面向高速网络的递归域名服务器在线识别方法

刘晓梅[1,2]，孙永[1,2]，黄彩云[1,2,3]，刘洋[1,2]，邹学强[4]

(1.中国科学院信息工程研究所，北京 100093；

2.信息内容安全技术国家工程实验室，北京 100093；

3.中央民族大学，北京，100081；

4.国家计算机网络应急技术处理协调中心 北京 100029)

摘　要：针对获取活跃递归域名服务器的问题，利用网络流量测量技术对递归域名服务器识别技术进行研究，提出一种基于连接度估计的递归域名服务器在线识别方法。经实验验证，当选取合理连接度阈值时，该方法对活跃的递归域名服务器的识别具有较高准确率，可达 97%。对在线连接度估计算法进行误差分析和性能评估，结论表明连接度越高，其对应的估计值误差越低。与连接度精确测量方法相比，连接度估计算法内存占用小，更适用于高速骨干链路的流量在线分析。

关键词：网络安全，递归域名服务器，在线识别，连接度

中图分类号：TP302　　**文献标识码**：A　　**文章编号**：

Online recursive domain name server identification method for high speed backbone traffic

LIU Xiao-mei[1,2],SUN Yong[1,2],Huang Cai-yun[1,2,3],LIU Yang[1,2],ZOU Xue-qiang[4]

(1.Insititute of Information Engineering, Chinese Academy of Science, Beijing 100093, China;

2. National Engineering Laboratory of Information Security Technology, Beijing, 100093,China;

3. Minzu University of China,Beijing, 100081,China;

4. National Computer Network Emergency Response Technical Team/Coordination of China,Beijing 100029)

Abstract: In order to obtain active recursive domain name server, recursive domain name server identification technology was researched by adopting the network traffic measurement technology. We proposed an identification method of recursive domain name server based on connectivity estimation. According to our experiments, by selecting a reasonable threshold we can realized to identify recursive domain name server with a high accuracy which is up to 97% . At the same time, the error analysis and performance evaluation of the online connectivity estimation algorithm showed the higher connectivity, the lower error. Compared to the accurate connectivity measurement method, connectivity estimation algorithm took smaller memory and it is more suitable for the online analysis of high speed backbone traffic.

Key words: network security, recursive domain name server, online analysis ,connectivity estimation

---

收稿日期：　　；**修回日期**：

基金项目：中国科学院战略性先导科技专项基金资助项目（XDA06030200）

Foundation Items: Strategic Priority Research Program of the Chinese Academy of Sciences (XDA06030200)

1 引言

域名系统(Domain Name System)是互联网的核心基础设施,它通过域名服务器将域名和 IP 地址相互映射，使用户更便捷地访问互联网。目前域名服务器按服务类型可分为：根域名服务器、权威域名服务器和递归域名服务器。其中，递归域名服务器负责用户和域名系统的直接交互，在整个域名系统中处于重要的地位。然而，递归域名服务器面临严峻的安全威胁: 2014 年 3 月初，谷歌提供给大众的公用 DNS 服务器遭到长达 22 分钟的 DNS 劫持，所有使用该 DNS 服务的网络流量都遭到绑架，传到巴西和委内瑞拉境内[1]；2014 年 12 月，国内运营商递归域名服务器遭受了峰值流量高达 6Gbps 的 DDoS 攻击，造成各省的递归域名服务器延迟增大，核心解析业务受到严重影响[2]。因为递归域名服务器与用户之间是一对多的关系，因此当递归域名服务器受到攻击时，它会直接影响大量使用它的用户正常访问互联网，造成严重的网络故障。识别递归域名服务器的安全程度对维护递归域名服务器乃至整个域名系统的安全稳定都具有重要意义。

对递归域名服务器进行污染评估、DDos 攻击规模评估等是识别递归域名服务器安全程度的有效途径。获取准确的递归域名服务器是这些评估工作的前提和基础。目前，获取递归域名服务器 IP 地址主要有两种方式：主动方式和被动方式。主动方式是根据域名服务器的工作原理，主动发起域名请求方式获取[1]。虽然主动方式识别率较高，但主动探测前需要获取大量的 IP 地址且浪费较高资源。被动方式指不依赖具体的递归域名服务器，通过分析网络流量中递归域名服务器的相关特征进行识别，有效的克服了主动方式的缺点。但由于递归域名服务器和客户端使用相同的协议，两者之间的流量特征无明显区别，难以从网络流量中通过协议分析直接识别出递归域名服务器。现有的被动方式主要通过解析离线 DNS 流量，构造 DNS 流量图，通过分析图形中节点模式识别出递归域名服务器[2]。但由于离线流量无法及时反映网络状态，时效性较差，导致无法从中识别出活跃程度发生变化的递归域名服务器。因此亟需一种直接对在线流量进行分析，及时准确地识别出活跃度高的递归域名服务器的方法。但骨干网流量大，在线计算和存储资源有限，流量在线分析面临着极大挑战。

基于上述分析与考虑，本文提出一种基于连接度估计的递归域名服务器在线识别方法。其贡献如下：

1) 对 DNS 流量进行分析，挖掘出可用于识别递归域名服务器的特征：主机连接度和域名连接度。

2) 实现了一种流量中主机连接度和域名连接度的高效在线计算算法，并对在线计算结果进行误差分析。

3) 结合多种主动探测技术和 IP 定位技术，对不同连接度阈值下所识别出的递归域名服务器进行验证，获得具有较高识别准确率的连接度阈值。

2 相关研究与进展

2.1 DNS 背景知识

Root-DNS: 根域名服务器，域名解析系统中最高级别的域名服务器，负责返回顶级域名的权威域名服务器地址。

Authoritative DNS (ADNS): 权威域名服务器，负责管理某区域，对于该区域的域名查询直接从本地数据库查找并响应。

Recursive DNS (RDNS): 递归域名服务器，代替用户向权威域名服务器提出查询，解析权威域名服务器的响应信息，并向用户返回响应信息。它包括转发递归域名服务器(Forward RDNS,$RDNS_f$)和直接递归域名服务器(Direct RDNS,$RDNS_d$)。转发递归域名服务器不解析用户请求，只负责将用户请求转发给直接递归域名服务器进行解析。

当用户需要访问互联网上某一主机时，首先需要向 DNS 请求查询对方的 IP 地址。下面以用户查询域名 www.example.com 为例，说明 DNS 的工作模式，如图 1 所示；

1. 用户向 $RDNS_d$ 或 $RDNS_f$ 发起 www.example.com 的查询请求；若用户向 $RDNS_f$ 发起域名查询请求，$RDNS_f$ 将域名查询请求转发给 $RDNS_d$。
2. $RDNS_d$ 查询缓存中无该记录时，将请求转发给 Root-DNS；
3. Root-DNS 将 ADNS-com 的服务器地址转发给 $RDNS_d$；
4. $RDNS_d$ 再将请求转发给 ADNS-com；

[1] http://safe.it168.com/a2014/1017/1674/000001674600_all.shtml

[2] http://www.cww.net.cn/tech/html/2014/12/12/201412121010478633.htm

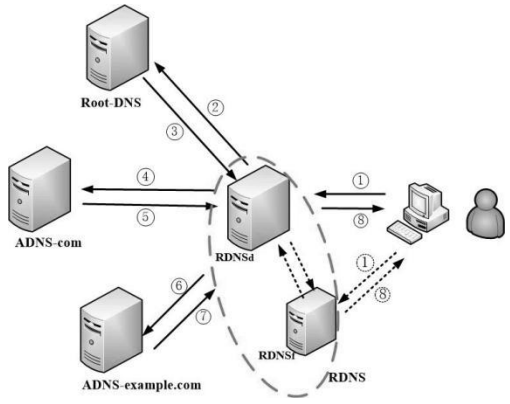

图 1　DNS 解析域名 www.example.com 的过程

5. ADNS-com 将 ADNS-example.com 的服务器地址转发给 $RDNS_d$;
6. $RDNS_d$ 再将请求转发给 ADNS-example.com;
7. ADNS-example.com 将域名 www.example.com 的解析 IP 返回给 $RDNS_d$;
8. $RDNS_d$ 缓存该解析结果;若用户直接向 $RDNS_d$ 请求，则 $RDNS_d$ 将应答 IP 直接返回给用户；若用户向 $RDNS_f$ 发起请求，则 $RDNS_d$ 将应答返回给 $RDNS_f$，再由 $RDNS_f$ 向用户响应。

2.2 递归域名服务器识别研究

依据是否需要构造特定的域名查询请求，目前识别递归域名服务器的方法可以分为两类：主动方式和被动方式。主动方式可分为两类：第一类方式中用户向预设的 IP 列表发送特定的域名查询请求，根据是否收到应答和应答类型对域名服务器进行分类，进而识别出递归域名服务器。第二类方式中，由于权威域名服务器只与递归域名服务器直接交互，通过获取与权威域名服务器直接交互的节点的 IP 地址集合，达到获取递归域名服务器的目的。文献[1, 3, 4]通过设定权威域名服务器 ADNS，利用 PlantLab[5]节点向设定的 IP 地址发起随机主机名的域名查询。权威域名服务器 ADNS 接收 DNS 请求的源 IP 集合就是递归域名服务器集合。但是主动方式也存在很多缺点：第一类方式中，主动探测前需要提前获取大量的 IP 地址；第二类方式中，当前很多递归域名服务器都进行安全设置，阻止某顶级或二级域名下任意主机的域名查询，域名查询请求可能被其过滤，导致主动方式耗费大量资源探测无响应结果的 IP 地址。

被动方式通过分析网络流量中递归域名服务器的相关属性特征进行识别，解决了主动方式受递归域名服务器部署限制以及需要提前获取大量 IP 地址的缺点。现有的被动方式主要解析离线 DNS 流量，但离线流量需要占据大量的存储空间，且无法反映 DNS 系统的最新状态。因此亟需一种从在线流量中及时准确地识别活跃的递归域名服务器的方法。网络流量测量技术是在线分析流量的基础，主要有两大类：抽样技术和数据流技术。抽样技术可分为分组抽样和流抽样。分组抽样的典型方法有系统抽样、随机抽样以及分层抽样[6]。He 等人[7]利用流量的自相似性改进了系统抽样，降低了抽样的系统开销。但是该方案没有考虑分组大小的多样性，针对这一问题，Raspall[8]提出了改进方案，使测量精度可以更少地依赖于流量的特征。但分组抽样中分组大小受传输技术限制，为消除这一限制，研究者们提出流抽样方法[9]。虽然抽样方法可以反映出流量的一些特征，但抽样方法仅分析部分流量，得到的结果存在一定误差。为了更加确切地获取原始流量的特征，研究者们提出了数据流方法。数据流方法使用有限的计算和内存资源对网络流执行一趟计算，是高速网络流量测量的重要方法，广泛应用于近似测量高速链路上流量统计信息，如熵估计、连接度估计等。其中，检测主机连接度是网络流量的一个重要测度。Zhao 等人[10]在抽样方法的基础上引入 Bitmap，通过 Bitmap 结构存储网络流中的报文信息，降低了内存资源的消耗，提高了算法的准确性。Li 等人[11]通过将 Bitmap 中的比特位由单一主机占有改进为多个主机共同使用的方式，提出了一种基于位共享的主机连接度检测算法，进一步节省了内存消耗。但多点共享 Bitmap 会带来主机映射冲突问题，Yoon 等人[12]通过为所有主机设置一个虚拟向量的方法，有效地解决这一问题。

3 基于数据流处理方法的连接度模型设计

定义 1：源 IP 连接度（*Src-con*）。在一定时间周期内，DNS 流量中同一源 IP 地址对应不同目的地址的个数。

定义 2：域名连接度（*Dom-con*）。在一定时间周期内，DNS 流量中同一源 IP 地址对应不同域名的个数。

3.1 域名系统内角色的连接度分析

域名系统中有用户、RDNS、ADNS 三种角色，*Src-con* 和 *Dom-con* 是角色的基本属性，而角色因功能不同会导致其基本属性不同。Src-con 与角色交互范围有关。RDNS 与用户、ADNS 直接交互。通

常用户规模较大，同时中国域名总数已超2060万[14]，管理不同域的ADNS数量庞大，RDNS交互范围广。ADNS仅与向它请求的RDNS交互，RDNS规模不及用户规模，交互范围较窄。用户仅与其配置的2～3个RDNS进行交互，交互范围窄。

Dom-con反映角色所交互的域名量。RDNS接收并响应其服务的所有用户的域名请求，同时，它也向不同域的ADNS转发域名请求，交互的不同域名量很大。用户仅向RDNS发起域名查询请求。ADNS仅向RDNS应答其管理域下的域名请求。与RDNS相比，后两者交互不同的域名量较小。

3.2 连接度估计模型：

下面以计算 *Src-con* 为例，介绍连接度估计模型的基本原理[12]。表1给出了模型所用到的符号定义。

表1 符号定义

符号	定义解释
B	所有不同源共享的位数组，元素个数为m，每个元素占1bit，所有元素初始为0，槽i被哈希映射则B[i] = 1
S_i	每个源独有的虚拟位向量，元素个数为s，每个元素占1bit，初始为0，槽j被哈希映射则$S_i[j] = 1$
R	随机数数组，元素个数为s
(src, dst)	每个数据包的源IP、目的IP对
H_m	哈希映射中使用的哈希函数，返回值小于m；哈希过程：$B[H_m$ (src XOR R[H_m (dst) mod s])] = 1

建立连接度估计模型，为每个节点建立虚拟位数组，利用哈希函数对所有源共享的位数组进行更新，计算每个节点的 *Src-con* 估计值。模型的伪代码见算法1，具体原理如下：

设 A_j 是 S_i 中槽 j 为空的事件，1_{A_j} 是事件 A_j 的随机变量；槽 j 为空，$1_{A_j} = 1$，否则，$1_{A_j} = 0$；

设 A_i 是 B 中槽 i 为空的事件，1_{A_i} 是事件 A_i 的随机变量；槽 i 为空，$1_{A_i} = 1$，否则，$1_{A_i} = 0$；

设 n 是测量周期内所有源不同连接度的总数，*k(Src-con)* 是某源对应不同目的地址个数；

U_m 是 B 中空槽个数，即'0'的个数，U_s 是 S_i 中'0'的个数，V_m 是B中'0'的比例，V_s 是 S_i 中'0'的比例，则 $V_m = U_m/m$，$V_s = U_s/s$；

$$U_s = \sum_{j=0}^{s-1} 1_{A_j} \tag{1}$$

$$\mathrm{E}(V_s) = \frac{1}{s}E(U_s) = \frac{1}{s}\sum_{j=0}^{s-1} E\left(1_{A_j}\right)$$

$$= \frac{1}{s}\sum_{j=0}^{s-1} Prob(A_j) = \left(1-\frac{1}{m}\right)^{n-k}\left(1-\frac{1}{s}\right)^{k}$$

$$\approx e^{-\frac{n-k}{m}}e^{-\frac{k}{s}} \quad \text{当}(\mathrm{n}-\mathrm{k}), \mathrm{m}, \mathrm{k}, \mathrm{s} \to \infty$$

$$\approx e^{-\frac{n}{m}-\frac{k}{s}} \quad \text{当} \mathrm{k} \ll \mathrm{m} \tag{2}$$

$$\hat{k} \approx -s * \frac{n}{m} - s * \ln(E(V_s)) \tag{3}$$

$$U_m = \sum_{i=0}^{m-1} 1_{A_i} \tag{4}$$

$$\mathrm{E}(V_m) = \frac{1}{m}E(U_m) = \frac{1}{m}\sum_{i=0}^{m-1} E\left(1_{A_i}\right)$$

$$= \frac{1}{m}\sum_{i=0}^{m-1} Prob(A_i) = \left(1-\frac{1}{m}\right)^{n} \approx e^{-\frac{n}{m}}$$

$$\text{当} \mathrm{m} \to \infty, \mathrm{n} \approx -\mathrm{m} * \ln(E(V_m)) \tag{5}$$

$$\hat{k} \approx s * \ln(E(V_m)) - s * \ln(E(V_s)) \tag{6}$$

虽然实际上 m、n 等都很大，但不是理想条件下的无穷大，模糊测量结果的误差可以表述为：

$$\sigma = \frac{|s*ln(E(V_m))-s*\ln(E(V_s))-k|}{k} = \left|s * \ln\left(\frac{1-\frac{1}{m}}{1-\frac{1}{s}}\right) - 1\right| \tag{7}$$

，当取 m=1M，s=200 时，σ =0.25%，误差很小，可以接受。

算法1　*Src-con* 估值算法

输入：*DNS Stream*

输出：*Conset{src,Src-con}*

```
1: function ESTIMATE_Src(DNS Stream)
2:     IPset{src, dst} ← GetStreamIP (DNS  Stream)
3:     HASH_TABLE  *h ← createhashtable()
4:     Initialize  B(m) ← 0
5:     Initialize  R(s) ← Random()
6:     while  NULL  != IPset  do
7:         i ← Hm(dst)
8:         j ← Hm(src  XOR  R[i])
9:         if Check_one(B, j) = false then
10:            B[j] ← 1
11:        end if
12:        Insert_hash_tableh(src)
13:    end while
14:    Vm ← B_zero_ratio(B, m)
15:    while NULL  != IPset  do
16:        for I = 1 → s  do
17:            j ←Hm(h → src  XOR  R[I ])
```

```
18:            if Check_one(B, j) = true then
19:                h → src.count_one + +
20:            end if
21:        end for
22:        Vs ← S_ zero_ratio(s,h→src.count_one)
23:        (h → src.Src-con) ← s ∗ (ln(Vm) –ln(Vs))
24:        PRINT   h → src, h → src.Src−con
25:        h ← (h → next)
26:    end while
27: end function
```

4 实验分析与验证

4.1 数据集介绍

根据 3.1 对域名系统中不同角色的连接度分析，从互联网捕获 DNS 流量，对连接度作为识别递归域名服务器的特征进行可行性验证。同时为验证该特征属于 DNS 流量中的共性特征，排除单一数据集对分析结果的影响，分别获取某教育网网关（EDU）和某企业网网关（ISP）24 小时内的 DNS 流量，如表 2 所示。对 DNS 数据集进行解析，获取流量中的源 IP、目的 IP、域名等属性。

表 2 数据集介绍

数据集	位置	日期	规模
EDU	北京某教育网网关	2014/05/20 9：40-2014/05/21 9：40	总 DNS 连接数为 1547531 合法 DNS 连接数 1525231 不同源主机数为 329
ISP	北京某企业网网关	2014/03/24 17：50-2014/03/25 17：50	总 DNS 连接数 677188 合法 DNS 连接数 651918 不同源主机数为 367

4.2 DNS 流量中 RDNS 特征识别

利用 IP 定位等技术，分别从 EDU 数据集和 ISP 数据集中获取 3 个确定 RDNS 的 IP 地址，如表 3、表 6 所示；通过辨识 DNS 应答包的权威标志识别 ADNS 的 IP 地址，随机选取 3 个确定 ADNS 的 IP 地址，如表 4、7 所示；同时随机选取 3 个确定客户端 IP 地址，如表 5、表 8 所示。同时，将数据集按 4h 为周期分为六个周期 T1~T6，在每个时间段 Ti(i=1,2,…6)和 24h 内，分别对 EDU 和 ISP 中所选定的 IP 地址精确计算 *Src-con* 和 *Dom-con*，并对结果进行对比分析。EDU 数据集在 T1~T6 和 24h 内的 *Src-con* 对比结果如图 2、3 所示，在 T1~T6 和 24h 内的 *Dom-con* 对比结果如图 4、5 所示。ISP 数据集在 T1~T6 和 24h 内的 *Src-con* 对比结果如图 6、7 所示，在 T1~T6 和 24h 内的 *Dom-con* 对比结果如图 8、9 所示。

由图 2 可知，各周期内，RDNS 8.8.8.*和 159.226.8.*的 *Src-con* 明显高于其他 IP。但 RDNS 114.114.114.* 在单个T_i周期内很难同 ADNS、Client 区分开来。进一步分析，我们发现 RDNS 114.114.114.*是流行度低的递归域名服务器，用户数较少，导致其 *Src-con* 较低。由图 3 分析，24h 内 EDU 数据集中 RDNS 的 *Src-con* 均明显高于其他 IP。同时对图 6、7 分析，结果与图 2、3 一致， 24h 内 ISP 数据集中 RDNS 的 *Src-con* 均明显高于其他 IP。故对 *Src-con* 选取合理的阈值，便可将 RDNS 识别出来。

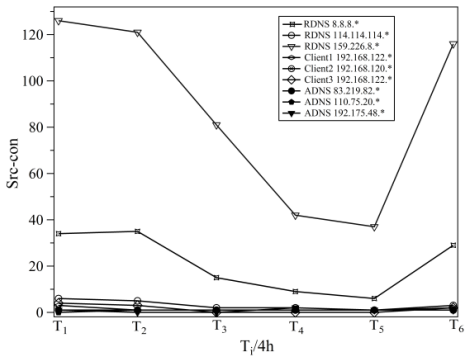

图 2 EDU: T1~T6 内 RDNS\Client\ADNS 的 Src-con 对比

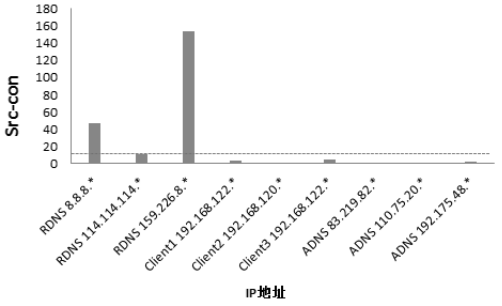

图 3 EDU: 24h 内 RDNS\Client\ADNS 的总 Src-con 对比

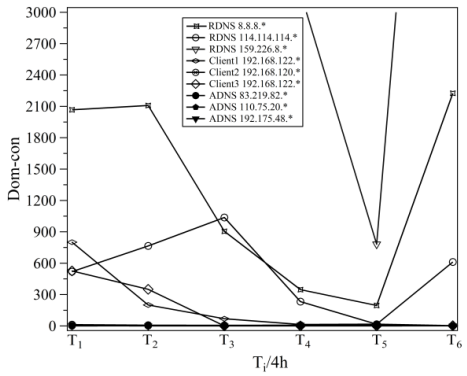

图 4 EDU: T1~T6 内 RDNS\Client\ADNS 的 Dom-con 对比

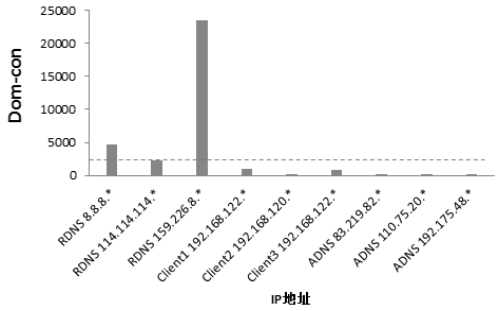

图 5 EDU: 24h 内 RDNS\Client\ADNS 的总 Dom-con 对比

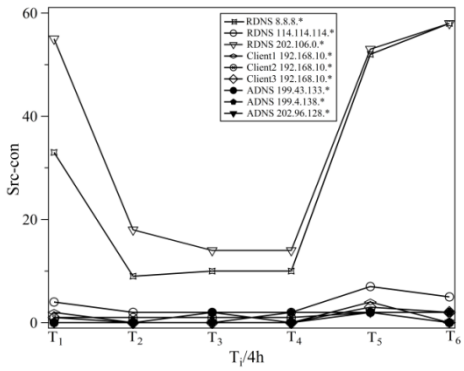

图 6 ISP: T1~T6 内 RDNS\Client\ADNS 的 Src-con 对比

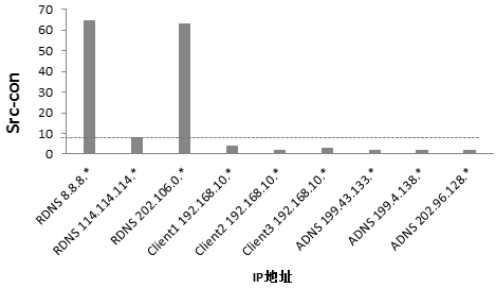

图 7 ISP: 24h 内 RDNS\Client\ADNS 的总 Src-con 对比

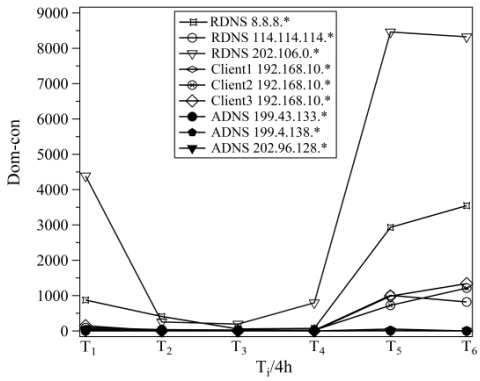

图 8 ISP: T1~T6 内 RDNS\Client\ADNS 的 Dom-con 对比

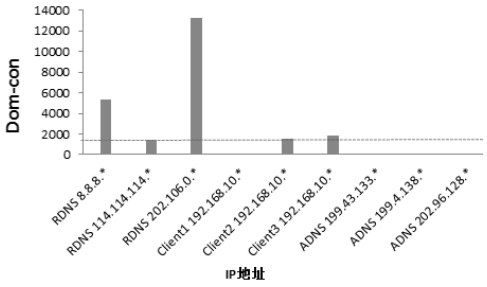

图 9 ISP: 24h 内 RDNS\Client\ADNS 的总 Dom-con 对比

对图 4 进行分析，RDNS 159.226.8.*和 8.8.8.*的 *Dom-con* 明显高于其他 IP。但分析不同周期内 114.114.114.*的 *Dom-con*，我们很难区别其与 Client、ADNS。由图 5 分析可知，RDNS 在 24h 内的 Dom-con 均明显高于其他 IP，对 *Dom-con* 选择合理的阈值，可识别出 RDNS。对图 8、9 进行分析，我们仍能明显区分出 RDNS 202.106.0.*和 8.8.8.*，但很难区分出流行度很低的 RDNS 114.114.114.*。因此，我们只能设置适当的 *Dom-com* 阈值来识别流行度高的 RDNS。

综合两个数据集的分析：

1）在挖掘 RDNS 的特征方面，同时具有较高的 Src-con 和 Dom-con 可作为识别特征。

2）在计算连接度周期方面，以 24h 为周期时，RDNS 的连接度明显高于其他主机。此外，虽然流行度低的 RDNS 难以识别，但由于其用户群体少，我们对其污染情况不做研究，另一方面，因为被动识别方法本身具有识别局部网络特征的特点，所以我们只关注识别结果的准确性。

3）在连接度阈值选取方面，存在合理连接度阈值对活跃的 RDNS 进行区分。

4.3 连接度估计误差分析

在 3.2 介绍的连接度估计算法中，估算误差除

了与m, n取值有关以外，还与哈希冲突有关。稳定、独立性较好的哈希算法可以使数据报文在哈希槽内尽可能地服从均匀分布，降低哈希冲突的概率，减小估算误差。由文献[13]可知，通过对不同数据集的验证测试，BOB 哈希具有较好均衡性、较高性能和较小误差。因此，本文进行连接度估计时选择 BOB 哈希处理 DNS 数据流。

对连接度估计误差进行计算时，选用从带宽为 5Gbps 骨干网链路上捕获的 24h 的 DNS 数据包作为测试数据集。经统计，该测试集中包含 7424594 个不同的源主机。对这些源主机分别计算 Src-con 的精确值S_{1i}和估计值S_{2i}。当某主机的精确连接度S_{1i}很大时，在连接度估计模型中，其虚拟位数组中'0'的比例为 0，由公式(6)以及对数的求值原理可知，该主机的连接度估计值S_{2i}为 Inf。此时，若S_{1i}在所有主机的S_1中排名前 1%时，S_{1i}和S_{2i}之间的误差可以约等于 0，否则误差为∞。设λ为S_1的阈值，δ为S_2的平均相对估计误差，δ的计算公式为

$$\delta = \frac{1}{n} * \sum_{i=1}^{n} \left(\frac{|S_{1i} - S_{2i}|}{S_{1i}} * 100\% \right) \quad (8)$$

n 为S_1大于阈值λ的主机数，δ随着 λ 的变化情况如图 10 所示。

表 3　EDU: RDNS 列表及对应 Src-con,Dom-con

RDNS	Src-con							Dom-con						
	T1	T2	T3	T4	T5	T6	24h	T1	T2	T3	T4	T5	T6	24h
8.8.8.*	34	35	15	9	6	29	47	2067	2108	905	347	196	2225	4735
114.114.114.*	6	5	2	2	1	3	11	519	765	1037	232	16	611	2213
159.226.8.*	126	121	81	42	37	116	154	13761	10454	7712	3185	782	9496	23513

表 4 EDU: ADNS 列表及对应 Src-con,Dom-con

ADNS-IP	Src-con							Dom-con						
	T1	T2	T3	T4	T5	T6	24h	T1	T2	T3	T4	T5	T6	24h
83.219.82.*	1	1	1	1	1	1	1	2	2	2	2	2	2	2
110.75.20.*	0	1	1	1	1	1	1	0	2	4	4	2	4	6
192.175.48.*	1	0	0	2	1	1	2	3	0	0	7	6	6	9

表 5 EDU: Client-IP 列表及对应 Src-con,Dom-con

Client-IP	Src-con							Dom-con						
	T1	T2	T3	T4	T5	T6	24h	T1	T2	T3	T4	T5	T6	24h
192.168.122.*	3	1	1	1	1	2	4	801	200	69	14	17	217	1008
192.168.120.*	1	1	1	1	1	1	1	12	7	5	7	7	4	16
192.168.122.*	4	3	0	0	0	2	5	526	350	0	0	0	288	836

表 6 ISP: RDNS 列表及对应 Src-con,Dom-con

RDNS	Src-con							Dom-con						
	T1	T2	T3	T4	T5	T6	24h	T1	T2	T3	T4	T5	T6	24h
8.8.8.*	33	9	10	10	52	58	65	875	408	57	72	2931	3545	5363
114.114.114.*	4	2	2	2	7	5	8	36	19	21	16	1007	821	1478
202.106.0.*	55	18	14	14	53	58	63	4387	254	193	797	8459	8325	13311

表 7　ISP:ADNS 列表及对应 Src-con,Dom-con

ADNS-IP	Src-con							Dom-con						
	T1	T2	T3	T4	T5	T6	24h	T1	T2	T3	T4	T5	T6	24h
199.43.133.*	0	0	2	0	2	0	2	0	0	3	0	8	0	8
199.4.138.*	0	0	0	2	2	2	2	0	0	0	2	2	2	2
202.96.128.*	0	0	0	0	2	0	2	0	0	0	0	4	0	4

表 8 ISP:Client-IP 列表及对应 Src-con,Dom-con

Client-IP	Src-con							Dom-con						
	T1	T2	T3	T4	T5	T6	24h	T1	T2	T3	T4	T5	T6	24h
192.168.10.*	2	0	0	0	4	0	4	53	0	0	0	54	0	67
192.168.10.*	1	1	1	1	2	2	2	94	38	25	21	724	1212	1524
192.168.10.*	1	0	0	0	3	2	3	148	0	0	0	986	1338	1848

由图 10 分析可知，随着λ的逐渐增大，δ逐渐减小，符合连接度估计模型的原理，即S_1越高，对其的估计误差越小。同时，S_{2i}为 *Inf* 的主机，其精确连接度值S_{1i}也很高，不存在误差为∞的情况。当λ=50 时，δ较高，为 30.59%；当λ=100 时，δ接近 10%；当λ=200 时，δ为 3.88%。当λ=300 时，δ仅为 1.71%。由于 RDNS 具有高的连接度，只有合理的高阈值才能将RDNS从所有主机中识别出来，因此我们只观察高连接度的估计误差。当λ>100 时，δ不足 10%，误差较小，可以接受。因此，S_2值越高越接近精确值，识别出 RDNS 的概率越大。

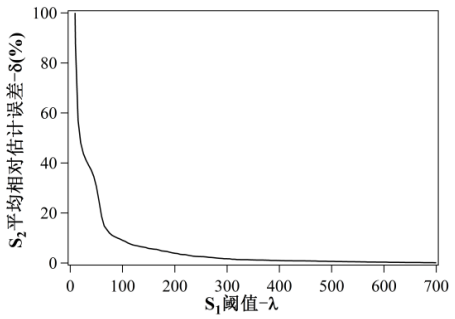

图 10　阈值 λ 与估计误差 δ 的关系

4.4 RDNS 在线识别与验证

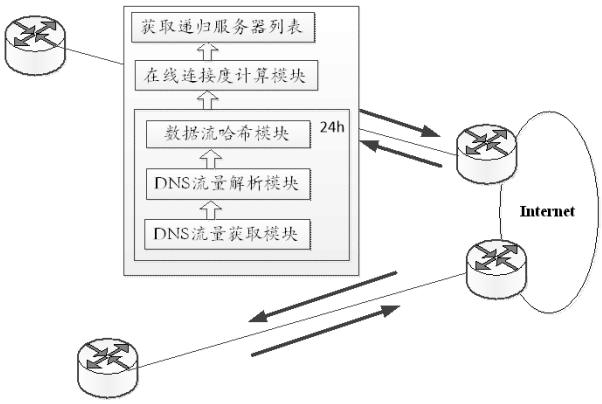

图 11　连接度模型在线部署图

由 4.2 节可知，RDNS 具有高 *Src-con* 和高 *Dom-con* 的特征。根据这一特征，将连接度统计模型部署到在线网络设备中，部署方法及环境如图 11 所示。DNS 流量获取模块从底层获取网络流量，之后通过协议分析获取 DNS 流量，将 DNS 流量上传给 DNS 流量解析模块，解析出源、目的 IP 地址与域名。数据流哈希模块通过对这三个属性的哈希映射来更新 *Src-con* 和 *Dom-con* 的共享数组，同时哈希存储源 IP 地址。读取流量 24h 以后，利用共享数组和源 IP 的哈希表对每个源 IP 分别计算 *Src-con* 和 *Dom-con*，将计算得到的 *Src-con* 和 *Dom-con* 分别取阈值，获取 RDNS 的交集列表集合。阈值的选取和识别 RDNS 的数目如表 9 所示。

对不同阈值下识别出的递归域名服务器结合多种主动探测[15]技术和 IP 定位技术进行验证。通过分析表 9 可知，当阈值选取合理时，基于连接度对 RDNS 的识别是有效的，最高准确率可达 97%。

4.5 连接度估计算法性能评估

将性能评估测试设备分为 2 组，一组运行连接度精确测量算法（每个源 IP 维护一组连接信息），另一组运行本文所使用的连接度估计算法，两组设备在同一个带宽为5Gbps的骨干网链路上同时进行在线测试。每隔 1min 对两个算法所使用的内存量进行统计。内存消耗对比如图 12 所示，连接度精确测量算法的内存消耗增速快，120min 后，达到连接度估计算法所消耗内存的两倍多。以此可以推断，24h 以后，连接度估计算法所占用的内存相比于精确测量方法会节省约 72%的内存资源仅占 2GB。因此，本文所提出的连接度估计算法具有节省内存的优点，更适用于高速骨干链路的流量在线分析。

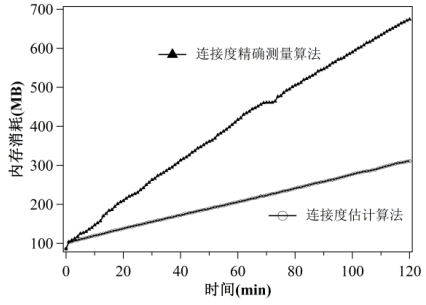

图 12 内存占用量对比

针对丢包率，连接度精确测量算法在 108min 时出现了千分之三的丢包率。而连接度估计算法经 953min 才出现同样大小的丢包率。因此，连接度估计算法处理能力更强。

表 9 RDNS 结果识别与验证

阈值		识别 RDNS 交集数量	验证通过的 RDNS 数量	准确率(%)
Src-con	Dom-con			
100	3000	642	437	68.07%
	4000	581	419	72.12%
	5000	523	393	75.14%
300	3000	459	425	92.59%
	4000	443	414	93.45%
	5000	412	388	94.17%
500	3000	437	423	96.80%
	4000	424	413	97.41%
	5000	395	387	97.97%

5 结论

对递归域名服务器进行污染评估是网络安全的重要问题，而准确识别递归域名服务器是对其执行污染评估的基础。针对主动方式需要提前获取大量 IP 地址、耗费网络资源，被动方式分析离线流量无法及时掌握递归服务器的状态等缺点，本文提出一种基于连接度估计的 RDNS 在线识别方法，实现了一种流量中主机连接度和域名连接度的高效在线计算算法。对识别结果验证表明，在选取合理连接度阈值的前提下，该方法对递归域名服务器的识别准确率较高。

但是，本文所提出的方法不能保证识别结果有较高的召回率。在今后的研究工作中，在保证算法较高的准确率和较好的处理能力的前提下，将通过从 DNS 流量中挖掘更多有效特征来提高识别结果的召回率。

参考文献：

[1] D. Dagon, N. Provos, C. P. Lee, and W. Lee, “Corrupted DNS Resolution Paths : The Rise of a Malicious Resolution Authority,”[C] in Proceedings of the Network and Distributed System Security Symposium, 2008, vol. 1.

[2] C. D. Cranor, E. Gansner, B. Krishnamurthy, and O. Spatscheck, “Characterizing Large DNS Traces Using Graphs,” [C] in Proceedings of the ACM SIGCOMM Workshop on Internet Measurement, 2001.

[3] C. Huang and D. A. Maltz, “Public DNS System and Global Traffic Management,”[C] in Proceedings of International Conference on Computer Communications.

[4] K. Schomp, T. Callahan, M. Rabinovich, and M. Allman, “Assessing DNS Vulnerability to Record Injection,” [C] in Proceedings of International Conference on Passive and Active Measurement, 2013.

[5] B. Chun, D. Culler, and T. Roscoe, “Planetlab: an overlay testbed for broad-coverage services,” [C] in Proceedings of annual conference of the ACM Special Interest Group on Data Communication, 2003, vol. 33, no. 3, pp. 3–12.

[6] K. Dhandere, H. Kim, and T. J. Pan, “The Application and Effect of Sampling Methods on Collecting Network Traffic Statistics.”

[7] G. He and J. C. Hou, “On sampling self-similar Internet traffic,” [J] Comput. Networks, vol. 50, pp. 2919–2936, 2006.

[8] F. Raspall, “Efficient packet sampling for accurate traffic measurements,” Comput. Networks[J], vol. 56, no. 6, pp. 1667–1684, 2012.

[9] D. Song and P. B. Gibbons, “New Streaming Algorithms for Fast Detection of Superspreaders,” [C] in Proceedings of Network and Distributed System Security Symposium, 2005.

[10] Q. Zhao, J. Xu, and A. Kumar, “Detection of super sources and destinations in high-speed networks: Algorithms, analysis and evaluation,”[J] IEEE J. Sel. Areas Commun., vol. 24, no. 10, pp. 1840–1852, 2006.

[11] T. Li, S. Chen, W. Luo, and M. Zhang, “Scan detection in high-speed networks based on optimal dynamic bit sharing,”[C] Proc. Int. Conf. Comput. Commun., pp. 3200–3208, 2011.

[12] M. K. Yoon, T. Li, S. Chen, and J. K. Peir, “Fit a compact spread estimator in small high-speed memory,”[J] IEEE/ACM Trans. Netw., vol. 19, pp. 1253–1264, 2011.

[13] C. Henke, C. Schmoll, and T. Zseby, “Empirical Evaluation of Hash Functions for PacketID Generation in Sampled Multipoint State of Art,” [C] in Proceedings of International Conference on Passive and Active Network Measurement, 2009, pp. 197–206.

[14] 《第 35 次中国互联网络发展状况统计报告》，国家互联网应急响应中心,2015/02

[15] 基于分布式平台的 DNS 信息探测系统设计与实现，哈尔滨工业大学，孙瑞 .2013. DESIGN AND IMPLEMENTION FOR DNS INFORMATION DETECTION SYSTEM BASED ON DISTRIBUTED PLATFORM. Sun Rui,2013