

# 一种基于多维特征分析的网页代理服务发现方法

陈志鹏<sup>1,2</sup>, 张 鹏<sup>2\*</sup>, 黄彩云<sup>1,2</sup>, 刘庆云<sup>2</sup>, 邢丽超<sup>1,2</sup>

<sup>1</sup> 中国科学院大学 网络空间安全学院, 北京 中国 100049

<sup>2</sup> 中国科学院信息工程研究所 信息内容安全技术国家工程实验室, 北京 中国 100093

**摘要** 网页代理提供了一种快捷的中继服务, 与其它类型的代理服务相比, 如隐匿网络/VPN 服务/Socks 代理等, 用户不需要安装任何软件就免费使用。因此, 网页代理在绕过访问限制、隐藏身份等方面的便利性上有其不可比拟的优势。然而, 网页代理在获取个人隐私信息、推送垃圾广告、隐匿行踪等方面也给人们的网络生活带来严重的安全威胁。所以, 如何快速有效地将它们与大量正常网页区分开来成为网络空间安全面临的一个重要挑战。针对这一问题, 本文提出了一种基于多维特征分析的网页代理发现方法——ProxyMiner。在主动发现方面, 引入了网页代理特有的结构特征和内容特征, 通过机器学习的方法进行预测发现。在被动发现方面, 基于用户访问网页代理特有的访问模式, 通过构建二分图对代理用户进行谱聚类分析, 获取代理用户群体访问的顶级域名, 从而发现网页代理。此方法仅基于客户端 IP 地址和目标 URL, 不需要任何有关 HTTP 头(经常会被恶意修改)或数据包(通常是加密的或不可用的)的信息。实验结果表明, 在相同数据集上, 相比于传统检测方法, ProxyMiner 可以显著提高网页代理检测效果, 降低平均检测时间。

**关键词** 网页代理; 服务发现; 主被动结合; 谱聚类分析

中图法分类号 TP393.8 DOI号 10.19363/J.cnki.cn10-1380/tn.2018.07.04

## A Web Proxy Detection Method based on Multiple Feature Analysis

CHEN Zhipeng<sup>1,2</sup>, ZHANG Peng<sup>2\*</sup>, HUANG Caiyun<sup>1,2</sup>, LIU Qingyun<sup>2</sup>, XING Lichao<sup>1,2</sup>

<sup>1</sup> School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China

<sup>2</sup> State Key Laboratory Of Information Security, Institute of Information Engineering,  
Chinese Academy of Sciences, Beijing 100093, China

**Abstract** Web proxies offer a quick and convenient solution for routing web traffic towards a destination. In contrast to more elaborate relaying systems, such as anonymity networks, VPN services or Socks proxies, users can freely connect to web proxies without installing any special software. Therefore, web proxies are an attractive option for bypassing restrictions and hiding identity. However, it has become a much more serious problem for personal privacy, malicious advertisements and property safety due to its dynamics, and evasiveness. Therefore, how to quickly and effectively detect the web proxies from a large number of web pages is an important challenge. To solve this problem, this paper presents an active and passive web proxy detection method based on multiple feature analysis, named ProxyMiner. On the active side, the DOM features unique to Web proxy are introduced, and the method of machine learning is used for predictive analysis. On the passive side, based on the access model specific to the proxy service user, spectral clustering analysis is performed on the proxy user by constructing a bipartite graph, and the top-level domain names accessed by the proxy user group are obtained to discover the proxy service. This method is based solely on the client IP address and the destination address, and does not require any information about HTTP headers (often maliciously modified) or data packets (usually encrypted or unavailable). The experimental results show that ProxyMiner can significantly improve the detection performance and reduce the average detection time compared to traditional detection methods.

**Key words** web proxy; service discovery; active and passive; spectral clustering analysis

## 1 前言

随着信息及网络技术的发展, 网络空间已成为

继陆海空天之外的第五空间, 是世界各国战略必争高地。网络空间运行体系的组成要素可被分为 4 种类型: 载体、资源、主体和操作。其中, 网络空间载

通讯作者: 张鹏, 博士, 副研究员, pengzhang@iie.ac.cn。

本课题得到国家重点研发计划(No. 2016YFB0801304)的资助。

收稿日期: 2018-03-30; 修改日期: 2018-05-30; 定稿日期: 2018-06-19

体是网络空间的软硬件设施,是提供信息通信的系统层面的集合;网络空间资源是在网络空间中流转的数据内容,包括人类用户及机器用户能够理解、识别和处理的信号状态;网络空间主体是互联网用户,包括传统互联网中的人类用户以及未来物联网中的机器和设备用户;网络空间的操作是对网络资源的创造、存储、改变、使用、传输、展示等活动<sup>[1]</sup>。

代理服务作为网络空间载体的一种形式,在数据缓存、访问加速、负载均衡等方面发挥了不可替代的作用。根据 GWI (global web index) Social 统计代理使用报告<sup>[2]</sup>和 Roberts 等人<sup>[3]</sup>研究指出,代理服务由于其特有的功能被世界上许多国家的人们所采用。另一方面,根据 Sandvine<sup>1</sup> 公司最近公布的流量类型的统计数据显示,无论是从应用类型还是从网络用户的维度,网络流量使用的比重分布有很强的偏向性。在不同地域上这种流量的应用类型分布特性还有所区别,通过有效的代理服务,可以减少重复数据传输导致的带宽消耗,减少资金浪费;还可以改善目标用户的服务体验;对于热点内容,提供就近代理缓存服务,因而经历更少的网络延时,带来更快的响应速度。

但是,代理服务技术也是一把双刃剑,代理服务在给人们生产和生活带来便利的同时,一些恶意分子也充分利用代理服务进行网络犯罪和服务。例如,在订票高峰期,很多第三方抢票软件通过代理服务购票,但是,抢票软件不一定都可靠,一些抢票软件非但没有常规的抢票功能,反而会携带病毒或者存在其它的安全隐患(捆绑销售等等)。还有一些代理服务会改变用户数据包的目标传输地址,并将用户的浏览器请求重定向,然后对流氓网络的路由地址进行解析。其结果就是,原始的网络流量会被重定向至加载了广告和恶意软件的恶意站点。而这些代理服务与广告网站(或一些恶意软件站点)之间存在着利益关系,他们一同合作并创造出了大量的广告流量收益,然后双方就可以对这些收入进行分摊。Giorgos<sup>[4]</sup> 等人研究了 HTTP 代理服务中存在的大量恶意行为,如注入或修改广告、搜集用户敏感信息等。另外,一些欺诈交易也是通过恶意代理服务的方式进行的,给人们的生活带来重大损失。

代理服务导致上述问题的原因主要有两方面:一是代理服务承载协议种类的多样性,代理服务多种多样,如网页代理、HTTP 代理、Socks 代理、VPN 代理等等;二是代理服务形态的丰富性,代理服务的形态多表征,数量及分布动态变化,具有隐匿、动

态、时变的特点,使得服务的真实情况难以刻画,发现难。因此,通过对网络空间代理服务进行测绘,可以全面掌握代理服务的特性及其分布,最大可能地形式化、精确化还原代理服务的使用状况,对企业和国家的风险控制、溯源取证有着重要意义。然而,随着互联网的迅速发展,传统代理服务发现机制面临着一些新的挑战:代理规模不断扩大,流量逐步呈现出复杂化、多样化的趋势;多元化的网络传播途径对识别发现系统的实时性提出了更高的要求;加密服务及相关隐匿手段(如代理服务的流量隐匿成非代理流量)的使用,限制了传统依赖内容检测技术的应用范围。Staniford 和 Heberlein<sup>[4]</sup>首次提出了代理服务发现的概念以及基于网络包内容的黑名单或正则表达式发现方法。但是,基于黑名单的方法仅限于检测已知的代理服务,缺少很好的扩展性,并且大多数构造黑名单或正则表达式的方法是人工的方式;基于正则表达式的方法虽然可扩展性好,但是精确度相对不高,而且生成正则表达式的过程也是低效率的。此外,网页代理不用安装客户端或进行其他复杂的设置,并且大多是免费的。这意味着用户可以免费享受代理服务提供的所有优惠,而不必承担任何费用。正如图 1 所示,用户只要输入他们想要访问的 URL,就可以无限制地浏览任何网络资源。因此,如何快速有效地将它们与大量正常网页区分开来成为网络空间安全面临的一个重要挑战。

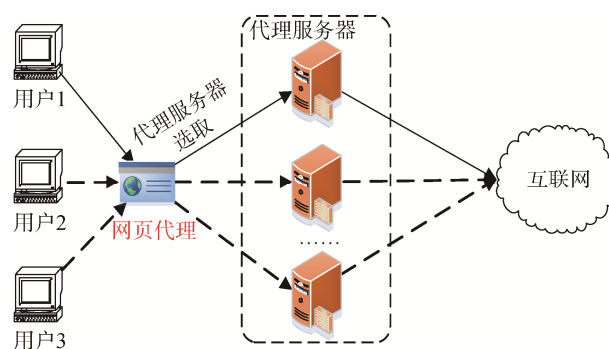


图 1 网页代理使用示例

Figure 1 User Case of the Web Proxy

为此,本文提出了一种多维特征分析的网页代理服务发现方法——ProxyMiner。在主动发现方面,首先通过编写爬虫主动获取代理及非代理的网页数据集,然后抽取网页的 URL 特征、内容特征、DOM 特征(Document Object Model)作为网页代理特征,最后通过构建机器学习模型进行训练,进而识别网页代理。在被动发现方面,通过识别和分析相似访问模

<sup>1</sup> Sandvine (2013). Global Internet Phenomena Report: 2H 2013. <https://www.sandvine.com/trends/global-internet-phenomena/>

式的用户群体来检测网页代理服务。利用代理用户群体抽象出网络用户的访问模式，可以使流量分析的成本显著降低。

本文的主要贡献是提出了主被动发现相结合的特征分析方法对网页代理服务进行发现。具体有两点，一是主动发现方面，首次将网页代理特有的局部特征引入模型中，结合黑名单等方法来发现网页代理。二是被动发现方面，基于对隐藏在网页代理服务后面的用户进行聚类分析，探索网络代理用户的行为相似性，并发现代理用户群体固有的特征，通过检查代理用户群体访问的 URL 进行网页代理服务发现。

本文的后续安排如下：本文第二章主要介绍了网页代理服务发现的相关工作；第三章形式化定义了本文的研究问题，详细阐述了 ProxyMiner 方法的架构和工作流程；第四章进行了详细的实验结果分析；第五章对本文的工作进行了总结。

2 相关工作

代理服务形式多种多样，承载的协议也是种类繁多的，我们首先按照网络层次将不同的代理服务协议和相应的网络层次进行了归纳整理，如图 2 所示，如 HTTP 协议、VPN 协议等等。代理服务的实现方式也是多种多样的，表 1 按时间顺序列举了图 2 所示的常用方法的多个代理服务工具。

Staniford 和 Heberlein<sup>[5]</sup>二人首次提出了代理服务发现的概念，并且提出了基于网络包内容的发现方法，毫无疑问，这种方法对加密流量的代理服务就无能为力了。

目前，在学术领域，代理服务发现的方法主要包括四类：基于黑名单技术的识别方法，基于启发式规则(如正则表达式规则)的识别方法，基于机器学习的识别方法，以及基于交互式主机行为的识别方法。

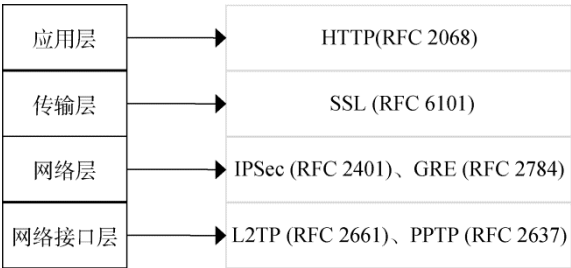


图 2 代理相关协议和方法  
Figure 2 Protocols and Methods on Proxy

基于黑名单的方法仅限于检测已知的代理服务，缺少很好的扩展性。另外，虽然基于黑名单方法在运

行时效率非常高，但是在构造黑名单的过程中，十分麻烦，因为如今大多数构造黑名单的方法还是人工的方式。为了增强可扩展性，同时为了应对代理域名周期性变化的问题，基于正则表达式的方法被提出了，如产生 Snort 规则<sup>[6]</sup>等等。基于正则表达式的方法虽然可扩展性好，但是精确度相对不高，而且生成正则表达式的过程也是低效率的。因此，基于签名的启发式方法被提出了，该方法主要是基于内容的，如基于指纹<sup>[5]</sup>和基于水印<sup>[7]</sup>等等。基于指纹的方法主要是基于流量的内容，如包的特征属性，内容等提炼出签名、正则表达式等方式进行检测；基于水印的方法主要是在流入主机的流量中注入水印特征，若在流出主机的流量中检测出含有水印的包特征，则判定该主机为提供代理服务的主机。这两种基于内容的方法在加密流量中就很难应用了。

基于机器学习的方法不对包内容进行检测，只是对代理服务的特征进行分析，一方面摒弃了侵犯用户隐私的担忧，另一方面也绕过了对加密内容检测内容困难的难题。Vahid<sup>[8]</sup>等人基于存储于服务器中的不同流量日志，用机器学习的方法来识别代理服务。Rueimin<sup>[9]</sup>等人提出了一种基于 RTT(Round-Trip Time)时间的方法来检测是否为代理主机。理论依据是若代理主机提供中继服务，其必然响应用户的请求，并重新建立 TCP 链接，这样基于代理主机的 RTT 总时间必然大于未经过代理主机的 RTT。此外，还有基于其它特征，如包大小、包时间戳、建立链接起止时间、包间延迟等等来发现代理服务<sup>[10-13]</sup>。这些基于时间戳的方法受网络环境影响很大，精度较低，缺少很好的鲁棒性。另外，Deng<sup>[14]</sup>等人通过流量分析 Shadowsocks 流量特征，采用随机森林方法来发现 Shadowsocks 代理服务。基于机器学习的方法，关键是确定相应代理服务的特征，尤其是相对稳定的或者能够动态更新的，否则会造成召回率较低。

当访问网页代理时，可能会出现安装恶意软件或者执行恶意脚本的情况。这时，可以结合虚拟化技术和蜜罐技术对代理网页进行识别。此类方法的工作原理是：使用蜜罐技术，将虚拟主机作为诱饵，访问待检测网页，通过监测访问后的主机动态行为(例如：创建新进程、改变注册表、下载文件、成功访问受限站点等)，判断该网页是否是网页代理。根据使用系统的不同，蜜罐技术可以细分为基于模拟的低交互式蜜罐<sup>[15]</sup>和基于真实系统的高交互式蜜罐<sup>[16]</sup>。诸葛建伟<sup>[17]</sup>等人对此有详细介绍，这里不再赘述。

在工业领域，也有许多商业上的产品提供代理服务发现的功能，他们采用的技术方法主要有：

表 1 常见代理服务工具  
Table 1 Common Proxy Tools

工具	发布时间	HTTP 代理	网页代理	VPN 代理	Socks 代理	分布式主机
Freenet	1999	√				
TriangleBoy	2000	√				
Garden	2000	√				
Anonymizer	2002	√				
DynaWeb	2002	√				
UltraSurf	2002	√				
Circumventor	2003		√			
Coral	2004					√
Hamachi	2004			√		
Psiphon	2004	√	√	√	√	
Firephoenix	2006			√		
GPass	2006			√		
Gtunnel	2007			√		
JAP	2007					√
Shadosocks	2012				√	
总计	15	7	2	5	2	2

表 2 代理服务发现机制常用方法  
Table 2 Proxy Service Detection Methods

方法	黑名单	启发式规则	统计特征学习	行为特征学习	集成学习
数据粒度	数据包	数据包	单一流	复合流	复合流
分类粒度	细	细	粗/细	粗	粗/细
复杂度	低	高	高	高	算法决定
实时性	好	好	一般	算法决定	算法决定
精度	差	较好	较好	较好	较好
扩展性	差	差	较好	一般	好
总结	端口隐匿, 黑名单动态变化, 监测效果有限	提取负载特征困难, 依赖于模式匹配规则, 有隐私保护敏感问题	依赖于流量特征提取, 难度与分类模型的复杂度	依赖用户行为特征建模时间	可以根据业务场景目标具体分析, 难以评估

DPI(Deep Packet Inspector, 深度包检测)、HTTP 头部字段检测、IP 黑名单检测、URL 黑名单检测(通过同已构建的代理库进行匹配, 判断待测 URL 是否在代理库中)、基于启发式规则检测和地理位置检测等方法。其中, IP2Proxy<sup>[18]</sup>检测流量中 HTTP 协议头部字段是否包含 REMOTE\_ADDR、VIA、XFF 字段, 若含有, 则认为是代理服务。但是, XFF 字段在 HTTP 协议中是可选字段, 并且该字段也是可以伪造的, 准确率不高。CIPAFILTER<sup>[19]</sup>通过 URL 黑名单进行发现, MaxMind<sup>[20]</sup>通过 IP 黑名单进行发现, 这两种方法对于代理地址不断变化更新的情况就无能为力了, 扩展性比较差。此外, 还有通过端口扫描进行代理服务发现, 端口扫描是通过对于指定的 IP 地址, 扫描代理常用的端口, 查看该端口的开放状况, 判断其是否为代理服务。为了增强判别的精度, 有人还对待测

IP 所对应的 DNS 服务器的 IP 地址进行反向解析, 验证该 IP 是否注册域名, 从而计算其是否为代理服务的可疑性。当然, 最朴素的方法就是模拟代理访问网站识别法, 通过设置指定的 IP 为代理, 以该 IP 来访问受限的网站, 根据其是否访问成功, 来判断其是否为代理服务。

综合以上学术界和工业界对代理服务的发现方法研究, 表 2 总结了 5 种代理服务发现方法, 并从数据粒度、分类粒度、复杂度、实时性、精度、可扩展性等多方面进行了对比分析。

3 网页代理服务发现方法

3.1 统一描述

网页代理服务通常作为用户(个人主机)通过网络服务提供商获取网络资源的信息中转站。但是网

网页服务协议多种多样, 对此, 我们对网页代理服务进行了统一的形式化定义: 六元组  $(C, P, S, B, A, QoS)$ , 其中  $C$ : 用户、 $P$ : 代理、 $S$ : 网络资源、 $B$ : 行为、 $A$ : 特征属性、 $QoS$ : 评价度。

对于代理行为, 我们统一定义数据包头部空间为  $H: \{0,1\}^L$ , 其中  $L$  为头部长度的。对于网络空间  $\mathcal{N}$ , 我们定义头部空间与端口的向量积:  $\{0,1\}^L \times \{1, \dots, P\}$ , 其中  $\{1, \dots, P\}$  为端口列表。进而分以下两部分定义代理中继服务函数:

第一类, 对于包头要经过封装, 更改包头的转换函数为  $T(h, p): (h, p) \rightarrow \{(h_1, p_1), (h_2, p_2), \dots\}$ , 若要经过多种封装, 可以通过公式(1)形式化定义为:

$$\Psi(h, p) = \begin{cases} T_1(h, p) & \text{if } p \in proxy_1 \\ \vdots \\ T_n(h, p) & \text{if } p \in proxy_n \end{cases} \quad (1)$$

第二类, 对于只是中继包, 不更改包头的网页代理服务的转换函数可以如公式(2)所示:

$$\Gamma(h, p) = \begin{cases} \{(h, p^*)\} & \text{if } p \text{ connected to } p^* \\ \{\} & \text{if } p \text{ is not connected} \end{cases} \quad (2)$$

则对代理每一跳可描述为:  $\Phi(\cdot) = \Psi(\Gamma(\cdot))$ 。对经过  $k$  级网页代理服务可以描述为:

$$\Psi(\Gamma(\dots(\Psi(\Gamma(h, p))) \dots)) \text{ 或 } \Phi^k(h, p)。$$

对于网页代理服务质量评估  $QoS$ , 可以通过一个向量来表示:  $QoS = (Q_1, Q_2, \dots, Q_n)$ 。其中, 向量的每一维度都表示网页代理服务性能中的某一指标, 如代理的跳数、带宽、延迟等等。对于经过多跳的网页代理服务, 可以通过联合操作( $\cup$ )进行计算, 以含有两个指标为例, 如公式(3)所示。

$$\begin{aligned} QoS &= QoS_1 \cup QoS_2 \\ &= (Q_{11} \cup Q_{21}, Q_{12} \cup Q_{22}, \dots, Q_{1n} \cup Q_{2n}) \end{aligned} \quad (3)$$

其中,  $QoS_1 = (Q_{11}, Q_{12}, \dots, Q_{1n})$ ,  $QoS_2 = (Q_{21}, Q_{22}, \dots, Q_{2n})$ 。

不同的指标, 其联合操作的规则是不一样的。例如, 带宽的联合操作为  $\min(QoS_1, QoS_2)$ , 而延迟的联合操作为  $QoS_1 + QoS_2$ 。

综上, 对于任何网页代理服务是可用的, 也就是对于任何一个经过网页代理服务的流  $f(\text{flow})$  是可达的, 除了需要满足一定的  $QoS$  外, 还需要满足没有访问环路和规则  $(f_{in,i}, \Phi_i, f_{out,i})$ , 其中  $f_{out,i-1} = f_{in,i}$  并且  $\Phi_i(f_{in,i}) = f_{out,i}$ , 初始流  $f = f_{in,0}$ 。没有访问环路意味着任何流不能经过相同的网页代理服务两次或多次, 即对于流  $f$  经过网页代理服务  $P_i$ , 不存在  $P_i = P_j$ , 而  $i \neq j$ 。

### 3.2 问题定义

目标: 对于输入的  $N$  个网页 URL  $\{(u_1, y_1), \dots, (u_n, y_n)\}$ , 其中, 对于时间  $t =$

$1, 2, \dots, n, u_n$  表示来自数据集里的网页代理 URL。而  $y_n \in (0,1)$  表示相应的标签,  $y_n = 1$  表明是网页代理的 URL,  $y_n = 0$  表明是非网页代理的 URL。

主动方面利用了 3.1 节提到的六元组  $(C, P, S, B, A, QoS)$  中的  $C$ 、 $P$ 、 $S$ 、 $A$  四种属性, 主要是充分利用  $A$  属性的特征, 通过特征发现, 并没有解析网页里的所有内容(解析全部网页内容会带来很大开销), 而仅是专注于抽取网页代理特有的特征: 在 DOM 结构里有着特定值的 Form 表单。主要分为两部分:

- 1) 特征抽取: 对于每一个数据集网页(或 URL), 构建一个多维特征向量作为机器学习判别的一个输入;
- 2) 机器学习: 基于特征向量集, 通过学习训练得到一个预测判别模型。

被动方面通过利用了 3.1 节提到的六元组  $(C, P, S, B, A, QoS)$  中的  $C$ 、 $P$ 、 $S$  三种属性, 通过用户线索发现网页代理服务使用访问关系, 并计算用户相似度聚类相似用户群体, 重点检测每组用户群的公共 Top  $N$  的 URL, 这就限制了 URL 的检查范围, 避免了对访问流中所有未知 URL 的详细检查。同时, 利用已有用户的代理访问关系建立用户访问反馈机制, 从而对潜在的网页代理服务进行服务发现。

简而言之, 我们在客户端和目的地之间建立了双向图, 如图 3 所示, 通过观察客户端和代理之间的双向图来聚合网络代理用户的相似性并找出最新的潜在网络代理。

定义一个函数更形式化的描述被动发现方法, 将输入网络流量日志作为输入, 并在特定窗口时间内输出包含三元属性的二部图。一个函数  $\text{Fin}$  被定义为:  $\text{logs} \rightarrow G(\text{srcIP}, \text{dst}, e)$ , 其中  $\text{logs}$  表示数据集的记录, 集合  $\text{srcIP}$  表示客户端 IP 集合, 集合  $\text{dst}$  是目标 URL,  $e$  是边集( $e \in \text{srcIP} \times \text{dst}$ )。具体来说,  $\text{srcIP}$  和  $\text{dst}$  是两个不相交的点集, 边的权重是在一定时间段内生成的相同边的数量。

为了研究网络流量中用户访问行为相似性, 我们利用二分图的单模投影图来提取行为模式中节点之间的关系。图 3(a)表示了通过两个代理服务( $P1, P2$ )的六个用户 IP 地址( $C1-C6$ )和四个目的 URL( $d1-d4$ )之间的数据通信的简单图的例子。图 3(b)表示相应的二分图, 而图 3(c)是用户主机( $C1-C6$ )的顶点集上的二分图的单模投影。两个节点通过单模投影中的边连接, 当且仅当两个节点连接到二分图中有至少一个相同节点。我们利用单模投影图来探索用户的社交行为相似度。然而, 我们的兴趣并不是创建一个代理用户集群, 而是找出新的潜在的代理服务, 即使



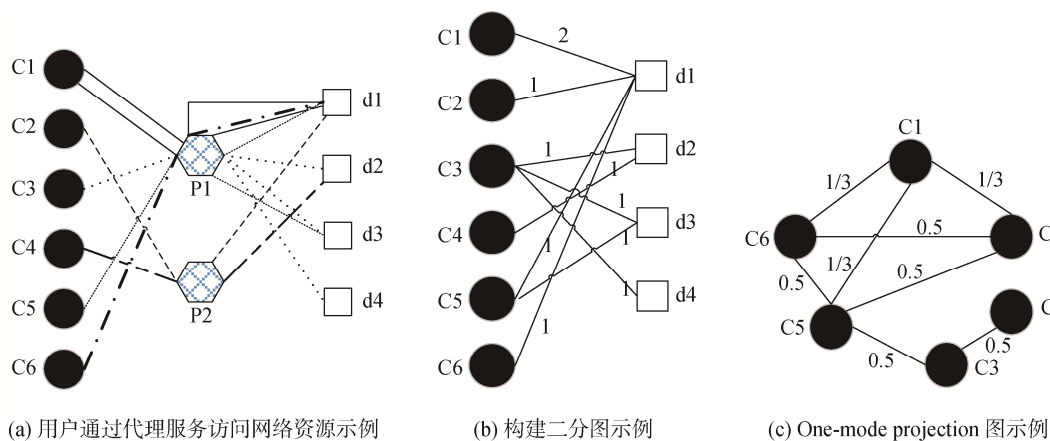


图 3 用二分图和单模映射对代理用户访问模式建模

Figure 3 Modeling the Proxy Pattern based on Bipartite and One-mode projection

它们可能是动态变化的。

### 3.3 ProxyMiner 方法概述

ProxyMiner 整个系统模块图如图 4 所示, 其主要分为主动发现模块和被动发现模块两部分。

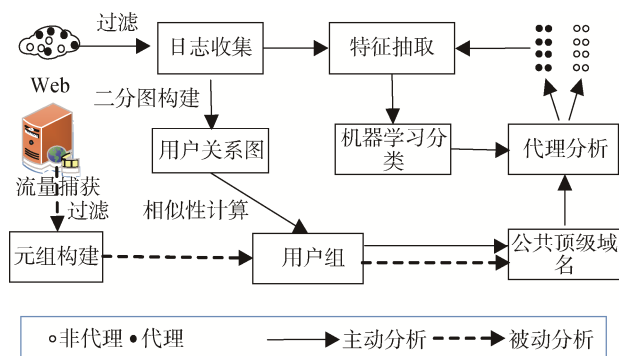


图 4 系统模块图

Figure 4 The Framework of ProxyMiner

主动发现模块主要是通过爬虫定期向相应的公开网页代理服务网站、论坛等方面去爬取相应的网页代理服务, 同时, 对有效的网页代理进行特征抽取。分别抽取网页的 URL 特征, 内容特征, DOM 特征。最后通过构建机器学习模型进行训练, 进而进行网页代理识别。被动发现模块主要是通过协议解析, 日志分析来构建二分图来对网络用户与其实际访问目的地之间进行抽象建模, 即客户端与目的地之间的双向节点。随后构建了二分图的单模投影, 从而可以进一步构建网络用户的相似性矩阵, 其相似性由两个主机之间的共同的目的地数量来表征。基于相似矩阵, 应用简单而有效的谱聚类算法来对用户群体聚类。这种行为模式聚类不仅减少了分析流量的规模, 而且还揭示了潜在的代理用户行为访问模式。最后, 我们不断获取代理用户群体访问的

顶级域名, 并将它们输送到验证模块(可以手动验证, 也可以批量自动验证), 手动验证就是通过浏览器设置网页代理服务后, 手动访问网络资源, 查看是否能够成功访问。对于自动验证, 本文基于 WebDriver<sup>[21]</sup>, 实现了模拟同时通过网页代理服务访问的验证小工具。对于验证完成后的网页代理服务, 可以进一步分析, 若发现新的且有效的网页代理服务特征, 可以进一步服务于主动方法的网页代理服务发现模块。

### 3.4 ProxyMiner 工作流程

#### 4. 主动发现

图 5 展示了主动发现模块的架构。从整体角度分析, 主动发现模块包含三个主要步骤。

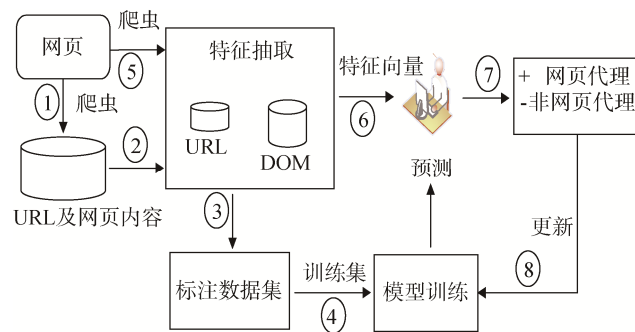


图 5 主动发现模块的流程图

Figure 5 The Process of Active Module

在第一步中, 通过基于 BeautifulSoup<sup>[22]</sup> (Beautiful Soup 是一个工具箱, 提供 web 数据解析等功能) 技术, 编写爬虫主动获取代理及非代理的网页数据集。

在第二步中, 进行代理特征抽取。分别抽取网页的 URL 特征, DOM 特征。

### 1) URL 特征

- a) 检测 URL 是否含有数字, 或者域名是 IP。
- b) 检测 URL 是否含有敏感词汇: 如 proxy, hide, block 等等。
- c) 检测 URL 是否含有特殊符号: 如 "-", "@" 等等。
- d) 检测 URL 是否含有嵌入域名, 如 http://rhe.rxxrh.com/http://www.google.com。
- e) 检测顶级域名是否使用非常用域名, 如 '.xyz' '.top' 等。
- f) 检测 URL 有效生存期是否过短。
- g) 检测每个 IP 承载的域名数量。如果一个 IP 承载多个域名, 那么特征向量对应的这一维度的值置为 1; 否则置为 0。

### 2) DOM 特征

#### a) 表单(Form)

**结构特征:** Form 表单里含有一个输入框, 输入框下方含有多个复选框(可选)。

**属性特征:** 复选框值含有敏感词汇; action 属性: action 属性值含有敏感字符串, 如 "Encrypt URL"、"Encrypt Page"、"Allow Cookies"、"Remove Scripts" 和 "Remove Objects" 等等。

b) Javascript 内容特征。据分析, 我们发现网页代理服务是提供中继服务的, 网页代理是通过 on-submit 函数来实现这一功能的, 相应的, 通过检测 on-submit 属性值是否包含字符串 "return update Location(this)"。若含有, 则特征向量对应的这一维值置为 1; 否则置为 0。

具体构建特征向量的方式有三种。URL 特征中的前 5 个特征是基于搜集大量网页代理 URL 数据集的基础上构建 URL 正则表达式, 通过字符串检测来获得的, 第 6 个特征是通过请求域名的 whois 信息计算相应的生存期来获得的。第 7 个特征是通过 nslookup 命令获取的。而 DOM 中的 Form 表单、复选框以及 Javascript 特征的检测都是通过 BeautifulSoup 的 python 解析器——lxml 解析器<sup>[23]</sup>进行解析分析。lxml 解析器基于 C 语言库 libxml2 和 libxslt, 对于 XML 文档解析有很快的速度和很强的容错能力。

第三步, 对于每一个网页, 通过第二步构建一个含有多维特征的向量, 作为训练集。

第四步, 为了验证特征抽取对代理识别的有效性, 本文分别在 URL 特征、DOM 特征和全集特征(FULL, 包括 URL 和 DOM 的所有特征)三种不同规模特征集上分别通过 Logistic Regression, Bayes, SVM-linear, SVM-RBF 四种机器学习方法构建训练

模型。

**Logistic Regression:** 这是一个简单的二分类模型: 基于距离分类超平面的距离进行分类。决策规则是基于符号函数  $\sigma(x) = [1 + e^{-x}]^{-1}$ , 具体公式如公式(4)所示, 其能将距离度量转化成判断代理与否的概率。

$$y = \sigma(w \cdot x + b), \quad (4)$$

其中, 参数  $w$  和  $b$  可以通过训练数据不断训练得到。

**Bayes:** Bayes 可以通过计算条件概率来获得, 如公式(5)所示。:

$$P(y = 0, 1|x) = \frac{P(x|y) \cdot P(y)}{P(x|y=1) + P(x|y=0)} \quad (5)$$

模型的判断是基于最大化特征的联合似然函数的值。

**Support Vector Machine (SVM):** SVM 分类器以最大化正确分类的边际为基准, 这对于微小的扰动是稳健的。SVM 中的决策规则基于计算特征之间相似性的核函数。在我们的研究中, 我们使用线性和 RBF 内核。

第五步, 测试集也是通过爬虫获取, 具体方式同第一步。

第六步, 抽取测试集的特征向量, 具体方式同第二步。

第七步, 通过训练出的模型分别进行代理识别并进行对比分析。

第八步, 对于获得的识别结果, 若存在误差, 可以将错误样例加入训练集中重新进行模型训练。

本文基于 Python Scikit-learn 库<sup>[24]</sup>集成了四种分类方法, 如 Logistic Regression、Bayes、SVM-linear、SVM-RBF。最后, 基于这四种机器学习方法训练模型的检测效果, 进行对比并对结果进行输出。对于每次最终输出的结果, 可以回馈到数据集, 便于更新下一轮的训练。

## B. 被动发现

图 7 展示了被动发现模块的框架。该模块的详细过程会在如下论述。

### 1) 访问模式测量

本节从测量的角度对我们的假设进行了有效验证和分析。

我们的假设是, 使用代理的用户的行为是相似的, 他们有一些固有的访问模式。也就是说, 用户倾向于寻求更高效的代理服务器, 他们通过代理服务更愿意访问那些仅能通过代理才能够使用的服务。当某个代理服务不起作用时, 代理用户会寻求其他的代理服务。

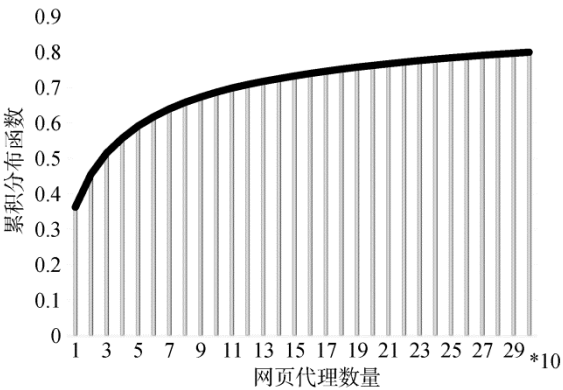


图 6 较流行的代理服务使用的累积分布图  
Figure 6 Cumulative Distribution of Popular Proxy Service

为了验证我们的假设, 我们对代理用户的访问模式进行了全面的测量研究。首先, 我们从一家大型机构获取真实网络流量, 该机构在 2017 年 1 月 3 日至 7 月 28 日的骨干路由器上部署了代理蜜罐(*PListI*)。然后, 我们分析并发现, 排名前 300 的域名大约提供了 80% 的代理服务, 正如其累积分布图 6 中显示的那样, 意味着我们可以仅关注那些经常使用的代理服务, 从而能够节省许多资源。

表 3 访问模式的稳定性  
Table 3 The Stability of Access Patterns

时间	种类(%)				
	社交网络	视频	文件分享	广告	色情
2017 年 1 月	22.3%	19.6%	16.1%	14.4%	12.9%
2017 年 3 月	20.5%	21.4%	13.8%	15.6%	14.1%
2017 年 5 月	19.3%	18.9%	15.7%	13.3%	13.6%
2017 年 7 月	20.8%	22.7%	14.8%	12.9%	14.7%
平均	20.73%	20.65%	15.10%	14.05%	13.80%
标准差	0.0107	0.0149	0.0089	0.0105	0.0060

另外, 我们还测量了代理用户潜在的较稳定访问模式。具体而言, 我们在收集数据集(2017 年 1 月)时记录了每个代理用户经常访问的网站类别, 然后每两个月重新检查其模式状态。结果总结在表 3 中。我们可以看到代理用户的访问模式是相对稳定的。平均七个月仅有微小的变化。另外更重要的是, 我们从 *PListI* 中随机选择 100 个代理用户, 并测试使用了多少个代理。我们发现 78% 的用户经常使用一种代理, 而其他用户至少使用两种代理。我们进一步调查发现, 用户更换为其他新的代理服务主要是因为旧的代理服务器不工作, 或者他们得到了更好的服务。因此, 受此发现的启发, 我们可以通过关注用户的

目的地主机来检测未知的代理。

2) 基于谱划分相似度矩阵的谱图聚类算法

在本节, 相似性度量  $S_{u,v}$  由单模投影图中的两个客户端主机  $u$  和  $v$  之间的加权边表示<sup>[25]</sup>, 因为加权边量化了在用户访问流量中的访问模式的行为相似度。设  $N(u)$  和  $N(v)$  分别表示两个客户端  $u$  和  $v$  通信的互联网主机的数量。然后, 我们使用  $W_{u,v}$  来表示单模投影中  $u$  和  $v$  之间边的权重, 如公式(6)所示。

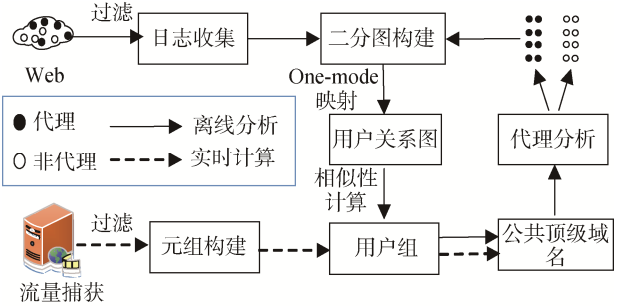


图 7 被动发现模块的流程图  
Figure 7 The Process of Passive Module

$$W_{u,v} = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$$

(6)

其中  $|N(u) \cap N(v)|$  表示在两个客户端  $u$  和  $v$  之间的二分图中共同目的地 URL 的总数, 而  $|N(u) \cup N(v)|$  表示  $u$  和  $v$  的各自的目的地的总数。请注意,  $u \neq v$ 。

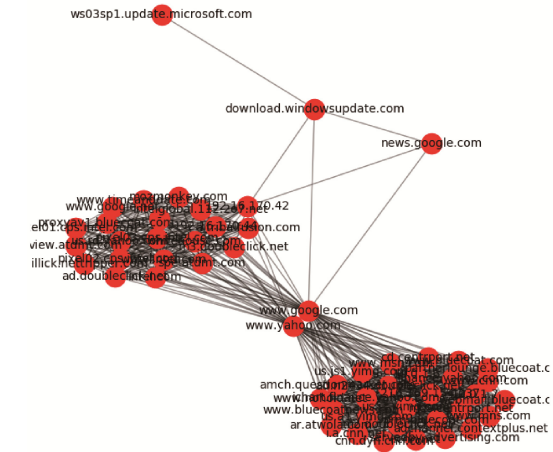


图 8 二分图的单模映射图  
Figure 8 The One-mode Projection of Bipartite Graph

一个关于主机通信的单模投影图的有趣观察是加权邻接矩阵中的聚类模式。图 8 中的散点图显示了两个不同群体的单模投影图, 即网页代理用户和非代理用户的目的地。这一观察促使我们进一步探索聚类技术和图划分算法, 以揭示网页代理服务用户特有的访问模式, 从而发现网页代理服务。我们的研究采用了一个简单的谱聚类算法<sup>[26]</sup>, 其中  $k=2$ (网



页代理服务用户群体和非网页代理服务用户群体)。

**算法 3.1:** 使用增广谱聚类算法发现代理用户行为聚类的算法

**输入:** 从代理蜜罐获得的网络流日志及非代理用户的日志

**输出:** 两个集群  $C_1, C_2$ , 其中 (1 表示为页代理服务; 0 代表非网页代理服务)

- 1: 基于网络流日志构建二分图.
- 2: 对二分图进行单模映射, 产生带权邻接矩阵及相似性矩阵  $S$ .
- 3: 计算对角矩阵  $A$ 。其中  $A(i, i) = \sum_{j=1}^n s_{i,j}$ .
- 4: 计算 Laplacian 矩阵  $L = A^{-1/2}SA^{1/2}$  并且计算出第二小的  $K$  个特征值.
- 5: **For**  $i = 1, 2, \dots, n$ , 其中  $y_i \in \mathbf{R}^k$  是  $S$  第  $i$  行向量.
- 6: 用 K-means 算法聚类成集群  $C_1, C_2$ .

算法 3.1 概述了该方法的主要步骤。算法的输入是一个三元组, 包含客户端 IP, 目标 URL 和在给定时间窗口(我们设置  $t = 5$  分钟, 可以在配置文件里设置)期间的权重(访问时间)。第一步是构造加权的二部图, 然后生成二部图的单模投影并获得相似度矩阵  $S$ 。因此, 我们应该找到一个图划分, 使得不同组之间的边具有非常低的权重(这意味着不同群体中的节点是不相似的), 并且群组中的边具有高权重。所以, 最简单直接的方法就是采用 mincut 策略。不幸的是, 由于 mincut 解的不平衡条件, 我们引入拉普拉斯矩阵  $L = A^{-1/2}SA^{1/2}$ , 其中  $A$  是对角矩阵,  $A(i, i) = \sum_{j=1}^n s_{i,j}, i = 1, 2, \dots, n$ , 我们利用 Rayleigh-Ritz 定理计算特征向量并找出  $L$  的第二小特征值。最简单的方法就是使用特征值的符号作为指标函数<sup>[26]</sup>。该算法的输出是客户端 IP, 并且每个 IP 地址都分配给一个具有相似的访问模式关系的群体。

### C. 代理验证

在获得网页代理群体之后, 我们会保留这些客户端 IP 的实时列表并提取其中所有经常访问的域名。

当我们使用网页代理进行浏览网站资源时, 网页代理的域名不会改变。从某种意义上说, 检查一个 URL 是否为网页代理的最有效和最简单的方法是检查我们是否可以通过代理 URL 成功访问受防火墙限制的某些网站。因此, 我们基于 WebDriver 开发了一个模拟访问工具。该工具可以远程模拟用户访问浏览器的行为来检查 URL 是否可以成功访问。如果成功, 该域名或 IP 可被视为一个网页代理服务。

## 4 实验结果与分析

### 4.1 实验环境

本实验的实验环境为一台 4 核 2.5GHz 主频 CPU, 4GB 内存的服务器(具体型号是 Intel(R) Xeon(R) CPU E5-2682 v4)。

主动方面是利用 Python Scikit-learn 实现了网页代理服务分类方法, 其中包括 Logistic Regression、Bayes、SVM-linear、SVM-RBF。

被动方面是利用 Python 里的复杂网络分析库 networkx 对代理用户的访问模式进行建模分析。

### 4.2 数据集

由于网页代理生存期是短暂的, 许多 URL 的生存期不超过一周, 许多管理员不断动态更新网页代理服务的域名。为了能在一个大规模的数据集里对发现方法进行评价和测试, 并且保证实验数据可以重复利用, 本文首先将相关的网页数据集(当网页代理服务可用时)通过爬虫爬取下来并同时保存, 然后后续的所有离线分析过程, 可以通过 python 里的 urllib.request 模块加载相应的网页数据进行离线分析。

主动发现的训练数据集主要有两类数据集: 网页代理数据集和非网页代理数据集。

代理网页: 包括主要爬取来自公开代理列表, 如网站 anonymster<sup>[24]</sup>, 网站 proxy4free<sup>[28]</sup>, 网站 PHPProxy, 网站 CGIPProxy 和网站 Glype<sup>[29]</sup>以及一些组织或机构公开的网页代理数据集(如 BlackLists<sup>[30]</sup>里的网页代理 URL)。

非代理网页: 非代理网页数据集主要是爬取来自 Alexa 网站<sup>[31]</sup>的前 26,066 条记录。

被动方面的数据集主要包含来自代理蜜罐的数千个网页以及从一个骨干路由器获取的一些日志。

### 4.3 实验结果

#### A. 主动发现

本文采用了常见的评价指标——准确率和召回率作为评价主动发现方法的评价指标。同时, 也使用集成了  $TP$ (True positive rate) 和  $FP$ (False positive rate) 两种指标的 F1 指标进行评测。最终, 本章还借鉴了  $ROC$  (receiver operating characteristic curve) 曲线及其覆盖的面积  $AUC$ (Area Under Curve) 指标, 对比了模型间的性能差异。

#### 特征对比

本章的第一个实验是为了验证不同的特征对代理识别的有效性, 分别在 URL 特征、DOM 特征和全集特征(FULL, 包括 URL 和 DOM 特征)三种不同规

模特征集上进行训练分析。

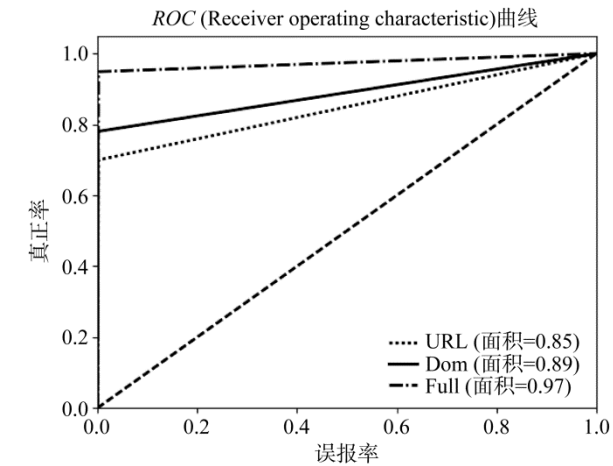


图 9 不同特征集下的 ROC 曲线  
Figure 9 ROC of different Feature Sets

表 4 不同特征集对分类性能的影响				
Table 4 The Performance on different Feature Sets				
特征集	特征构建时间(秒)	准确率(%)	召回率(%)	F1-score(%)
URL	39.53	96.00	71.50	82.00
DOM	1099.35	93.75	78.75	85.00
FULL	1143.24	95.00	96.50	95.25

图 9 展示了这三种不同特征向量的 ROC 曲线，可以发现：基于全集特征的方法具有较高的准确率。与图 9 相对应，表 4 列出了在不同特征集下的时间消耗及相应的准确率，召回率和  $F1\text{-score}$ 。可以发现，与构建基于 URL 特征的时间相比，抽取并构建基于 DOM 特征的需要消耗更多的时间。相反地，基于 DOM 特征的方法在召回率和  $F1\text{-score}$  方面均优于仅基于 URL 特征的。而基于全集特征的(结合 URL 特征和 DOM 特征)方法具有最好的性能。

分类方法对比

学习方法主要包括了四种分类器，因此本文评估了四种分类器在三种不同特征集下的分类效果。

表 5 给出了具有四种机器学习方法的三个特征集的结果。可以看出，产生最佳值的分类器是基于 URL 的特征的 LR 和 Bayes(几乎相同)，LR 基于 DOM 的功能和用“FULL”全集的 LR。其中，训练时间不包括特征提取时间(我们可以在表 4 中看到结果)。因此，通过广泛的实验结果，Logistic 回归(LR)在四种算法中表现最好。特别是，LR 的准确率比贝叶斯提高了 16%以上。更重要的是，结合四种模型的所有特征，大大提高了检测召回率。值得注意的是，

我们的实验显示的准确度约为 95%，平均召回率为 96.5%。

表 5 不同模型检测结果					
Table 5 The Results on different Models					
特征	方法	训练时间(秒)	准确率(%)	召回率(%)	F1-score
URL	LR	0.02	97	71	0.82
	Bayes	0.01	95	72	0.82
	SVM-linear	1.28	95	72	0.82
	SVM-RBF	2.82	97	71	0.82
	Average	1.03	96	71.5	0.82
	Std.	1.33	1.15	0.577	0
DOM	LR	0.02	97	77	86
	Bayes	0.01	81	82	81
	SVM-linear	1.28	97	78	86
	SVM-RBF	2.82	100	78	87
	Average	1.03	93.75	78.75	85
	Std.	2.33	8.62	2.22	2.71
FULL	LR	0.03	99	96	0.97
	Bayes	0.01	83	98	0.90
	SVM-linear	0.49	99	96	0.97
	SVM-RBF	1.43	99	96	0.97
	Average	0.49	95	96.5	0.95
	Std.	0.66	8	1	0.035

B. 被动发现

在本节中，我们评估被动方法发现代理服务的有效性。具体而言，我们要回答以下问题：第一，对于网页代理服务较短的使用期限特点，所提出的方法是否会能够找到一些之前未发现的网页代理(具有较好的可扩展性)。第二，对代理用户群体进行聚类分析是否是一种更有效的网页代理服务检测。在接下来的具体实验中，基于丰富的数据集，我们评估了上文提出的被动发现代理服务的方法。

表 6 列出了四个不同时期的统计结果。用户数量是指代理用户群中的源 IP 数。代理数目是指我们实验中真实代理的数量。发现数目是指被动发现的疑似网页代理服务的数量。验证代理数目代表经过验证后的真实代理的数目。召回率是指被动检测出的网页代理服务的比例。

通过分析表 6 的结果，我们进一步发现：最重要的时间开销是在用户之间构建二分图的时候。但是召回率与用户数量无关。我们使用两个关键指标进行了实验：密度和扩展度<sup>[32]</sup>，如图 10 所示。密度是验证时的代理检测率。更高的密度值意味着更高效地使用资源。密度越高，验证工作量越小。扩展度表

明了我们的方法为每个用户 IP 找到的新代理的平均数量。例如, 当网页代理群体中有 100 个用户 IP 时, 最终识别出了 130 个网页代理服务, 可以得出扩展度为 1.3。因此, 更高的扩展度表明, 对于每个客户端 IP, 会找到更多的网页代理。许多因素可能导致不同的扩展度, 例如新的逃避方式, 代理用户群体访问的 URL 等等。当扩展度变得越来越低时, 或许是到了更新聚类机制的时候。

表 6 统计结果  
Table 6 The Statistics Result

周期	用户数量	代理数目	发现数目	验证代理数目	召回率(%)	时间( $\mu$ s)
1	106	1794	2173	1548	86.29	1302.3
2	101	1244	1666	1062	85.37	1011.2
3	83	1345	5009	1200	89.22	941.0
4	38	237	3336	207	87.34	363.0

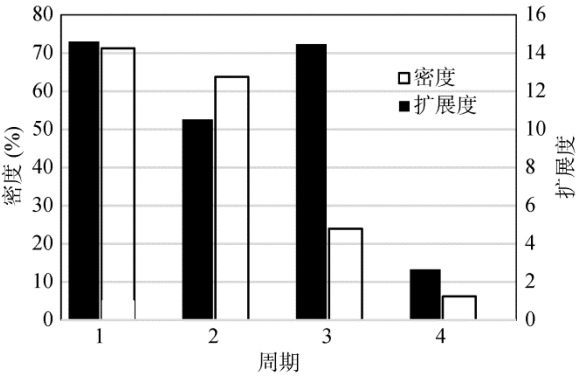


图 10 密度 vs. 扩展度  
Figure 10 Density vs. Expand

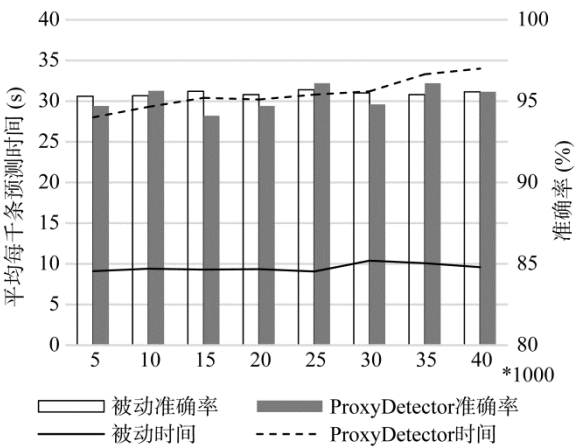


图 11 时间和准确率方面的对比

Figure 11 The Time and Accuraty vs. ProxyDetector

另外, 值得一提的是, 由于代理用户访问模式的稳定性, 我们不需要每天都构建二分图, 所以我

们不需要过度关注构建二分图的开销。

最后, 我们在检测的准确率和时间消耗方面对比了被动发现与 ProxyDetector<sup>[13]</sup>的差异, 如图 11 所示, 在准确率相当的情况下, 被动发现的预测时间大大减少, 更重要的是, 被动发现可以发现新的未知网页代理服务。

## 5 结论

代理服务是信息安全领域的热点问题。随着网络攻击技术和防御技术的不断发展, 该问题一直受到研究人员的广泛关注。针对这一问题, 首先梳理了代理服务的应用背景, 介绍了其基本概念, 并且从多角度分析了代理服务的不同类型划分, 然后, 对网页代理服务进行统一的形式化描述(忽略其具体使用的代理协议)。接着, 本文提出了一种基于主被动发现相结合的多维特征分析的网页代理服务发现方法——ProxyMiner。

主动发现方面, 本文提出了一种轻量级网页代理发现机制。实验结果表明, 相比于传统检测方法, 主动方法可以显著提高网页代理检测准确度, 大幅降低平均检测时间。在未来的工作中, 将重点围绕以下几个方面开展研究。首先, 引入 URL 白名单机制以避免一些不必要的资源消耗; 其次, 逐步考虑其他代理类型的检测方法, 尤其是挖掘其它类型代理服务的有效特征; 最后, 研究如何提高分类算法在动态环境下的鲁棒性。

被动发现方面, 提出了一种通过对代理用户进行聚类来发现网页代理服务的新方法。需要关注的一个挑战是网页代理服务在动态的环境中不断变化。为了应对这个挑战, 我们发现用户所使用的网页代理服务在不工作时寻求其他可用的网页代理服务器。因此, 我们的方法的亮点是提取代理用户访问模式中的固有特征, 而不是直接寻找网页代理服务的短期使用(动态更新)的特征, 即使用潜在的和稳定的访问模式来发现频繁变化的网页代理服务。通过我们的实验, 我们证明该方法可以有效发现网页代理服务, 并且优于以前提出的基于特征的方法, 特别适合用于发现未知网页代理。未来的工作包括如何挖掘更多网页代理服务特点, 并将其应用于网页代理服务发现。更重要的是, 应该考虑一些错误的负样本情况(例如, 一个用户可能首先在一个本地内容分发网络上登录, 然后再连接到网页代理服务器或者采用多个代理等方式)。

致谢 本文得到国家重点研发计划的资助和支持。

同时, 很多同行对本文的工作也给予了支持和建议, 在此一并表示感谢。

## 参考文献

- [1] 方滨兴, 定义网络空间安全, *网络与信息安全学报*, 4(1): 1-5, 2018.
- [2] Global Web Index Q4,2013-Q3,2014 based on the Internet users aged 16-64 <http://insight.globalwebindex.net/chart-of-the-day-90-million-vpn-users-in-china-have-accessed-restricted-social-networks?ecid=>.
- [3] Roberts H, Zuckerman E, Palfrey J. 2007 Circumvention Landscape Report: Methods, Uses, and Tools[J]. *Berkman Center for Internet & Society at Harvard University*, 2009.
- [4] Tsirantonakis G, Ilia P, Ioannidis S, et al. A Large-scale Analysis of Content Modification by Open HTTP Proxies[C]//Proceedings of the 24th Network and Distributed System Security Symposium (NDSS 2018). 2018.
- [5] Staniford-Chen S, Heberlein L T. Holding Intruders Accountable on the Internet[C]// Security and Privacy, 1995. Proceedings. 1995 IEEE Symposium on. IEEE, 1995:39-49.
- [6] Snort. <https://www.snort.org/>, 2016.
- [7] Peng P, Ning P, Reeves D S. On the secrecy of timing-based active watermarking trace-back techniques[C]// Security and Privacy, 2006 IEEE Symposium on. IEEE, 2006:15 pp.-349.
- [8] Aghaei-Foroushani V, Zincir-Heywood / N. A Proxy Identifier Based on Patterns in Traffic Flows[M]. IEEE, 2015.
- [9] Lin R M, Chou Y C, Chen K T. Stepping stone detection at the server side[C]// Computer Communications Workshops. 2011: 964-969.
- [10] He T, Venkitasubramaniam P, Tong L. Packet Scheduling Against Stepping-Stone Attacks with Chaff[J]. *Proc IEEE Military Communications Conf*, 2006:1 - 7.
- [11] Kumar R, Gupta B B. Stepping Stone Detection Techniques: Classification and State-of-the-Art[M]// Proceedings of the International Conference on Recent Cognizance in Wireless Communication & Image Processing. Springer India, 2016.
- [12] Shullich R, Chu J, Ji P, et al. A SURVEY OF RESEARCH IN STEPPING-STONE DETECTION[J]. *International Journal of Electronic Commerce Studies*, 2011, 2(2).
- [13] Zhang Y, Paxson V. Detecting stepping stones[C]// Conference on Usenix Security Symposium. USENIX Association, 2001: 263-279.
- [14] Deng Z, Liu Z, Chen Z, et al. The Random Forest Based Detection of Shadowsock's Traffic[C]//Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2017 9th International Conference on. IEEE, 2017, 2: 75-78.
- [15] Vasilomanolakis E, Karuppayah S, Fischer M, et al. This network is infected: HosTaGe-a low-interaction honeypot for mobile devices [C]. In Proceedings of the Third ACM workshop on Security and privacy in smartphones & mobile devices. San Francisco, CA, USA, 2013: 43-48.
- [16] Nicomette V, Kaâniche M, Alata E, et al. Set-up and deployment of a high-interaction honeypot: experiment and lessons learned [J]. *Journal in Computer Virology*, 2011, 7(2): 143-157.
- [17] Zhuge JW, Tang Y, Han XH, Duan HX. Honeypot technology research and application [J]. *Journal of Software*, 2013, 24(4): 825-842 (in Chinese).  
(诸葛建伟, 唐勇, 韩心慧, 段海新. 蜜罐技术研究与应用进展 [J]. *软件学报*, 2013, 24(4): 825-842.)
- [18] IP2Proxy, <http://www.fraudlabs.com/ip2proxy.aspx>, 2017.
- [19] CIPAFILTER, <https://cipafilter.com/>, 2017.
- [20] MaxMind, <https://www.maxmind.com/>, 2017.
- [21] SeleniumWebDriver. <http://docs.seleniumhq.org/projects/webdriver/>, accessed :16/11/2017.
- [22] BeautifulSoup,<https://www.crummy.com/software/BeautifulSoup/>, 2017.
- [23] lxml parser, <http://lxml.de/>, 2017.
- [24] Scikit-learn, <http://scikit-learn.org/stable/>, 2017.
- [25] Xu, Kuai, Feng Wang, and Lin Gu. "Behavior analysis of internet traffic via bipartite graphs and one-mode projections." *IEEE/ACM Transactions on Networking (TON)* 22.3 (2014): 931-942.
- [26] Luxburg U. A tutorial on spectral clustering[M]. Kluwer Academic Publishers, 2007.
- [27] Anonymster. <https://anonymster.com/best-free-web-proxy-sites-list/>, 2017.
- [28] Proxy4free, <http://www.proxy4free.com/list/webproxy1.html>, 2017.
- [29] J. Brozycki. Detecting and preventing anonymous proxy usage, SANS Inst, 2008.
- [30] URL blacklist. <http://urlblacklist.com>, 2017.
- [31] Alexa. <http://www.alexa.com/>, 2017.
- [32] Invernizzi, L., et al. Evilseed: A guided approach to finding malicious web pages. 2012.



**陈志鹏** 于 2014 年在北京邮电大学计算机技术专业获得硕士学位。现在中国科学院大学网络空间安全专业攻读博士学位。研究领域为网络安全、大数据处理和挖掘。Email: [chenzhipeng@iie.ac.cn](mailto:chenzhipeng@iie.ac.cn)



**张鹏** 于 2013 年在中国科学院计算技术研究所获得博士学位。现在中国科学院信息工程研究所副研究员, 主要研究方向为服务计算和大数据处理和挖掘。Email: [pengzhang@iie.ac.cn](mailto:pengzhang@iie.ac.cn)



**黄彩云** 于 2015 年开始在中国科学院信息工程研究所就读硕博研究生, 主要研究方向为网络安全、大数据处理和挖掘。Email: [huangcaiyun@iie.ac.cn](mailto:huangcaiyun@iie.ac.cn)



**刘庆云** 于 2015 年在中国科学院计算技术研究所获得博士学位。现在中国科学院信息工程研究所高级工程师, 主要研究方向为网络安全和大数据挖掘。Email: [liuqingyun@iie.ac.cn](mailto:liuqingyun@iie.ac.cn)



**邢丽超** 中国科学院大学硕士研究生。就读中国科学院信息工程研究所。于 2017 年从西南交通大学获得通信工程学士学位。主要研究方向为信息过滤与内容计算。Email: [xinglichao@iie.ac.cn](mailto:xinglichao@iie.ac.cn)