doi:10.11959/j.issn.1000-436x.2016274

基于 Cookie 的网盘资源在线溯源方法

林海伦[1]，李焱[2]，王伟平[1]，岳银亮[1]，林政[1]
(1. 中国科学院信息工程研究所，北京 100093；2. 国家计算机网络应急技术处理协调中心，北京 100029)

摘 要：网盘作为一种基于互联网的信息传播载体，其所分享的敏感资源已经在网络流量中占有越来越多的比例，因此，获取网盘资源的分享链接对于网络安全有着重要的意义。提出了一种高效可扩展的基于 Cookie 的网盘资源溯源方法—CookieTracing。该方法通过在海量的 HTTP 会话中建立 Cookie 与 HTTP 会话的索引表来实现网盘资源和下载网盘资源的跳转链的关联，同时通过累计散列算法加快溯源结果的验证。实验结果表明，所提方法具有较好的性能和可扩展性。

关键词：网盘资源；分享链接；URL 跳转链；Cookie；HTTP 会话
中图分类号：TP319 **文献标识码**：A

Cookie based online tracing method for cyberlockers resource

LIN Hai-lun[1], LI Yan[2], WANG Wei-ping[1], YUE Yin-liang[1], LIN Zheng[1]
(1. Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China;
2. National Computer Network Emergency Response and Coordination Center, Beijing 100029, China)

Abstract: Cyberlockers have recently become an Internet-based agent of information dissemination. In light of the non-negligible fraction accounted by the traffic flows originating from cyberlocks, it is necessary to trace them for network security. An efficient and scalable cookie based online cyberlockers resource tracing method was proposed, called CookieTracing. It can achieve an efficient association between cyberlockers resource and its download redirect chain by construction of index table between cookie and HTTP sessions in massive HTTP sessions. Meanwhile, through cumulative hash algorithm, it can speed up the validation of tracing results. Experimental results show that this method performs good efficiency and scalability.

Key words: cyberlockers resource, shared links, URL chain, Cookie, HTTP session

1 引言

随着互联网技术的飞速发展，网络作为一个开放式的平台，为用户提供了众多可以分享和下载资源的服务，如 P2P [注1]、BitTorrent [注2]以及目前比较流行的网盘。由于网盘操作简单，用户无需安装软件就可以一键分享、下载资源；而且与 BitTorrent 等传统资源分享模式相比，下载速度快。网盘具备的这些特点导致P2P和BitTorrent使用量急剧下降[1~3]。目前，统计已有很多研究对网盘的使用情况，Maier 等[1]对网盘的网络流量进行了统计分析，发现网盘流量占普通网络流量总数的 17%。Gehlen 等[2]对网盘的点击量进行了统计分析，发现网盘是排名前 10 的网络应用，并且占据 5%的点击量。Allot 等[3]则对网盘在移动终端上的网络流量进行了统计分析，发现网盘流量占据移动终端网络流量总数的 19%。通

注1：https://en.wikipedia.org/wiki/Peer-to-peer。
注2：https://en.wikipedia.org/wiki/BitTorrent。

收稿日期：2015-10-25；**修回日期**：2016-06-30
基金项目：“核高基”国家科技重大专项基金资助项目（No.2013ZX01039-002-001-001）；国家自然科学基金资助项目（No.61303056, No.61402464, No.61402473, No.61502478, No.61602467）
Foundation Items: The National Science and Technology Major Project of Hegaoji (No.2013ZX01039-002-001-001), The National Natural Science Foundation of China(No.61303056, No.61402464, No.61402473, No.61502478, No.61602467)

过上述分析可以看出，网盘已成为重要的网络资源分享和下载的方式。

当用户利用网盘分享资源时，网盘会给该资源生成唯一与之对应的 URL 标识，用户将该链接分享至网络社交平台，其他用户即可点击该链接下载分享资源，这些用户点击分享链接后会弹出一个带有下载按钮的页面（本文将其定义为入口页面），该页面的 URL 即为资源的分享链接，页面会描述该下载资源的属性信息，如资源发布者、资源发布时间、资源下载次数等。

当用户单击入口页面中的下载按钮下载该资源时，用户使用的浏览器会自动向服务器发出一系列 HTTP 请求(本文将其定义为资源下载的 URL 跳转链)，直至成功建立下载资源的 HTTP 会话。如何从海量的网络流量中获取网盘下载资源所对应的入口页面对于网络审查[4]、网络取证[5]、网络流量监控[6]等具有重要意义，本文将这一过程定义为网盘资源溯源。

众所周知，Referer 是 HTTP 表头的一个字段，用来指定当前请求资源的来源地址。然而，在真实流量统计中，大约只有 17%的 HTTP 会话存在 Referer 字段。因此，只依赖 Referer 字段无法获取绝大部分下载资源的入口页面。同时，网络地址转换(NAT, network address translation)[7]、多路多播技术[8]和 HTTP 代理[9]等技术的使用也导致公网路由节点捕获的 HTTP 会话的 IP 地址无法作为精确追溯其 URL 跳转链的依据。而 Cookie 中包含计算机和浏览器的信息，可以用来辨别用户身份、进行 session 跟踪。

为此，本文提出了一种高效可扩展的基于 Cookie 的网盘资源在线溯源方法——CookieTracing，该方法的创新之处有以下几点。

1) 提出了一种基于 Cookie 的网盘资源溯源方法，基于散列技术，通过建立 location 字段与 HTTP 会话以及 Cookie 与 HTTP 会话的散列表实现网盘资源溯源。

2) 通过缓存 HTTP 会话的 Cookie、URL 和 location 字段，采用累计散列算法加快溯源结果验证，从而适应在线流量的溯源。

2 相关工作

目前，针对网盘资源溯源，与之相关的研究工作主要有 2 类。

一类是针对网页木马、恶意网页识别[10]提出的针对 URL 跳转链的入口 URL 识别方法。由于网页木马以及恶意网页为了躲避检测，通常都会经过多次 URL 重定向将用户浏览器最终引向恶意代码网页[10]。这种 URL 多次跳转给网页木马和恶意网页的识别带来了很大的挑战。

为此，已有很多工作围绕网页木马、恶意网页等入口 URL 的识别展开研究，如 Lee 和 Jenefa 等[10, 11]针对 Twitter 上存在的恶意 URL 识别提出了 WarningBird 方法，该方法通过收集同一恶意网页的多条 URL 跳转链获取入口 URL，通过入口 URL 的特征识别恶意网页。Zhang 等[12]针对网页木马识别提出了 Arrow 方法，该方法首先通过蜜罐技术收集同一恶意软件的不同 URL 跳转链；其次，对比 URL 跳转链各个节点的 IP 和域名获取恶意软件的入口 URL；最后，针对该入口 URL 提取 URL 模式，根据 URL 模式识别网页木马。

通过分析可以发现，WarningBird 和 Arrow 方法[10~12]都是通过收集恶意网页代码的 URL 跳转链，离线学习入口 URL 的特征，根据这些特征实现恶意网页代码及其入口 URL（恶意网页、挂马网页）的识别。这种方法虽然可用于网盘资源的溯源，但是还存在一些不足。目前，众多的网盘对应的分享资源的 URL 跳转链特征并不一致，而且通过调研发现即使对于同一网盘的分享资源的不同下载，其特征也会变化，所以现有的方法难以直接适用于网盘的分享资源的溯源。

另一类是针对 NAT 和 HTTP 代理导致骨干网关上数据分组的 IP 地址无法标识用户而提出的在 NAT 主机进行识别的技术[13,14]。例如，Goldberg 等[13]通过分析 HTML 网页内容，以及 HTTP 会话中的 user-agent 字段，实现了对不同用户发出的一系列 HTTP 请求的关联。Maier 等[14]通过对用户浏览器的版本和配置等信息产生“浏览器指纹”的方法，识别出不同用户浏览器所发出的的 HTTP 会话。Neasbitt 等[15]提出了一种基于网络流量跟踪的用户—浏览器交互重构方法。上述这些方法虽然能够识别不同用户的 HTTP 请求，但是存在以下缺陷：骨干网络大部分的 HTTP 会话中只包含 user-agent，而没有其他的配置信息，如字体、插件、时间等，这将导致方法失效。不仅如此，这种方法需要缓存网页内容，针对骨干网络的巨大流量，这会极大地加剧空间开销。

通过对相关工作的分析可以看出，虽然目前已经出现了一些针对资源溯源的方法，但是这些方法无法有效处理网盘资源的溯源。特别地，随着网络大数据的爆炸性增长和网盘的流行，需要研究有效的网盘资源溯源方法，提高资源溯源的准确性。

3　CookieTracing 方法的原理

本节将详细介绍 CookieTracing 方法的原理。为此，首先给出 URL 跳转链和 CookieTracing 方法的形式化定义，然后介绍 CookieTracing 识别网盘分享源下载入口页面的处理流程。

3.1　问题定义

定义 1　URL 跳转链。给定一个网盘资源的分享链接，用户通过浏览器访问该链接发送下载资源的 HTTP 会话请求，到建立下载该资源的 HTTP 会话完成资源下载为止，这期间发出的一系列 HTTP 请求对应的所有 URL，称为该资源下载对应的 URL 跳转链。

由于时间、地点、位置的不同，对于同一个网盘资源的分享链接，每一次下载该资源对应的 URL 跳转链中的各个 URL 节点可能都不相同。CookieTracing 方法的目标就是基于不同用户下载该网盘资源产生的 URL 跳转链，查找 URL 跳转链中的公共节点，从而实现网盘资源入口页面的识别。下面通过一个例子来简单说明基于 URL 跳转链识别网盘资源入口页面的思想。

以用户 A、B、C 为例，他们利用同一网盘资源分享链接下载资源产生的 URL 跳转链如图 1 所示。

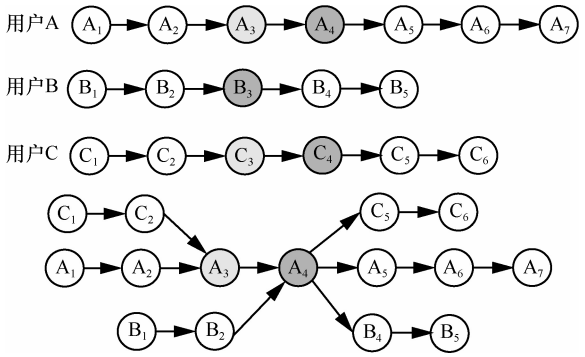


图 1　网盘下载资源入口页面查找示例

在图 1 中，A_4、B_3、C_4 分别表示用户 A、B、C 下载资源时的入口页面，如果能获取 A、B、C 各自资源下载的 URL 跳转链，提取出这 3 条 URL 跳转链的公共节点，就可以找到该资源的入口点 A_4（即 B_3、C_4）。

通过以上分析可以看出，网盘资源溯源需要经过以下几个步骤：首先，从网关流量中识别下载资源并计算资源的标识 ID；然后，获取下载资源的 URL 跳转链；最后，合并具有相同资源标识 ID 的不同 URL 跳转链，获取唯一的公共 URL 节点，该节点即为该下载资源对应的入口页面。因此，本文提出的网盘溯源方法——CookieTracing，就是基于不同用户通过浏览器访问资源产生的 Cookie 信息，采用上述处理方式对网盘资源进行溯源。

3.2　CookieTracing 方法流程

在本节，将详细介绍 CookieTracing 方法进行网盘资源溯源的处理流程。

3.2.1　下载资源的标识 ID 计算

通过分析发现，用于网盘资源传输的 HTTP 会话具有以下几个特点：1) 下载资源 HTTP 会话的 content type 的取值有几种，分别为 video/mp4、application/stream 等；2) 在真实流量统计中显示，93%的下载资源 HTTP 会话的 content length 都在 50 MB 以上。因此，可根据上述特点识别出所有包含网盘下载资源在内的下载资源。

由于下载资源在网络上是按分组传输的，在大流量环境中传统缓存整个下载资源数据计算资源 MD5 的方法无法适用于在线流量的计算，原因在于：一方面，这种方式极大地消耗了内存资源，另一方面，也增加了分享链接的获取时间。为此，CookieTracing 采用了累计散列的方法计算下载资源的标识 ID，该方法对于按分组到达的数据，对每个字节累计进行散列，将下载数据映射成一个 64 bit 的散列值，从而获得下载资源的标识 ID。真实流量中，下载资源的部分数据即可以对资源进行区分，因此，CookieTracing 方法只对下载资源的前 20%~30%数据做累计散列，用来实现下载资源的标识 ID 的计算。

3.2.2　资源的 URL 跳转链提取

对于网盘分享资源下载生成的 URL 跳转链中，每个节点对应的 HTTP 会话的 Cookie 信息可能存在多个键—值（key-value）相同的项，本文将其定义为 token。其中，某些 token 是网盘服务器用来追踪用户，标识用户的访问记录。为此，本文定义了 token 的区分度 *dif*，计算公式如下

$$dif = \frac{N_{\text{token-cookie}}}{N_{\text{cookie}}}$$

其中，$N_{\text{token-cookie}}$ 为包含该 token 的 HTTP 会话数；N_{cookie} 为总的 HTTP 会话数。

为了提高 URL 跳转链计算的准确性，本文定义 HTTP 会话的关联度 sim_{token}，计算公式如下

$$sim_{\text{token}} = N_{\text{simtoken}}$$

其中，N_{simtoken} 为 2 个 HTTP 会话的 Cookie 区分度高的 token 的个数。如果 2 个 HTTP 会话的关联度 sim_{token} 大于阈值 sim_0，则认为这 2 个 HTTP 会话属于同一条 URL 跳转链。因此，只要获取与下载资源 HTTP 会话有着高关联度的一系列 HTTP 话单就可获取 URL 跳转链。

根据 HTTP 重定向原理可知，下载资源 HTTP 会话的 URL 与重定向 HTTP 会话的 location 相同，而重定向的 HTTP 会话存在 Cookie 信息。因此，在计算网盘资源下载的 URL 跳转链时，首先通过下载资源 HTTP 会话的 URL 获取重定向 HTTP 会话；然后，通过重定向 HTTP 会话即可获取完整的 URL 跳转链。

3.2.3 资源的入口页面计算

根据网盘资源的标识 ID，对网盘资源下载的 URL 跳转链进行分组，将具有相同标识 ID 对应的资源下载的 URL 跳转链进行合并，对合并之后的 URL 跳转链上的节点进行遍历，查找 URL 跳转链上的割点，若该割点是合并的 URL 跳转链上的唯一的公共 URL 节点，那么该节点即为该网盘资源的入口。

基于上述 CookieTracing 方法的原理和处理流程，下面将详细介绍 CookieTracing 方法的实现。

4 CookieTracing 方法实现

在本节，首先介绍 CookieTracing 方法的整体框架，然后介绍各个模块的具体实现。

4.1 基本框架

CookieTracing 方法主要包含 4 个部分：HTTP 会话收集、HTTP 会话索引、URL 跳转链计算和资源入口计算，在进行网盘资源溯源时，该方法整体的处理框架如图 2 所示。

1) HTTP 会话收集模块负责对输入的网络流量进行解析，获取所需的 HTTP 会话，并缓存 HTTP 会话的头部信息，以便降低存储空间开销。

2) HTTP 会话索引模块负责解析 HTTP 会话，对海量的 HTTP 会话建立 Cookie 字段与 HTTP 会话的关联。

3) 资源 URL 跳转链计算模块，负责根据下载资源 HTTP 会话获取重定向 HTTP 会话，并根据重定向 HTTP 会话的 Cookie 信息提取资源下载的 URL 跳转链。

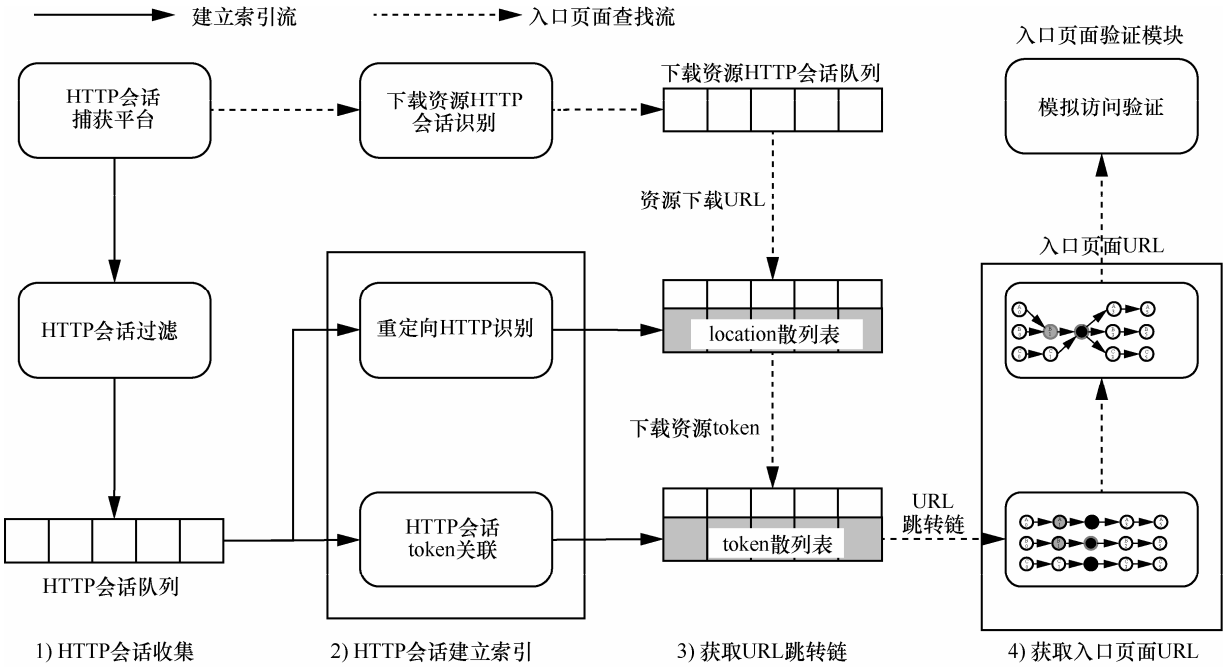


图 2 CookieTracing 实现架构

4) 资源入口页面计算模块负责合并同一下载资源的多个 URL 跳转链，获取合并的 URL 跳转的唯一公共节点，并通过比较分享链接下载资源的标识 ID 与 Load Runner[16]模拟访问收集的资源标识 ID，验证所找到的资源入口页面的正确性。

4.2 模块实现

本节将详细介绍 CookieTracing 方法中每个模块的具体实现细节。

4.2.1 HTTP 会话收集

该模块通过网络流量处理平台解析 HTTP 会话信息。首先，过滤出 2 类需要的 HTTP 会话。

1) 如果 HTTP 会话的 content-type 字段的值为 text/html，且存在 Cookie 字段，则将这类 HTTP 会话信息的三元组：（URL，Cookie，TCP 连接建立时间戳）缓存于 HTTP 会话队列。

2) 如果 HTTP 会话的 content-type 字段的值为 video/x-ms-wmv、video/mp4 等音视频 MIME 类型，且该 HTTP 会话的 content-length 大于某阈值，则该 HTTP 会话即为下载资源的 HTTP 会话。将这类 HTTP 会话的四元组：（URL，Cookie，TCP 连接建立时间戳，下载资源标识 ID）缓存于资源下载 HTTP 会话队列。

其次，计算下载资源 HTTP 会话的下载资源标识 ID，本文采用了累计散列算法，计算一个 64 bit 的散列值作为下载资源的标识 ID，具体的计算方法如算法 1 所示。

算法 1　资源标识 ID 计算

输入　*resourceSize*, *key*, *totalAccumulationLen*

输出　*resourceID*

1) Set *accumulationLen*=0;
2) Initialize *hashL*, *hashH* with a random 64 bit constant;
3) for *i*=1 to *resourceSize* do
4) $hashL=hashL\text{^}key[i]\times BASE_1$;
5) $hashH=hashH\text{^}key[i]\times BASE_2$;
6) *accumulationLen*++;
7) *resourceID*=*hashH*<<32 | *hashL*;
8) if *accumulationLen*=*totalAccumulationLen* then
9) break;
10) end if
11) end for
12) return *resourceID*

从算法 1 中可以看出，资源标识 ID 的计算的时间复杂度与下载资源的大小有关，算法的复杂度为 $O(N)$。

4.2.2 HTTP 会话索引创建

该模块对 HTTP 会话队列中的 HTTP 会话建立索引，规则如下。

1) 如果 HTTP 会话中存在 location 字段，则以 location 字段指定的 URL 作为 key，HTTP 会话作为 value，存入 location 索引表。为了降低空间开销，该索引对存储的 HTTP 会话只做一定时间缓存（本文选取的时间间隔为 5 min）。记该索引表为 location-HTTP 索引表，结构如图 3 所示。

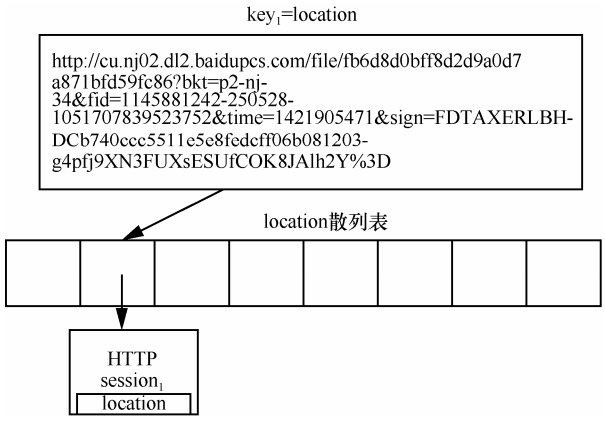


图 3　location-HTTP 会话索引表结构

2) 若 HTTP 会话包含 Cookie，则将 Cookie 分割为 token，以 token 作为 key，包含此 token 的 HTTP 会话作为 value，缓存于 token 索引表，记为 token-HTTP 索引表。其中，每个 token 关联的 HTTP 会话链表按照数据分组的捕获时间进行排序。token-HTTP 索引表的结构如图 4 所示。

为了降低算法的时间开销和空间开销，在建立 token 索引表时会去除区分度 *dif* 不高的 token，如去除存在于大多数 HTTP 会话的 token。

空间开销分析：考虑到互联网访问服务通常是由 IIS 或 Apache 服务器提供的，IIS 或 Apache 默认的 HTTP 会话的大小为 1 MB，如前所述本文选取缓存 5 min 时间间隔内的 HTTP 会话，通过对实际的骨干网络某个节点的流量分析发现，流量中每秒包含约 10 个网盘资源访问 HTTP 会话。因此，5 min 内可能的网盘资源访问 HTTP 会话数量约为 3 000 个，所需的空间开销共计约为 3 GB。

对于 location-HTTP 索引表，其所需的空间开销主要由 URL 和 HTTP 会话的编号 ID 所需的空间

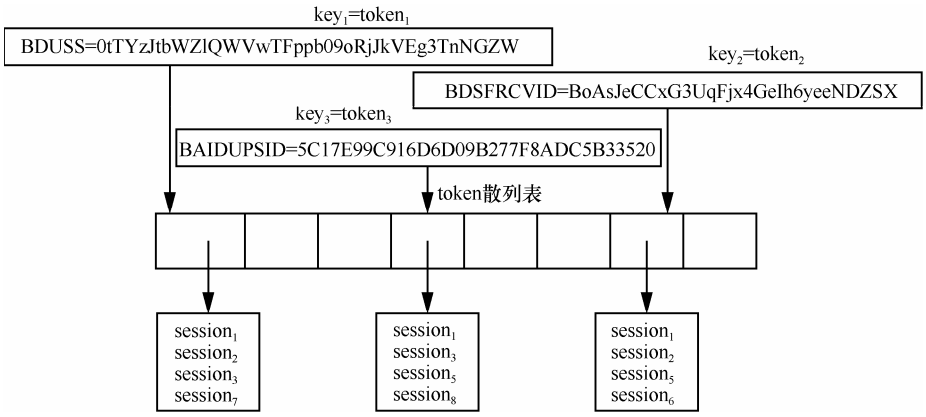


图 4 token-HTTP 索引表结构

开销组成：URL 的平均长度不超过 1 024 byte；HTTP 会话的编号 ID 的长度为 8 byte。因此，5 byte 时间间隔内索引表所需的空间开销约为 3 MB。

对于 token-HTTP 索引表，其所需的空间开销也是主要由 token 和 HTTP 会话的编号 ID 所需的空间开销组成：每一个 token 不超过 8 byte；HTTP 会话的编号 ID 的长度为 8 byte。一个 HTTP 会话的 Cookie 中的 token（属性）的平均选取数量不超过 5 个，因此，5 min 时间间隔内索引表所需的空间开销约为 0.24 MB。

通过分析可以看出，在 CookieTracing 方法中，HTTP 会话索引总的空间开销不超过 4 GB。

4.2.3 URL 跳转链计算

该模块的处理过程包括以下几步。

1) 将从下载资源的 HTTP 会话队列中出队的 HTTP 会话的 URL 作为 key，查找 location-HTTP 索引表，获取重定向 HTTP 会话。

2) 将重定向 HTTP 会话的 Cookie 分割成 token，以 token 为 key，查找 token-HTTP 索引表，获取所有包含这些 token 的 HTTP 会话，本文将这些 HTTP 会话链定义为疑似 HTTP 会话链。

3) 遍历疑似 HTTP 会话链，统计 HTTP 会话在疑似 HTTP 会话链中出现的频率。如果其频率大于指定关联度阈值，即认为其属于下载资源的 URL 跳转链。

下面通过一个例子来说明，下载资源 URL 跳转链的计算。给定一个下载资源，其对应的重定向 HTTP 会话包含的 Cookie 可分为 4 个 token，分别记为 $token_1$、$token_2$、$token_3$ 和 $token_4$，以这些 token 为 key，查找 cookie-HTTP 索引表，获取 4 个 token 分别对应的 HTTP 会话链，如图 5 所示。

在图 5 所示的例子中，规定每一个 HTTP 会话若其出现在 HTTP 会话链中的频率大于 1，则该 HTTP 会话属于 URL 跳转链。因此，比较 $token_1$、$token_2$、$token_3$ 和 $token_4$ 关联的 4 条 HTTP 链，发现编号为 1、2、4、8 的 HTTP 会话在 4 条 HTTP 会话链中出现的频率都大于 1，所以它们属于下载资源的 URL 跳转链。根据 token-HTTP 索引表中，HTTP 会话链按照数据分组获取的时间排序，因此，该下载资源的 URL 跳转链即为 1→2→4→8。

4.2.4 资源入口页面计算

与从疑似 HTTP 会话中获取 URL 跳转链的方法类似，CookieTracing 方法基于统计的方式，从下载资源的 URL 跳转链中获取资源的入口页面，主要包含以下几个步骤。

1) 将具有相同下载资源标识 ID 的 URL 跳转链进行合并。

2) 遍历合并的 URL 跳转链，寻找割点，若该割点在该下载资源对应的所有的 URL 跳转链中出现的频率最高，则该节点即为该下载资源真正的入口页面。

3) 通过 Load Runner[16]模拟用户访问网盘资源的分享链接，重新下载该资源，然后通过累计散列计算该资源的标识 ID 值并与 CookieTracing 计算出的标识 ID 做对比，如果二者相同，则该网盘资源的入口页面被确定。

5 实验与分析

为了验证本文提出的基于 Cookie 的网盘资溯源（CookieTracing）方法的性能，本节将对 CookieTracing 的有效性进行实验分析，首先测试

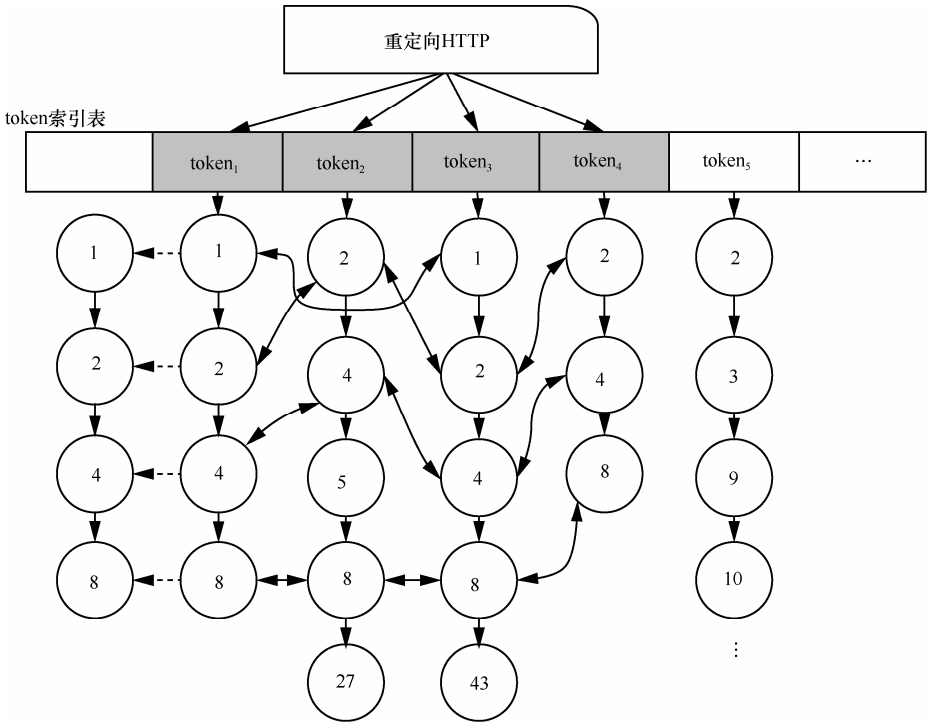


图 5　URL 跳转链获取过程

CookieTracing 方法进行网盘资源溯源的查准率和查全率；然后测试 CookieTracing 方法的运行效率。

5.1　实验设置

1) 评价指标

在实验中，针对有效性测试，使用查准率和查全率进行评价。其中，查准率指查找到的正确资源入口点占查找到的网盘资源入口点的比例；查全率指查找到的正确资源入口点占所有网盘资源入口点的比例。在运行效率测试中，使用获取时间进行评价（指获取入口点的时间）。

2) 基准方法

为了验证 CookieTracing 方法对网盘资源溯源的性能，采用最新的方法 WarningBird[10, 11]作为基准方法（详见第 2 节）。

3) HTTP 会话索引存储

在实验中，本文采用基于内存的 key-value 数据库 Redis [注3]存储 HTTP 会话索引。

在实验中，首先，通过百度网盘搜索引擎获取视频资源的分享链接。然后，利用 Load Runner 模拟用户请求这些分享资源链接，收集各自对应的 URL 跳转链。最后，在网关上统计随着下载资源增多，CookieTracing 方法和 WarningBird 方法进行网盘资源溯源的查准率、查全率，以及它们的运行时间。下面分别介绍 CookieTracing 方法对应的有效性、运行效率实验结果。

5.2　实验结果

1) 有效性测试

CookieTracing 方法与 WarningBird 方法查准率的实验结果如图 6 所示。

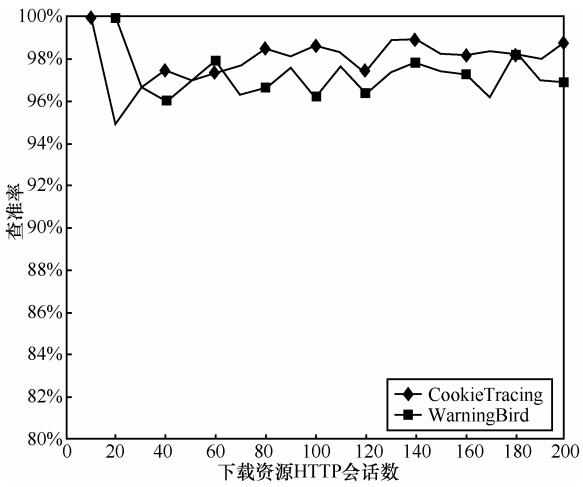


图 6　查准率实验结果

从图 6 中可以看出，CookieTracing 方法和 WarningBird 方法的查准率基本一致，平均查准率分别是 98.67%、97.76%，导致这一现象的原因在于：

注3：http://redis.io/。

这 2 种方法在网盘资源的入口点查找时采用的算法基本一致，都是通过合并资源的 URL 跳转链，计算跳转链中的公共节点获得资源的入口点。值得注意的是，在网关上由于流量捕分组采集不稳定因素，导致网盘资源溯源的查准率在一定范围内呈现波动现象，但整体上呈稳定趋势。

CookieTracing 方法与 WarningBird 方法查全率的实验结果如图 7 所示。从图 7 可以看出，与 WarningBird 方法相比，在对网盘资源进行溯源时，CookieTracing 的查全率远远高于 WarningBird 方法。其中，CookieTracing 方法的平均查全率为 98.86%，而 WarningBird 方法的平均查全率为 16.67%。主要原因在于：WarningBird 方法采用基于 HTTP Referer 字段的方法，在真实流量统计中，HTTP 会话存在 Referer 字段的比例很少，只依赖 Referer 字段难以获取绝大部分下载资源的入口页面。而 Cookie 在资源请求访问中是普遍存在的，基于 Cookie 进行网盘资源溯源将是一种非常有力的方式。

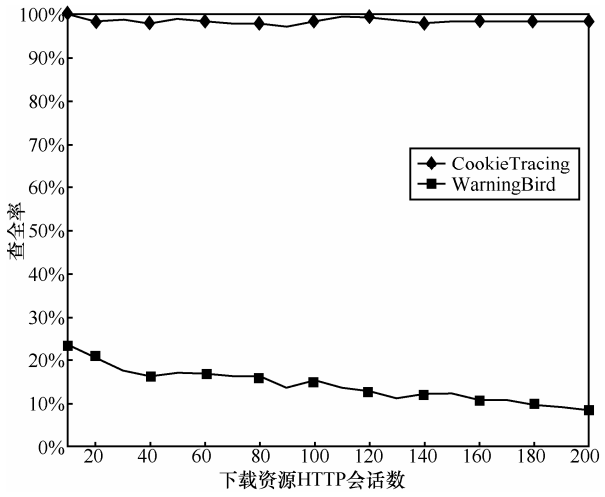


图 7 查全率实验结果

由此可见，虽然 WarningBird 方法具有和 CookieTracing 方法几乎相当的查准率，但是在查全率方面，WarningBird 方法仅是 CookieTracing 方法的$\frac{1}{5}$，这进一步验证了基于 Cookie 方式的 CookieTracing 方法对网盘资源溯源的有效性。

2) 运行效率测试

本节将评估 CookieTracing 方法与基准方法 WarningBird 在网盘资源入口识别上的运行效率，实验结果如图 8 所示。

从图 8 中可以看出，随下载资源的增加，CookieTracing 方法资源入口的查找时间明显快于 WarningBird 方法，并且随着下载资源 HTTP 会话的增加，CookieTracing 方法的查找时间基本保持线性增长，而 WarningBird 方法呈指数增长，这说明在实时性方面 CookieTracing 方法明显优于 WarningBird 方法。主要原因在于 CookieTracing 方法采用累计散列算法计算资源 ID 标识，能够加快资源 ID 的计算。

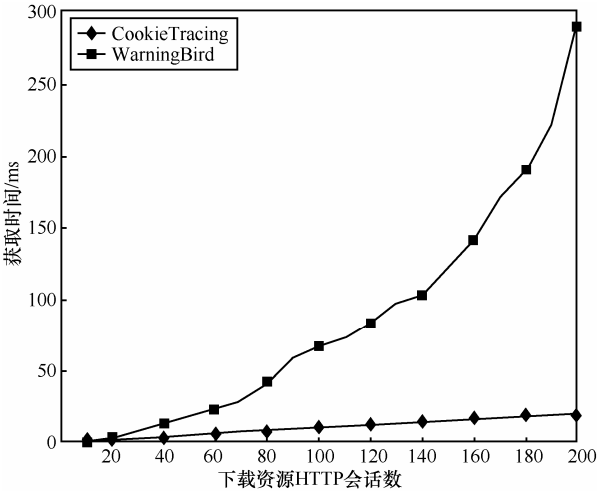


图 8 运行效率实验结果

基于以上实验分析可以看出，与基准方法相比，CookieTracing 方法在进行网盘资源溯源时，不仅可以获得更高的准确率，而且在实时性方面也能获得更好的效果，这些都表明 CookieTracing 方法的有效性，这也说明在网盘资源溯源中，采用 Cookie 是一项非常有用的技术。

6 结束语

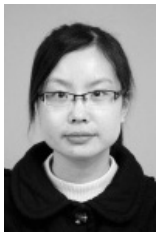
如何从骨干网络节点上的海量流量中识别出网盘资源下载的 HTTP 会话的入口页面对于网络审查、网络取证、网络审计等具有重要意义。为此，本文提出一种基于 Cookie 的网盘资源在线溯源方法——CookieTracing。CookieTracing 方法首先获取下载资源的 URL 跳转链，然后通过对比同一下载资源对应的不同 URL 跳转链获取唯一公共 URL 节点，认为该 URL 即为下载资源对应的入口页面。最后通过 Load Runner 模拟用户访问该 URL，验证溯源的正确性。实验结果表明 CookieTracing 方法具有很好的性能。

参考文献：

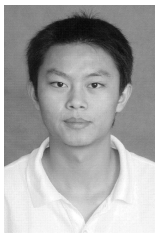
[1] MAIER G, FELDMANN A, PAXSON V, et al. On dominant charac-

teristics of residential broadband Internet traffic[C]//9th ACM SIGCOMM Conference on Internet Measurement. ACM, 2009:90-102.
[2] GEHLEN V, FINAMORE A, MELLIA M, et al. Uncovering the big players of the Web[M]. Springer Berlin Heidelberg, 2012.
[3] MOBILE TRENDS A. Global mobile broadband traffic report[R/OL]. Allot Communications, Technical Report, http://www. allot. com/MobileTrends Report, 2010.
[4] BERGHEL H. The discipline of Internet forensics[J]. Communications of the ACM, 2003, 46(8): 15-20.
[5] WATTS S, NEWBY J M, MEWTON L, et al. A clinical audit of changes in suicide ideas with internet treatment for depression[J]. BMJ open, 2012, 2(5): e001558.
[6] PANAH A, PANAH A, PANAH O, et al. Challenges of security issues in cloud computing layers[J]. Rep Opin, 2012, 4(10): 25-29.
[7] GOKCEN Y, FOROUSHANI V A, HEYWOOD A. Can we identify NAT behavior by analyzing traffic flows[C]//IEEE Security and Privacy Workshops (SPW). 2014: 132-139.
[8] LIU T T, YANG W, XU C L, et al. A SNR-based multi-channel multicast scheme for popular video in wireless networks[J]. Journal of Networks, 2013, 8(3): 628-635.
[9] HAYTON S J, JONES D R, LOBO A R, et al. Using entity tags (etags) in a hierarchical HTTP proxy cache to reduce network traffic: U.S. Patent Application 13/360,891[P]. 2012-1-30.
[10] LEE S, KIM J. Warningbird: a near real-time detection system for suspicious URLs in twitter stream[J]. IEEE Transactions on Dependable and Secure Computing, 2013 (3): 183-195.
[11] JENEFA A, RAVI R. Classifier: a real-time detection system for suspicious URLs in Twitter stream[J]. International Journal, 2014, 2(2).
[12] ZHANG J, SEIFERT C, STOKES J W, et al. Arrow: generating signatures to detect drive-by downloads[C]//20th International Conference on World Wide Web. ACM, 2011: 187-196.
[13] GOLDBERG J, WESTERLUND M, ZENG T. A network address translator (NAT) traversal mechanism for media controlled by real-time streaming protocol (RTSP)[J/OL].http://tools.ietf.ory/html/draft-ietf-mmusic-rtsp-nat-03.
[14] MAIER G, SCHNEIDER F, FELDMANN A. NAT usage in residential broadband networks[M]. Passive and Active Measurement. Springer Berlin Heidelberg, 2011.
[15] NEASBITT C, PERDISCI R, LI K, et al. Clickminer: towards forensic reconstruction of user-browser interactions from network traces[C]// The 2014 ACM SIGSAC Conference on Computer and Communications Security, 2014: 1244-1255.
[16] JINYUAN C. The application of load runner in software performance test[J]. Computer Development & Applications, 2012, 5: 014.

作者简介：



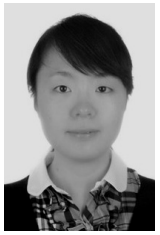
林海伦（1987-），女，山东临沂人，博士，中国科学院信息工程研究所助理研究员，主要研究方向为数据挖掘、知识图谱。



李焱（1984-），男，湖北随州人，国家计算机网络应急技术协调中心工程师，主要研究方向为分布式系统和云计算。

王伟平（1975-），男，吉林舒兰人，博士，中国科学院信息工程研究所研究员、博士生导师，主要研究方向为大数据存储与处理。

岳银亮（1982-），男，河南许昌人，博士，中国科学院信息工程研究所副研究员，主要研究方向为大数据存储与智能化处理。



林政（1984-），女，山东青岛人，博士，中国科学院信息工程研究所助理研究员，主要研究方向为自然语言处理、情感分析。