高的准确率。

3 方法介绍

本文的目标是辨识出攻击者利用域名生成算法生成的恶意域名。为了达到这个目标，本文首先辨识出同一 DGA 或其变体生成的域名聚类，然后针对每一个聚类集合，提取能够有效区分域名恶意性的特征，过滤掉正常的域名集合。

域名和域。一个域名 *d* 由多个标签（点）拆分的字符串组成。最右部分为一个域名的顶级域名（TLD, top-level domain），然后是其二级域（2LD, second-level domain），依次类推。例如 www.baidu.com，顶级域为 com，二级域为 baidu，三级域为 www，而 baidu.com 为二级域名，其中，顶级域的标签不止一个，如.co.uk。

3.1 同一 DGA 生成的域名聚类

3.1.1 基于二分图的聚类

正常情况下，良性域名的一次 DNS 查询返回的解析 IP 往往是 1~3 个不等，而 DGA 或利用 Fast-Flux 逃逸技术生成的域名，其解析 IP 往往大于 4 个[22]，在聚类前，本文过滤掉解析 IP 低于 3 个的域名。然后将解析到相同或相似 IP 地址集合中的域名聚类，这些 IP 很可能是恶意代码使用的 C&C 控制器。设 2 个域名 α 和 β 在时段 E 对应的解析 IP 地址集合为 $S(\alpha)$ 和 $S(\beta)$，相似性得分计算式为

$$Sim(\alpha,\beta)=\frac{S(\alpha)\cap S(\beta)}{S(\alpha)\cup S(\beta)}\frac{1}{1+\mathrm{e}^{\gamma-\min(|S(\alpha)|,|S(\beta)|)}} \quad (1)$$

其中，$|S(\alpha)|$ 和 $|S(\beta)|$ 分别表示该域名对应的解析 IP 个数，式(1)中 $\frac{S(\alpha)\cap S(\beta)}{S(\alpha)\cup S(\beta)}$ 为集合 $S(\alpha)$ 和 $S(\beta)$ 的 Jaccard index，用于测量 2 个集合的相似性得分，$1+\frac{1}{\mathrm{e}^{\gamma-\min(|S(\alpha)|,|S(\beta)|)}}$ 作为权重因子，用于提升域名的相似度。尽管 $S(\alpha)\cap S(\beta)=1$、$S(\alpha)\cup S(\beta)=3$ 和 $S(\alpha)\cap S(\beta)=10$、$S(\alpha)\cup S(\beta)=30$ 的 Jaccard index 得分相同，但是，通过增加权重因子计算发现，后者的相似性大于前者。实验过程中将 γ 的值取为 3，即当 $\min(|S(\alpha)|,|S(\beta)|)$ 为 3 时，其相似度重系数为 0.5。

本文用二分图表示域名和 IP 的映射关系，利用谱聚类算法对相似域名聚类。设 $G=(V,E)$ 是一个无向图，顶点 V 可分割为 2 个互不相交的子集 (A, B)，其中，A 为域名集合，B 为 IP 集合，$K=|A|$、$L=|B|$ 分别表示域名及 IP 的个数。图中的每条边所关联的 2 个顶点 i 和 j 分别属于不同的顶点集 $i\in A, j\in B$。随后构建稀疏矩阵 $\boldsymbol{M}^{k\times l}$ 表示上述二分图，其中，k 为域名个数，l 为解析 IP 个数。如果存在域名到 IP 的映射则 $\boldsymbol{M}_{i,j}=1$，否则 $\boldsymbol{M}_{i,j}=0$。然后计算 2 个域名对应的解析 IP 的相似性，得到相似矩阵 $\boldsymbol{S}=\boldsymbol{N}\boldsymbol{N}^{\mathrm{T}}$。其中，$\boldsymbol{N}^{\mathrm{T}}$ 为规范化后的稀疏矩阵 $\boldsymbol{M}^{k\times l}$，详细算法如下所示。

算法 1 谱聚类算法对域名聚类

输入 邻接矩阵 $\boldsymbol{M}^{k\times l}$

输出 相似域名聚类 $(A_1, A_2, \cdots, A_n)$

1) 规范化邻接矩阵 $\boldsymbol{M}^{k\times l}, \forall j=1,2,\cdots,k, \sum_{i=1}^{l}\boldsymbol{M}_{i,j}=1$。

2) 计算相似矩阵 $\boldsymbol{S}=\boldsymbol{N}\boldsymbol{N}^{\mathrm{T}}$，$\boldsymbol{S}\in k\times k$，$\boldsymbol{N}^{\mathrm{T}}$ 为规范化后的稀疏矩阵 $\boldsymbol{M}^{k\times l}$。

3) 计算前 ρ 个特征值 $u_1, u_2, \cdots, u_\rho$，基于前 ρ 个特征值计算特征向量矩阵 $\boldsymbol{U}^{k\times\rho}$，其中，特征向量 $\boldsymbol{u}_i$ 表示简化后的 ρ 维的第 i 个域名。

4) 使用 k-means 算法对特征向量矩阵 $\boldsymbol{U}^{k\times\rho}$ 聚类。

3.1.2 基于域名字面特征的聚类

本文通过分析流经中国教育科研网江苏省网边界的上层 DNS 解析流量发现，不同 DGA 生成的域名在字面上要么由随机英文字符组成，要么由全数字和数字、英文字符混合组成，甚至一些高级恶意代码使用的 DGA 可以生成一系列可读的域名（类似于英文单词）。本文分别统计了域名的 *n*-元特征、

字符熵值、数字特征及有意义字符所占百分比，利用 *k*-means 算法对这些域名进行聚类。

n-元组特征。为了区分域名的可读性，本文分别提取每个域名对应 2LD 和 3LD 的两元、三元特征。针对每一域名的 2LD 和 3LD，本文分别统计其两元、三元出现次数的平均值方差及中位数，平均值计算方法参考式(2)，*n*-元组提取详见算法 2。

$$avg = \frac{\sum_{n\text{-}gram\ p\ \text{in}\ n\text{LD}} count(p)}{len(n\text{LD}) - n + 1} \tag{2}$$

其中，p 为 *n*-元组，$len(n\text{LD})$ 表示 *n*LD 长度。

算法 2　*n*-元统计值计算

输入　域名 *dname*、*n*、规范化后的英文字典

输出　域名中 *n*-元字符的统计次数 *sum*

1) 提取域名的 *n-gram* 字符串 *str*，并插入容器 *V*

2) 构建字典树

3) 遍历字典树，统计 *str* 在字典数出现的次数 *count*

4) *sum* += *count*

5) return *sum*

熵值特征。为了进一步区分域名中字符的分布情况，本文计算每个域名 2LD 和 3LD 中所含字符熵的变化情况。不同 DGA 生成的域名所包含字符的熵值变化范围往往不一致，本文可以利用这个测度进行区分，如式(3)所示。

$$E(n\text{LD}) = -\sum_{i=0}^{len(n\text{LD})} pro_i \text{lb}(pro_i) \tag{3}$$

其中，$len(n\text{LD})$ 表示域名 *n*LD 的长度，pro_i 是 *n*LD 中第 i 个字符对应的概率。

数字特征。同一 DGA 生成的域名具有较高的相似性，基于这个特点本文分别提取域名的长度特征、2LD 和 3LD 的长度、域名中 *n* 级域中 *n* 的取值及域名 2LD 和 3LD 域中数字字符所占的百分比特征。

有意义字符（单词）百分比。一些高级 DGA 为了提高抗逃逸能力，生成类似于人工生成的域名，与其他 DGA 相比具有可读性，基于这种现象，本文分别统计其 2LD 和 3LD 中有意义字符比 P_r 为

$$P_r = \max\left(\sum_{i=1}^{n} \frac{|w_i|}{len(n\text{LD})}\right) \tag{4}$$

其中，$|w_i|$ 表示 *n*LD 中有意义字符数，$|w_i| \geqslant 3$。

待提取到上述字面特征后，本文利用 *k*-means 算法对这些域名聚类，聚类结果用 $B=(B_k)$ 表示，其中，k=1,2,⋯,n。完成聚类后利用聚类裁剪和关联性分析实现聚类集合后处理。设 $X \in A_i$ 表示二分图的聚类集合，聚类裁剪的目标是去掉接近一对一关系的映射，这种映射关系与基于 DGA 产生大量域名而仅有少部分作为 C&C 控制器这一事实不符。若 X 符合 $\frac{|IP(X)|}{|X|} \geqslant \gamma$ [14]，则将其裁剪掉。聚类关联有助于进一步减少噪声和聚类数量，使其输出更贴近同一 DGA 或其变体生成的域名。设 $X \in A_i$、$Y \in B_j$ 分别表示 2 种不同聚类算法的输出，$I_{i,j}$ 集合是 X 和 Y 域名集合交集，$U_{i,j}$ 为 X 和 Y 域名集合并集。本文计算 2 个域名集合的 Jaccard index 得分 δ，其中，$\delta = \frac{I_{i,j}}{U_{i,j}}$，若 $\delta > \theta$ 则合并 2 个聚类集合，将其视为同一 DGA 生成的域名。图 3 统计了聚类关联后不同类别 AGD 所占百分比。横坐标分别表示使用英语单词生成的具有可读性的域名，使用动态 DNS 生成的域名，使用 26 个英文字符随机生成的域名，使用数字组成的域名，将数字、英文字符混合构成的域名以及混合域名（包括连接符、数字、字母等）。

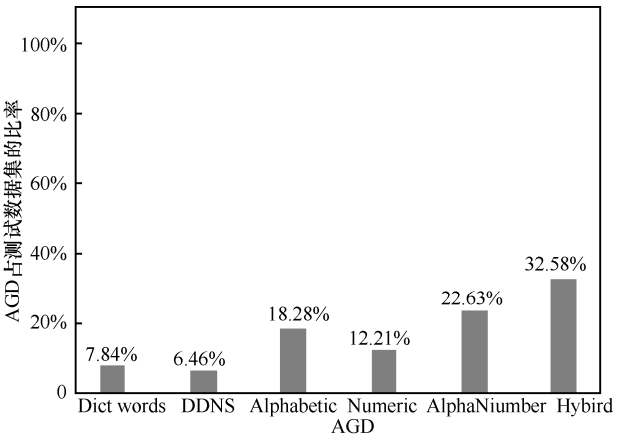

图 3　AGD 占测试数据集的比率

3.2　恶意域名识别

聚类关联后，本文用 $C=(C_k)$ 表示关联后的域名聚类，其中，k=1,2,⋯,n。研究发现，聚类集合中的域名既包含恶意域名，也包括正常域名。如阿里云盾的 Challenge Collapsar/DDoS 攻击防御方法，首

先网站提供者需要为真实使用的域名绑定一个别名，该别名由阿里云盾使用机器算法自动生成，当用户通过浏览器访问该域名时，会根据域名解析出的别名先去访问云盾服务器，而后才能访问源服务器。一旦出现 Challenge Collapsar/DDoS 攻击，攻击流量会经过云盾节点，从而触发清洗机制防御攻击，由此可见 DGA 生成的正常域名也会夹杂在上述聚类中。为了进一步有效辨识出其中的恶意域名，本文分析了正常域名（Alex 前 1 万）及恶意域名行为特征[7-11]，分别提取域名的 TTL 特征、whois 特征、解析 IP 相关特征，利用 SVM 算法过滤掉正常域名。

域名的 TTL 特征。DNS 响应分组中包含 TTL，记录了域名服务器将域名的解析信息保留在缓存中的最长时间。正常域名的 *TTL* 值一般为 1~5 天[23]，恶意域名为了提高其逃逸能力往往设置较短的 *TTL* 值。统计分析样本数据中的正常域名和恶意域名的 *TTL* 值发现，正常域名中大约 30%的域名 *TTL* 值大于 1 天，60%以上的域名 *TTL* 值大于 1 200 s；而恶意域名的 *TTL* 值较小，通常小于 500 s，大概占 45%左右，而小于 1 000 s 的恶意域名更是占总量的 70%以上。另外，正常网络服务为了减少开销，使用 Round-Robin DNS 和内容分发网络（CDN, content delivery network）实现负载均衡，其 *TTL* 值也比较低，如阿里云盾的 Challenge Collapsar/DDoS 攻击防御方法使用 DGA 生成的域名为 CDN 使用。由于单一通过分析 *TTL* 值区分域名的恶意性会产生较高的误报率和漏报率，因此需要与其他测度结合分析。本文分别提取域名 *TTL* 的平均值、标准方差、*TTL* 的变化次数及 *TTL* 变化范围的百分比作为区分测度。

域名的 whois 特征。whois 是当前域名系统中不可或缺的一项信息服务，通过它可以知道域名的解析 IP、注册人等相关信息。本文利用 virustotal 提供的 API 查询每个域名对应二级域名的 whois 信息，基于以往专家对域名的分析，正常域名的生命周期往往较长，且在这段时间内该域名提供的服务几乎都可以被访问，相比之下，botmaster 使用随机算法或使用 Flux 技术注册的域名生命周期短，便于频繁切换和更新，可以提高其生存性和隐蔽性。另外，正常域名为了提高知名度，在注册时拥有较完整的注册信息，而恶意域名为了掩人耳目，注册信息随机且不完整。表 1 统计了 virustotal 官网域名 whois 的主要信息项及含义。本文提取域名的注册时间、到期时间、更新时间信息，并分析了 whois 信息的完整度，分别统计了正常域名和恶意域名的 whois 信息的完整度。统计发现，域名的 whois 信息项大概有 50 多条，其中 93.28%正常域名 whois 信息项有 30 条左右，仔细分析发现 Registrant 的多数信息真实存在，而 90%以上的恶意域名的 whois 信息小于 20 条且提供的注册信息可读性差。然后统计了域名注册时间的变化情况，发现 87%以上的正常域名注册时间比较长，大概在 5 年以上，其中，60%还未更新；而 90%的恶意域名注册时间短，低于 3 年且 65%以上都已经到期，已经更新。同时还发现，正常域名伴随着生存时间的增长，域名更新次数较多，而大多数恶意域名的 whois 信息更新次数较少，因此可以有效区分。

表 1　样本域名 whois 的主要信息项及含义

域名 whois 信息项	含义
Registrant	注册人信息（姓名、单位、电话、E-mail、通信地址）
Administrative Contact	管理员信息（姓名、单位、电话、E-mail、通信地址）
Technical Contact	技术人员信息（姓名、单位、电话、E-mail、通信地址）
Registrar Name	注册商信息
Created Time	创建时间
Updated Time	最近更新时间
Expire Time	截止时间
Domain Server	域名服务器列表
Status	状态信息

解析 IP 特征。为了逃避检测，攻击者采用多种逃逸技术，如 Fast-Flux，确保其 C&C 控制器的存活性。此外，CDN 也将其域名映射到多个 IP 上以实现负载均衡，但是 CDN 背后拥有的服务器通常是高性能的专用服务器，其 IP 长期稳定活跃，而 Fast-Flux 感染的主机通常是普通用户且无法对其直接控制，这些 IP 短期活跃且不断更新。另外，CDN 为了提高服务质量，减少用户访问时间，采用就近原则提供服务，其 IP 集合分布广且均匀，而 Fast-Flux 则不是这样，其 IP 分布比较发散。基于上述特点，本文分析正常域名和恶意域名的解析 IP 的使用位置、归属信息。研究发现，恶意域名的解析 IP 分布比正常域名发散，本文查询实验室的 IP 活动库获取其解析 IP

所在城市数、归属地数。根据 IP 所在城市数、归属地数，计算解析 IP 的使用位置及归属地熵值。熵值越大，位置越发散，越接近恶意域名。尽管 Fast-Flux 网络通过频繁切换 IP 逃避检测，但域名固定仍然难以避免单点失效，DGA 通过生产大量域名，定位其 C&C 服务器，恰好弥补上述不足。反而言之，Fast-Flux 网络的 IP 会映射到多个域名上，统计分析所有解析 IP 地址到域名的映射，其中，CDN 中，每个解析 IP 反向映射到的二级域名数目超过 39 个的只有 17.3%，而 Fast-Flux 网络中占 42.5%；CDN 中，每个解析 IP 地址反向映射的二级域名数目超过 10 个的仅有 15.3%，而 Fast-Flux 网络却有 45.4%[12]。由此看出，CDN 中域名的解析 IP 地址到二级域名的映射数较少。本文还提取了逆向 IP 查询得到的 IP 到域名映射的数量作为一个测度，来区分 CDN 域名和 Fast-Flux 域名。

域名活动历史特征。本文采用基于网络的特征和基于 IP 黑名单的证据特征刻画域名的活动历史，其中，基于网络的特征描述分配给域名注册者的网络资源情况。研究发现，不法分子经常滥用其域名及 IP 资源，这些域名或 IP 通常表现为具有较短的生存时间和较高的切换频率，而正常用户的网络特征相对稳定。本文分别统计域名对应解析 IP 所属的 BGP 前缀和自制系统（AS, autonomous system）数，即 BGP(D)、BGP(2LD)、BGP(3LD)、AS(D)、AS(2LD)、AS(3LD)数及这些 BGP 前缀归属的国家数。正常域名对应的 BGP 前缀、AS 值较小，而诸如 Fast-Flux 等恶意域名由于常进行频繁的切换导致 BGP 前缀、AS 值较大。另外，域名资源相对于 IPv4 地址资源比较便宜，不法分子通常重用已知的恶意网络资源，如 IP 地址、BGP 前缀及 AS 等资源。本文分别统计域名对应解析 IP 地址 A(D)、BGP(A(D)) 和 AS(A(D))在 Spamhaus Block List[24]上的数目，来刻画待检测域名与已知恶意域名或 IP 的关联程度，数目越多，关联度越高，越接近恶意域名。

待提取到所有特征后，构建特征向量，使用机器学习算法过滤掉不符合要求的域名。为了实现较好的性能而不至于过拟合，本文采用 5 折交叉验证方法对训练数据进行训练，本文的训练数据包括 Alex 前 1 万的域名及文献[7-11]提供的恶意域名，然后分别比较 C4.5 决策树、*K*NN、Bayesian Network 和 SVM 等分类算法的测试结果，实验发现 SVM 算法具有较高的精度。

4　性能评估

本文所有实验均在一台两路 Intel Xeon 服务器上执行，每一路上具有一个 Intel(R) Xeon(R) CPU E5-2650 的处理器，每个处理器包含八核，主频 2.00 GHz，内存为 128 GB。算法的实现采用 C++和 Python 语言，本文在 Python 中使用了 NumPy 包，并利用 scikit-learn 机器学习库构建机器学习模型。

本节对恶意域名检测算法的检测效率、准确率进行评估，首先简要介绍本文检测系统的概况，然后验证算法中阈值 γ 的取值对聚类灵敏度的影响及算法精度，最后与已有的算法进行对比分析。

4.1　系统概述

本节为恶意域名检测系统提供了一个全面概况，主要包括预处理、聚类模块及分类模块 3 个部分，如图 4 所示。

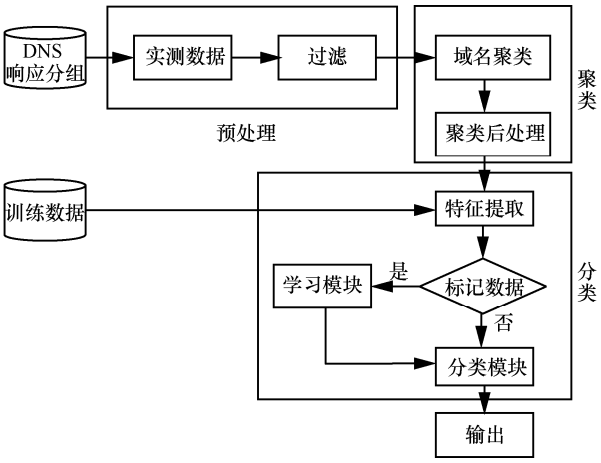

图 4　AGD 恶意域名检测系统概况

实测数据集。从 2017 年 2 月 12 日—26 日在中国教育科研网江苏省网边界采集到的能够正确解析的 DNS 交互分组，大约 100 万的不同域名集合。本文的恶意域名检测算法是基于域名本身所呈现的特性进行的。恶意域名本身的活动是一种特殊应用，这种应用无论在教育网还是商业网所呈现的流量特征是一样的。流量特征的不同和用户行为有关，而用户行为和网络中的应用是相关的[25]。不同的网络应用，呈现的流量特征也不尽相同，所以说本文的算法和结论可以用于大规模商用网络。

训练数据集。正常域名集，本文选取 Alex 网站连续三次排名前 1 万的域名集合（每月观测一次，从而保证其排名的稳定性），但由于 Alex 网站只提供二级域名，为了获取完整的域名，本文从实测域

名集获取与上述域名具有相同“二级域名标签”的域名构成合法样本集。恶意域名集合，本文选取僵尸网络、钓鱼网站、垃圾邮件和恶意软件等恶意域名，针对这些域名中没有三级域名标签的域名，本文采用与正常域名集相同做法构成恶意样本集。

预处理模块。提取 DNS 响应分组，剔除 DNS 查询不成功的域名，然后利用域名、IP 及不合法域名后缀黑名单对实测数据进行过滤，过滤掉已知恶意的域名。

聚类模块。分别将映射到相同解析 IP 地址集合的域名聚类和域名字面特征相近的域名聚类，分析 2 种聚类聚合的 Jaccard index 得分，识别同一 DGA 或其变体生成的域名。

分类模块。提取域名的 TTL、whois 更新、完整性特征、解析 IP 分布、归属等特征，利用 SVM 算法过滤掉正常域名。

4.2　聚类灵敏度及算法精度评估

完成聚类后，本文采用文献[14]方法评估的取值对聚类数目的影响，认为聚类越稳定，对其影响越小，而因 γ 的取值导致聚类数突然变化的情况可以作为决策边界。本文分别利用域名 2LD 和 3LD 长度取值范围 L_1、域名 2LD 和 3LD 数字比的取值范围 L_2 及域名公共后缀集合 L_3 这 3 组测度计算聚类集合内熵的平均值，该值用于表示聚类集合的一致性（如公共后缀）。当 γ=2.6 时，聚类数突然变化，本文把 γ=2.6 作为决策边界。聚类灵敏度参数取值分析如图 5 所示 γ 的取值范围为(0,2.6)，当 γ 不在(0,2.6)时，则不适合聚类裁剪。

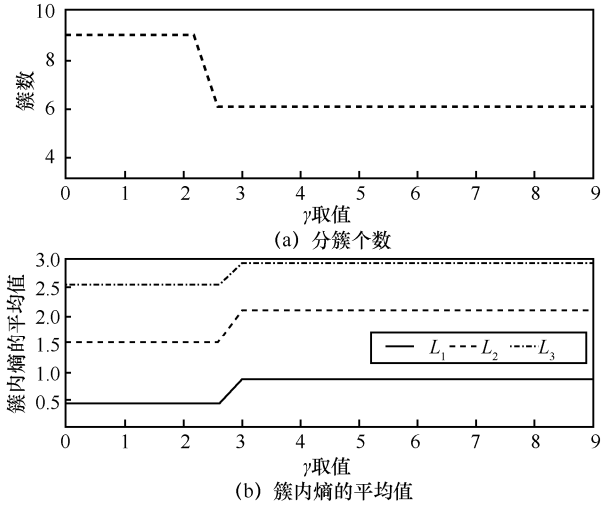

图 5　聚类灵敏度参数取值分析

随后本文提取 4 组特征：F_1 表示 *TTL* 特征、F_2 表示 whois 特征、F_3 表示解析 IP 地址特征、F_4 表示域名活动历史特征，构建 22 维特征向量用于识别未知恶意域名。为了达到较好的效果，本文采用 5 折交叉验证方法获取训练模型，分别研究不同的特征组合下域名的识别情况，如图 6 所示。

实验发现，F_1、F_2、F_3、F_4 这 4 组特征组合能比两两或三者特征组合的方法实现较好的效果。随后对比 C4.5、*K*NN、Bayesian Network、SVM 分类算法，实验发现，SVM 具有较高的检测精度，其准确率达到 98.4%，误报率为 0.9%，如图 7 所示。

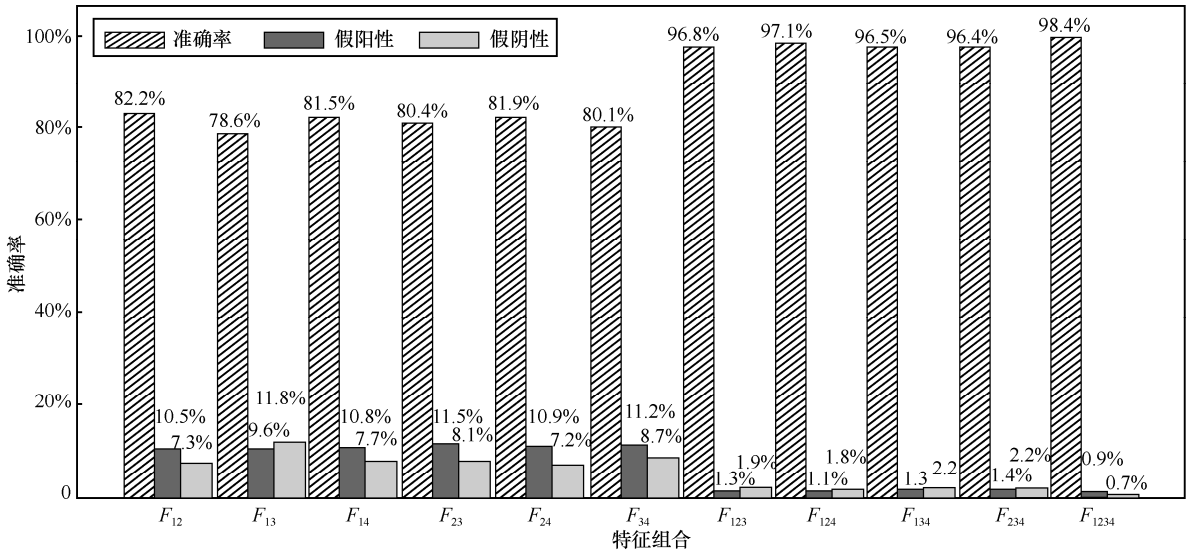

图 6　不同特征组合下准确率

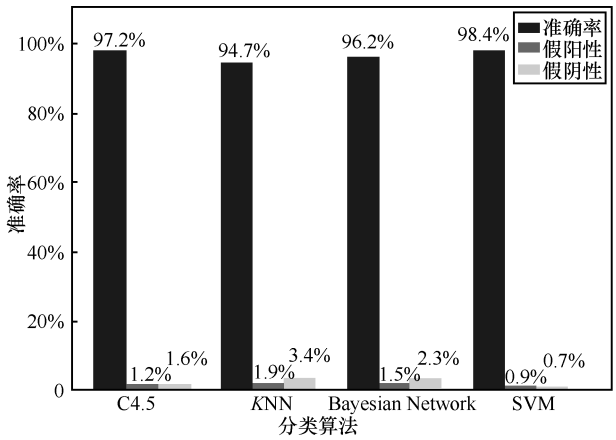

图 7　不同分类算法的准确率

4.3　监控窗口与准确率的关系

为了有效区分恶意域名，本文统计监控时间与准确率的关系。一般而言，监控窗口越长，信息越全，准确率越高，相反准确率降低。但是当窗口足够大时，数据量庞大，就会影响系统性能。为此本文考虑准确率与性能的权衡，重复上述实验，选择合适的监控窗口。本文在保持数据集不变的情况下，设定监控窗口为（3, 5, 8, 10, 13, 15）天，从准确率、假阳性和假阴性 3 个方面评估算法的检测精度。实验发现，算法的准确率随着观测窗口的长度增加而增大，但幅度不大。相应地，假阳性和假阴性有所减小，如图 8 所示。

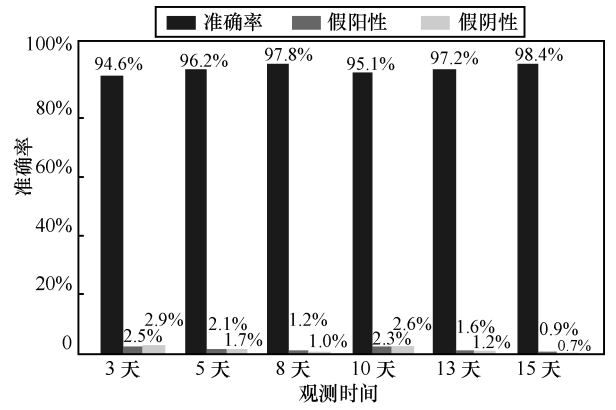

图 8　准确率与监控窗口的关系

当观测时间长度从 3 天增加到 8 天后，检测准确率从 94.6%上升到 97.8%，假阳性从 2.5%下降到 1.2%，假阴性也从 2.9%下降到 1.0%。当窗口继续增大时，准确率有所下降。进一步实验，本文设置第 8 天作为初始窗口，此时的准确率等同于窗口等于 3 的结果。本文观察一周的准确率变化情况，发现在第 15 天的准确率达到 98.4%，所以在实测过程中设定观测窗口为 7 天。

4.4　实测分析

完成训练后，利用本文提取的测度对实测数据进行检测发现，5 142 个可疑二级域名标签，进一步分析发现，其中 507 个二级域名所辖子域名出现于域名黑名单中。然后通过 McAfee Site Advisor 交叉验证剩余域名发现，4 375 个域名为恶意域名，剩余 260 个域名中 121 个域名无法正常访问，107 个域名对应于合法域名为误报。实测分析算法准确率可达到 97.84%，误报率为 0.88%，召回率为 77.5%，F 值为 86.5%，符合的训练结果。针对这些恶意域名本文进一步分析其对应的三级标签及解析 IP，研究发现 3 678 个为 DGA 域名，709 个存在 Fast-Flux 特性，剩余的域名部分为 Spam 和成人网站作为媒介进行恶意企图。

4.5　对比分析

本文首先将本文聚类关联算法（A_2）和文献[14]的算法（A_5）进行了对比，手动分析每一个聚类集合中 AGD 的识别情况。由于手动分析效率较低，为了验证本文算法和 A_5，每次随机选择 30 万个域名，进行了 5 次实验，取平均值作为比较值。在性能评估时，本文使用全部数据进行验证。如在聚类 1 中多数域名由随机字符和数字混合组成，本文认为这种域名是一种 DGA 生成的，如 blsh.sf123.com、cdn20.org、tl88.net。本文把其中的由随机字符组成或具有可读性的域名视为误报，如 0731131430.com、akamaiedge.net、outlook.com，把其他聚类中形如字符和数字混成的域名视为漏报，分别统计 2 种算法的正确率和漏报率，如图 9 和图 10 所示。

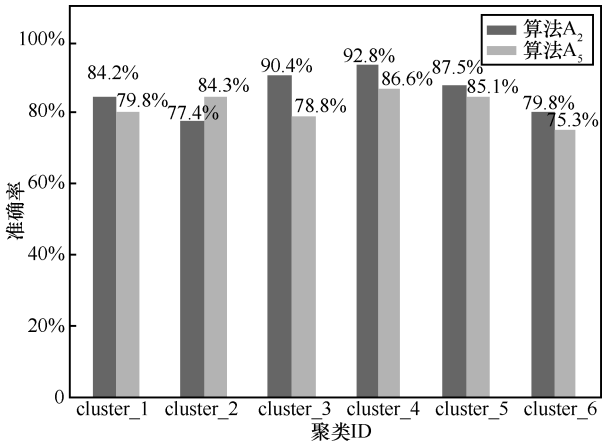

图 9　AGD 识别准确率

实验发现，本文算法准确率较高，漏报率相对较低，优于文献[14]。然后将本文算法（A_2）

分别与算法（A_3）[12]、算法（A_1）[23]和算法（A_4）[11]进行比较，分析在恶意域名检测方面的准确率和性能。实验发现本文算法与其他算法相比，准确率、假阳性、假阴性、召回率、*F* 值指标明显优于其他算法，这充分说明本文聚类和分类相结合的算法在分析 AGD 的恶意性方面更加有效，如图 11 所示。

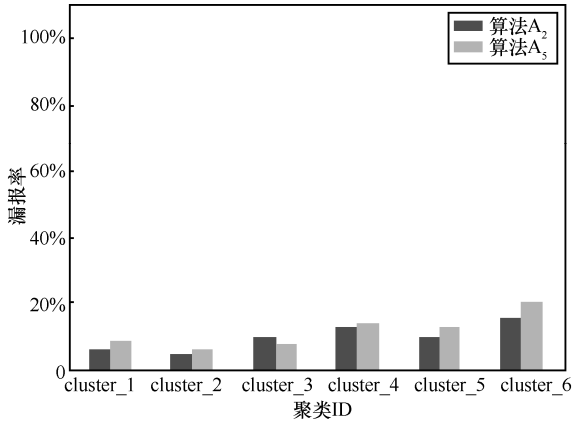

图 10　AGD 识别漏报率

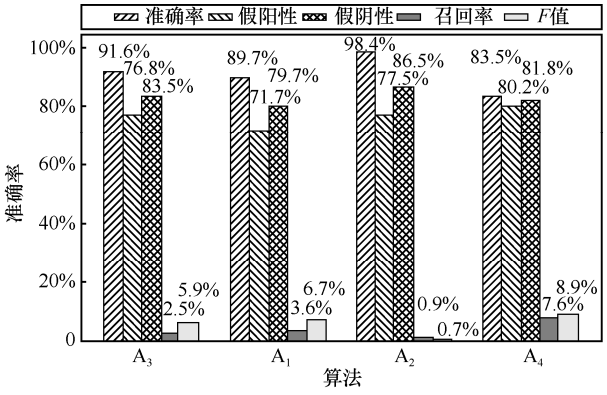

图 11　不同检测算法准确率

最后，本文分析了上述算法的系统开销。本文主要通过 virustotal 官网 API 获取 whois 信息，为了避免网络爬虫，virustotal 官网对其 API 的 API-key 进行了限制，一定程度上影响了系统性能。为了减少运算开销，本文在实验过程中将获取的 whois 信息存入 MySQL 数据库，重新执行本文算法。与其他算法相比，本文算法性能开销略微较大，但相差不大，另外，本文算法具有较高的检测精度，这充分说明，尽管本文算法不能完全取代其他检测算法，但一定程度上可以作为其他算法的补充，在运算开销几乎和其他算法差不多的情况下，可以准确地进行恶意域名辨识。不同检测算法的开销评估如表 2 所示。

表 2　不同检测算法的开销评估

算法	内存开销/MB	计算时间/s
A_3	临时内存 80.56，常驻 2.57	202.2
A_1	临时内存 80.14，常驻 2.64	205.1
A_2	临时内存 80.74，常驻 2.71	208.5
A_4	临时内存 80.76，常驻 2.71	208.6
A_5	临时内存 83.92，常驻 2.91	210.8

5　结束语

作为互联网的重要基础设施，DNS 在提供便利的同时，日益被攻击者利用来从事一些非法活动。本文设计了一种新的恶意域名检测方法，首先，定义不同的相似度标准，分别对域名聚类，利用 Jaccard index 分析不同聚类集合的关联性，辨识出同一 DGA 或其变体生成的域名，这是本文工作的一个特色，也是与其他工作的不同之处。其次，提取一些新的检测测度，并充分与其他特征相结合，对每一个聚类集合中的域名进行分类，过滤正常域名。实验发现，在不需要分析客户端 IP 对域名的查询记录及用户规模的情况下，本文算法便可有效区分出 AGD 中的恶意域名。与现有工作相比，本文算法在检测精度和检测效率方面有了很大提升。本文下一步的工作考虑设计实时性的恶意域名检测算法及恶意代码意图分析以实现秒级的响应。

参考文献：

[1] 江健, 诸葛建伟, 段海新, 等. 僵尸网络机理与防御技术[J].软件学报, 2012, 23(1): 82-96.
JIANG J, ZHUGE J W, DUAN H X, et al. Research on botnet mechanisms and defenses[J]. Journal of Software, 2012, 23(1): 82-96.

[2] YADAV S, REDDY A, REDDY A, et al. Detecting algorithmically generated malicious domain names[C]//The 10th ACM SIGCOMM Conference on Internet Measurement. 2010: 48-61.

[3] STONE G B, COVA M, CAVALLARO L. Your botnet is mybotnet: analysis of a botnet takeover[C]//ACM Conference on Computerand Communications Security (CCS).2009:635-647.

[4] DANIEL P, KHALED Y, MICHAEL K, et al. A comprehensive measurement study of domain generating mal-ware[C]//The 25th USENIX Security Symposium. 2016: 263-278.

[5] WANG T S , LIN H T , CHENG W T, et al. DBod: clustering and detecting dga-based botnets using DNS traffic analysis[J]. Computers & Security, 2017, 64:1-15.

[6] BILGE L , KIRDA E , KRUEGEL C , et al. Exposure: finding malicious domains using passive DNS analysis[C]//NDSS. 2011:1-17.

[7] ANTONAKAKIS M, PERDISCI R, NADJI Y. From throw-away traffic to bots: detecting the rise of DGA-Basedmalware[C]//Usenix Conference on Security Symposium.2012:24-40.

[8] SHARIFNYA R, ABADI M. DFBotKiller: domain-flux botnet detection based on the history of group activities and failures in DNS traffic[J]. Digital Investigation,2015,12(12):15-26.

[9] KHEIR N, TRAN F, CARON P, et al. Mentor: positive DNS reputation to skim-off benign domains in botnet c&c blacklists[C]//IFIP International Information Security Conference.2014:1-14.

[10] MANOS A, ROBERTO P, DAVID D, et al. Building a dynamic reputation system for DNS[C]//The 19th USENIX Security Symposium (USENIX Security'10). 2010:273-290.

[11] ANTONAKAKIS M, PERDISCI R, LEE W, et al. Kopis: detecting malware domains at the upper DNS hierarchy[C]//Usenix Conference on Security. 2011.

[12] 张维维, 龚俭, 刘尚东, 等. 面向主干网的 DNS 流量监测研究[J]. 软件学报, 2017, 28(9): 2370-2387.
ZHANG W W, GONG J, LIU S D, et al. DNS surveillance on backbone[J]. Journal of Software ,2017,28(9):2370-2387.

[13] THOMAS M, MOHAISEN A. Kindred domains: detecting and clustering botnet domains using DNS traffic[C]//Companion Publication of the International Conference on World Wide Web Companion. 2014:707-712.

[14] STEFANO S, FEDERICO M, LORENZO C, et al. Phoenix: DGA-based botnet tracking and intelligence[C]//International Conference on Detection of Intrusions& Malware.2014: 192-211.

[15] CELIK Z B, OKTUG S. Detection of fast-flux networks using various DNS feature sets[J]. Computers&Communications,2013:868-873.

[16] HUANG S Y, MAO C H, LEE H M. Fast-flux service network detection based on spatial snapshot mechanism for delay-free detection[C]//5th International Symposium on ACM Symposium on Information, Computer and Communications Security .2010: 101-111.

[17] ALMOMANI A. Fast-flux hunter: a system for filtering online fast-flux botnet[J]. Neural Computing& Applications, 2016: 1-11.

[18] 袁福祥, 刘粉林, 芦斌, 等. 基于历史数据的异常域名检测算法[J]. 通信学报, 2016, 37(10): 172-180.
YUAN F X, LIU F L, LU B, et al. Anomaly domains detection algorithm based on historical data[J]. Journal on Communications, 2016, 37(10): 172-180.

[19] WANGA K C, HUANGB C Y, LIN S J, et al. Fuzzy pattern-based filtering algorithm for botnet detection[J]. Computer Networks, 2011, 55(15): 3275-3286.

[20] WANG K, HUANG C Y, LIN S J, et al. A fuzzy pattern-based filtering algorithm for botnet detection[J]. Computer Networks the International Journal of Computer & Telecommunications Networking, 2011,55(15): 3275-3286.

[21] 张维维, 龚俭, 刘茜, 等. 基于词素特征的轻量级域名检测算法[J]. 软件学报,2016,27(9):2348-2364.
ZHANG W W, GONG J, LIU Q, et al. A Lightweight domain name detection algorithm based on morpheme features[J]. Journal of Software, 2016,27(9):2348-2364.

[22] LIN H T , LIN Y Y, CHIANG J W . Genetic-based real-time fast-flux service networks detection[J]. Computer Networks, 2013,57(2): 501-513.

[23] BILGE L, SEN S, BALZAROTTI D. Exposure: a passive DNS analysis service to detect and report malicious domains[J]. ACM Transactions on Information and System Security (TISSEC),2014,16(4):14-41.

[24] SHI Y, CHEN G, LI J T. Malicious domain name detection based on extreme machine learning[J]. Neural Process Letters,2017: 1-11.

[25] LI B D, SPRINGER J, BEBIS G, et al. A survey of network flow applications[J]. Journal of Network and Computer Applications, 2013, 36(2): 567-581.

[作者简介]

臧小东（1985−），男，山东济宁人，东南大学博士生，主要研究方向为网络安全、网络管理。

龚俭（1957−），男，上海人，博士，东南大学教授、博士生导师，主要研究方向为网络安全、网络管理。

胡晓艳（1985−），女，江西金溪人，博士，东南大学讲师，主要研究方向为网络体系结构、网络安全。