面向高速网络流量的恶意镜像网站识别方法

张蕾[1,2]，张鹏[2]，孙伟[3]，杨兴东[4]，邢丽超[1,2]

（1. 中国科学院大学网络空间安全学院，北京 100049；2. 中国科学院信息工程研究所，北京 100093；
3. 北京交通大学计算机与信息技术学院，北京 100044；4. 北京航空航天大学计算机学院，北京 100191）

摘 要：针对网络环境中造成危害的信息通过镜像网站进行传播从而绕过检查的问题，提出了面向高速网络流量的恶意镜像网站识别方法。首先，从流量中提取碎片化数据并且还原网页源码，同时加入标准化处理来提高识别准确率；然后，将网页源码分块，利用相似度散列算法对每个网页源码分块计算散列值，得到网页源码的相似度散列值，同时引入海明距离来计算网页源码之间的相似性；最后，截取网页快照，提取其 SIFT 特征点，通过聚类分析和映射处理得到网页快照的感知散列值，通过感知散列值计算网页相似性。在真实流量下的实验表明，所提方法的准确率为 93.42%，召回率为 90.20%，*F* 值为 0.92，处理时延为 20 μs。通过所提方法，在高速网络流量下可以有效地检测恶意镜像网页。

关键词：恶意镜像网站；相似度散列算法；网页相似性

中图分类号：TP309

文献标识码：A

doi: 10.11959/j.issn.1000−436x.2019089

IMM4HT: an identification method of malicious mirror website for high-speed network traffic

ZHANG Lei[1,2], ZHANG Peng[2], SUN Wei[3], YANG Xingdong[4], XING Lichao[1,2]

1. School of Cyber Security, University of Chinese Academy of Sciences, Beijing 100049, China
2. Institute of Information Engineering, Chinese Academy of Sciences, Beijing 100093, China
3. School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China
4. School of Computer Science and Engineering, Beihang University, Beijing 100191, China

Abstract: Aiming at the problem that some information causing harm to the network environment was transmitted through the mirror website so as to bypass the detection, an identification method of malicious mirror website for high-speed network traffic was proposed. At first, fragmented data from the traffic was extracted, and the source code of the webpage was restored. Next, a standardized processing module was utilized to improve the accuracy. Additionally, the source code of the webpage was divided into blocks, and the hash value of each block was calculated by the simhash algorithm. Therefore, the simhash value of the webpage source codes was obtained, and the similarity between the webpage source codes was calculated by the Hamming distance. The page snapshot was then taken and SIFT feature points were extracted. The perceptual hash value was obtained by clustering analysis and mapping processing. Finally, the similarity of webpages was calculated by the perceptual hash values. Experiments under real traffic show that the accuracy of the method is 93.42%, the recall rate is 90.20%, the *F* value is 0.92, and the processing delay is 20 μs. Through the proposed method, malicious mirror website can be effectively detected in the high-speed network traffic environment.

Key words: malicious mirror website, simhash algorithm, webpage similarity

收稿日期：2018−11−09；**修回日期**：2019−03−04

通信作者：张鹏，pengzhang@iie.ac.cn

基金项目：国家重点研究发展计划基金资助项目（No.2016YFB0801300）；国家自然科学基金资助项目（No.61602474, No.61602467, No.61702552）

Foundation Items: The National Key Research and Development Program of China (No.2016YFB0801300), The National Natural Science Foundation of China (No.61602474, No.61602467, No.61702552)

1 引言

镜像网站是将主/源网站的全部内容或部分内容上传到不同服务器，同时拥有自己的域名和 URL 的网站，被称为主/源网站的镜像网站。镜像网站的主要作用是当网站流量过高时对服务器进行减压和分流，提高不同地区或不同 ISP 用户的访问速度，保存历史性信息和数据。镜像网站主要有 2 种网页表现形式：一种是内容完全重复的镜像网页；另一种是部分内容相同的近似镜像网页，这种近似镜像网页在网页格式、网页标题、网站签名等方面与主/源网站有所不同，但所呈现的主体内容都是相同的。镜像网站在提供便利的同时，也带来很多问题和危害[1]。

在恶意镜像层面，当正常网页被恶意镜像后，正常网页的数据被盗用且内容被同步。网站镜像在占用资源的同时，也会影响正常网页的加载速度。当网页被恶意镜像后，如果不及时处理将会面临被搜索引擎降权，甚至不被搜索引擎收录的危险，尤其是对于一些刚上线没有权重的新网页。如果恶意镜像的网页挂有正常网页的联盟广告，当用户点击这些广告时，很容易造成用户账号被封。

在非法网页层面，例如通过租用一些共用的 IP 主机，将非法内容拷贝到镜像主机中（去除一些未加密的非法内容）。大部分恶意网页可以通过这种镜像手段绕过管控，令非法内容继续传播。虽然我国对此类网站的治理比较严厉，但是在镜像网站足够多的情况下，进行逐一治理比较困难，从而导致一些恶意网站的治理失败。

此外，更有一些非法分子对知名商业网站或金融网站进行镜像，实施诈骗，因此，对于镜像网站的检测是极其必要的。为此，本文提出了一种面向高速网络流量下恶意镜像网站的检测方法，主要创新点如下。

1) 提出了一种面向高速网络流量的网页源码相似度计算方法。该方法从高速网络流量中将碎片化数据拼接成完整的网页源码，对拼接的网页源码实现 Gumbo 标准化处理，并对标准化后的网页源码进行分块处理，通过 simhash 算法对每个块计算散列值，得到网页源码的 simhash 值，引入海明距离来计算网页源码之间的相似性。

2) 提出了一种基于网页快照相似度的镜像网页识别方法。当网页源码之间的相似性在设定的阈值范围内时，该方法提取网页图像的 SIFT（scale-invariant feature transform）特征点[2]，通过聚类分析和映射处理得到网页图像的感知散列值。当被测网页与基准网页的图像感知散列值的相似性大于设定的阈值时，判定该网页为与基准网页相对应的镜像网页。

目前，大多数恶意镜像网页的检测方法都是通过主动采集的方式来获取网页信息进行检测，使时间开销较大。本文所提检测方法的核心创新之处在于能够同时兼顾性能和准确率，通过被动的方式，直接对网关流量进行实时解析，从流量中还原得到网页信息，这大幅度降低了时间开销；同时通过与基准网页进行匹配，完成识别。与其他方法不同的是，本文并不是只通过网页内容或视觉特征进行检测，而是将网页内容与网页快照进行组合判断，在兼顾性能的同时还能有效地提高识别准确率。

2 相关工作

现有的镜像网站检测方法有很多种，这些方法大都是从网页文本中提取相应特征进行检测，具体分为以下 3 种。

1) 基于网页正文内容的相似性检测

网页正文内容是指网页信息的主体部分，而不是指网页中的所有文本内容。网页正文内容主要通过对网页进行 HTML 标签去噪处理后，通过一定的算法得到。基于网页正文内容的相似性检测分为两类[3]：基于聚类的网页相似性检测和基于特征匹配的网页相似性检测。

基于聚类的网页相似性检测方法以向量空间模型（VSM, vector space model）[4]为基础进行网页相似性检测，这类算法的缺点是表示网页的向量模型维度过高，使整个算法的计算时间过长，不适用于大量的网页相似性检测系统。基于特征匹配的网页相似性检测算法，主要提取的网页特征有特征词、特征串和特征句这 3 种，对整个网页进行分词并提取特征词，用特征词集合来表示网页，通过计算集合中的特征词的指纹来计算网页的相似性。这类算法目前典型的有 simhash 算法[5]、I-match 算法[6]等。

2) 基于链接的相似性检测

此类方法主要是通过比较网页的入链（即连向网页的 URL）信息是否一致来确定网页之间的相似性，当 2 个网页的 URL 相近时，判断这 2 个网页互为近似的镜像网页。目前，常用的基于 URL 检测的方法有基于散列检测[7]算法、基于 Bloom filter

检测[8]算法、基于深度学习检测算法[9]等，这类算法的不足之处在于当网页的入链较少，或者当网页是全新的网页时，相似的网页很难被检测出来。随着网络规模的日益增长，大多数网站都开辟了子域名，同一网站下相同的资源拥有不同的访问地址，而使用基于链接的镜像网页检测方法是无法检测出来的，此类方法对镜像网站的检测效果越来越不尽人意。

3) 基于视觉的相似性检测

基于视觉的检测融合了网页的文本、结构、图片、整体视觉效果等可视化特征，以视觉相似这一核心特点为出发点来分析表征网页身份的特征集合。Liu 等[10]从视觉特征入手，基于 DOM（document object model）树相似性对网页进行检测。Mao 等[11]基于网页视觉相似性提出了一种量化可疑性的算法。Zhang 等[12]使用页面图像特征与词汇特征来进行检测。虽然基于视觉特征的相似性检测误报率较低，但是计算复杂度较高。

在高速网络流量环境下，本文提出的方法先从流量中提取碎片化数据还原网页源码，然后将网页源码分块，利用相似度散列算法通过计算每个块的散列值，得到网页源码的相似度散列值，同时引入海明距离来计算网页源码之间的相似性。最后截取网页快照，提取其 SIFT 特征点，通过聚类分析和映射处理得到网页快照的感知散列值，计算感知散列值得到网页相似性。实验表明所提方法既具有较高的准确率和召回率，也能适应高速网络流量的应用环境。

3　方法原理

本文方法对恶意镜像网页的检测数据主要从高速网络流量中获取，这样可以降低时间成本，达到较高的准确率。在高速网络流量环境下有着海量的网站，并且网站的应答信息呈现出多源链接及碎片化的特征，接收到的网页内容可能来自不同的服务器，并且呈现出无序、杂乱和碎片化的状态，这给标识、拼接网站流量还原网站内容带来了严重的问题。

本文方法在实验室已有底层平台的基础[13]，研究并开发了一套可以完整还原网页内容的程序，从网络流量中解析网页数据，具体过程如图 1 所示。

首先将网络流量接入底层平台，经过 IP 碎片重组和 TCP/UDP 流还原后得到完整的分组段。然后对所需要的 HTTP 流量进行识别，得到 HTTP 会话（即请求/应答分组对）。分析请求分组和应答分组的开始行，得到网页的 URL；对分组头部进行分析得到 HTTP 分组头域；对分组实体部分解 Gzip 压缩和解 chunk 编码，可以得到服务器返回的网页实体内容。到此为止，可以得到不同网页的网页数据。

3.1　网页数据拼接

由于通过解析得到的网页实体内容并不一定是完整的，还需要对网页实体内容进行拼接，从而得到完整的网页数据，拼接流程如图 2 所示。

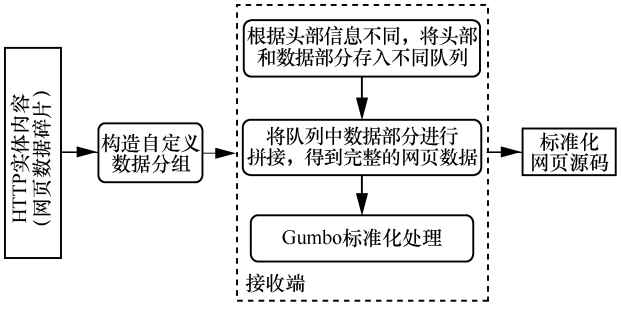

图 2　网页实体内容拼接流程

针对网页数据碎片，通过构造自定义数据分组的形式对传递到接收端的网页实体内容进行拼接。自定义数据分组格式如图 3 所示。

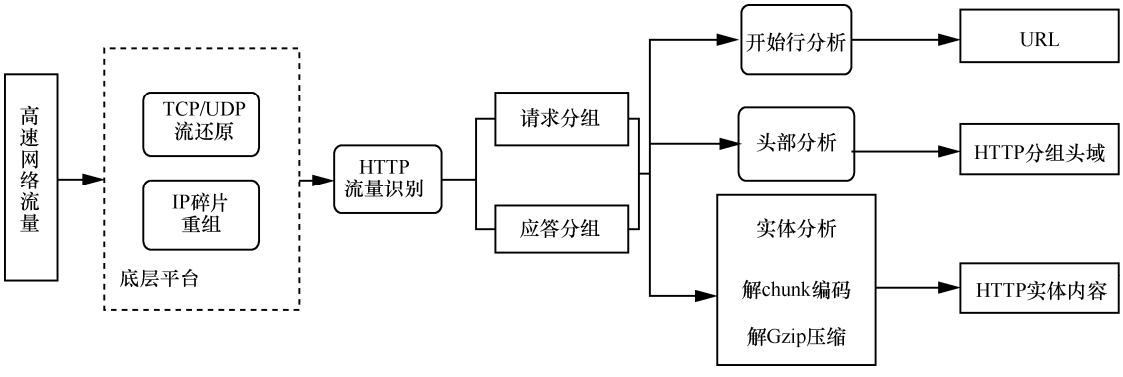

图 1　高速网络流量的数据解析过程

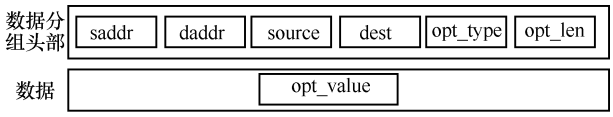

图 3　数据分组格式

数据分组头部由源 IP（saddr）、目的 IP（daddr）、源端口（source）、目的端口（dest）、数据类型（opt_type）和数据长度（opt_len）组成。数据部分（opt_value） 有 7 种类型，包括四元组（tuple4）、网页链接（url）、编码（charset）、数据块序号（http_seq）、数据开始（content_begin）、数据内容（content_data）、数据结束（content_end）。接收端根据每次接收的数据分组头部选项判断接收分组的类型，将数据分组放入对应的队列内。最后将队列中数据部分进行拼接，得到完整的网页数据，采用 Google 的 Gumbo 算法对网页内容进行对齐，补齐缺失的标签，得到标准化的网页源码。

3.2　数据分块流程

目前，常用的数据分块技术主要有固定大小分块技术（FSP, fixed-sized partition）和基于内容的分块重删检测技术（CDC, content defined chunking）[14]，这 2 种技术虽然达到了不错的去重效果，但对于数据块来说，少量字符的变化将会导致整个数据块“标识”的不同，从而影响相似性判定结果。本文方法所采用的分块流程如图 4 所示。

在本文方法中，对所获取的网页源码 Gumbo 标准化处理后再进行分块处理，这里所采用的分块技术是将源码按行分块，将分块后的结果存入列表中等待下一步处理。

3.3　网页源码特征提取

simhash 算法[5]是一种局部敏感散列算法，与传统的散列算法最大的不同点就在于对于 2 篇相似的文档，通过局部敏感散列算法计算出来的散列值也是相似的。本文利用 simhash 算法为每一个网页生成一个 fingerprint，使用海明距离计算相似性。simhash 算法计算流程如图 5 所示，分为 5 个步骤。1) 分词：提取特征项，提取出 n 个（feature weight, weight）对。2) 散列：利用散列函数计算得到各特征项的散列值，得到 n bit 的签名。3) 加权：对提取的特征向量进行加权。4) 合并：将每个特征项的加权值累加，得到一个序列串。5) 降维：对得到的 n bit 的序列串，序列串中小于 0 的位置为 0，否则置为 1。最终得到 fingerprint，即文档的 simhash 值。

从高速网络流量中获取网页的源码内容后，计算网页源码的 64 位 simhash 值，与基准网页的 simhash 值进行对比，判断相似性。计算网页源码的 simhash 值主要是将 simhash 与上述分块技术相结合，计算网页源块的散列值来计算整个源码的 simhash 值，计算过程如图 6 所示。

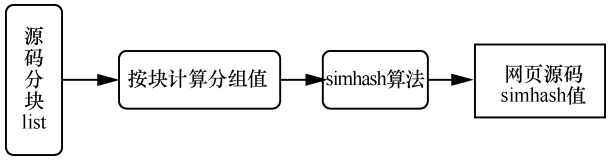

图 6　网页源码 simhash 值计算过程

这里所采用的 simhash 是在原 simhash 算法的基础上进行修改。首先，不再选择特征项并计算其权重，而视分得的每一个块为一个特征量；然后，不再根据特征项的出现次数设定不同的权重，而是设定每个特征项的权重为 1。

3.4　网页源码相似性计算

比较网页源码相似性流程如图 7 所示。

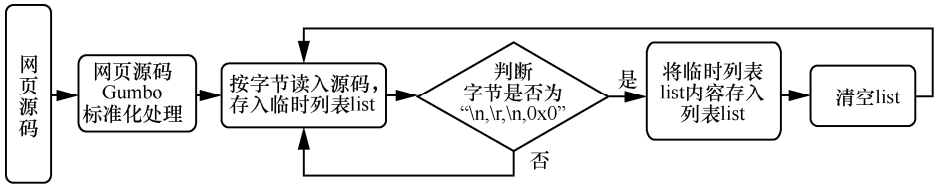

图 4　数据分块流程

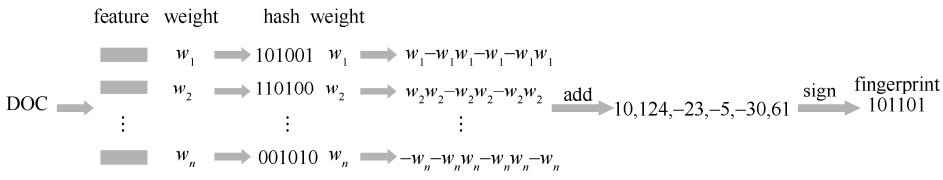

图 5　simhash 计算流程

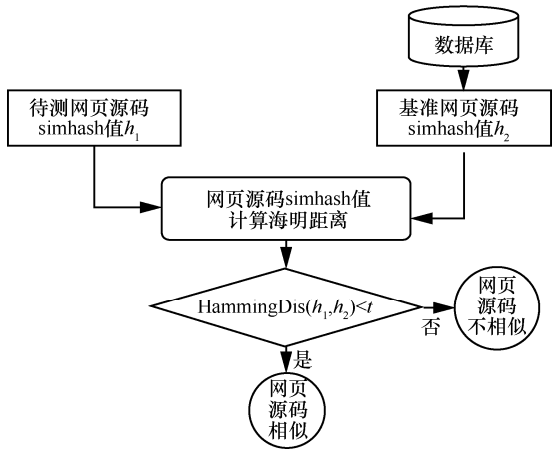

图 7　网页源码相似性比较

得到网页源码的 simhash 值以后，计算与数据库中基准网页的 simhash 值之间的海明距离，当海明距离小于设定的阈值 t 时，判断网页源码相似；否则，网页源码不相似。

3.5　网页快照特征提取

尺度不变特征变换（SIFT, scale-invariant feature transform）是图像的局部特征，顽健性较好。Mikolajzcyk 等[15]在对各种图像描述子进行测评后，认为 SIFT 描述子具有最佳性能，但是 SIFT 特征的计算量较大，对于实时性要求较高的场合，其处理速度过于缓慢。因此，本文采用图像 SIFT 特征提取与图像感知散列相结合的方式来计算网页快照的指纹，采用 SIFT 特征提取网页快照的局部稳定特征点，通过聚类分析对提取出的特征数据进行压缩，再利用图像感知散列得到最后的散列值，网页快照的相似性通过海明距离进行评测。

本文所采用的方法主要分为网页快照特征提取、特征数据压缩、最终映射这 3 个步骤，其中特征数据压缩涉及 SIFT 特征提取、聚类分析和图像感知散列这 3 个关键技术。

1) 网页快照特征提取

利用 SIFT 特征提取网页快照的局部稳定特征点，得到特征点集合，经过压缩得到中间散列值。

2) 特征数据压缩

通过聚类分析，将上一步得到的中间散列值进一步映射为最终散列值，这一步强调的是信息的压缩表示。

图 8 为特征数据压缩的具体流程，详细描述如下。

① 预处理

首先对网页快照进行预处理，将彩色图像进行灰度处理，然后采用双三次插值法将图像大小规格化，统一成大小为 $k\times k$ 的图像 $I_{k\times k}$，在本系统中，采用的是 256像素×256 像素。

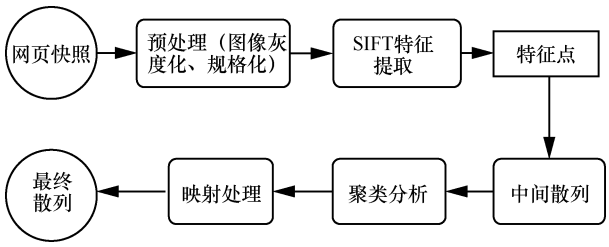

图 8　特征数据压缩流程

② SIFT 特征提取

利用 SIFT 特征，提取图像 $I_{k\times k}$ 的局部稳定特征点，得到网页快照的特征向量。

$$\boldsymbol{F}=\{\boldsymbol{F}_1,\boldsymbol{F}_2,\cdots,\boldsymbol{F}_n\},\ \boldsymbol{F}\in Z^{128} \tag{1}$$

其中，$\boldsymbol{F}_i(1\leqslant i\leqslant n)$ 表示所提取出的 1×128 维 SIFT 特征向量；集合 $\boldsymbol{F}$ 为 $n\times128$ 维的特征向量，表示提取出的所有局部稳定特征点矢量。

③ 特征点压缩

在上一步中得到了 $n\times128$ 维的 SIFT 特征点的特征向量 $\boldsymbol{F}$，将特征向量 $\boldsymbol{F}$ 按式(2)进行处理。

$$\boldsymbol{G}(i)=\sum_{j=1}^{n}\boldsymbol{F}_{i,j},1\leqslant i\leqslant128 \tag{2}$$

即将特征向量 $\boldsymbol{F}$ 按行求和，得到中间散列 $\boldsymbol{G}$，$\boldsymbol{G}$ 为 1×128 维的向量，达到了压缩特征点的目的。

④ 聚类分析

计算中间散列值的分布质心 C，如式(3)所示。

$$C=\frac{\sum_{i=1}^{n}\boldsymbol{F}_i}{n} \tag{3}$$

其中，质心 C 表示特征向量 $\boldsymbol{F}$ 的均值。这里根据质心 C 来决定簇类，小于 C 的被分为 C_1 簇，大于 C 的被分为 C_2 簇。

3) 最终映射

根据聚类结果，将中间散列映射为“0,1”序列，具体映射规则如式(4)所示。

$$g(i)=\begin{cases}0, & \boldsymbol{G}(i)\in C_1\\ 1, & \boldsymbol{G}(i)\in C_2\end{cases} \tag{4}$$

其中，$C_1<C_2$，最后得到 128 位散列值。

3.6　网页快照相似性计算

通过 3.5 节所介绍的算法，根据网页标题对所需测试的网页进行初步过滤，然后计算网页源码相

似性和网页正文相似性，当其阈值在设定的范围内时，提取网页快照的 SIFT 特征点，通过聚类分析和映射处理得到网页快照的感知散列值。当被测网页与基准网页的图像感知散列值相似性大于一定阈值时，判定该网页是恶意镜像网页。

本文采用海明距离来判定网页快照的相似性。首先，通过 3.5 节所介绍的算法计算出待测网页快照的 128 位图像散列值 ，基准网页图像快照的散列值为 g_0，计算 g_t 与 g_0 之间的海明距离，如式(5)所示。

$$\text{HammingDis} = g_t - g_0 \tag{5}$$

其中，“−”表示计算 g_t 与 g_0 之间的海明距离。根据计算出来的海明距离，可以得到待测网页快照与基准网页快照之间的相似度，如式(6)所示。

$$S = \frac{128 - \text{HammingDis}}{128} \tag{6}$$

S 的取值范围为[0, 1]，通过判断 S 与阈值之间的大小来判定网页快照之间的相似度。当 S 大于阈值时，待测网页快照与基准网页快照判定为相似；否则判定为不相似。

4 实验与分析

本文所检测恶意镜像网页包括钓鱼网页、网络博彩、低俗内容、其他违法违规等网页，利用 Python 爬虫程序爬取 Phishtank[17]上公开钓鱼网页，其余三类网页由于没有公开的数据集，通过捕获网关流量的方式进行筛选查找，最终获取到 4 369 个上述恶意网页。其中，网络博彩类（1 462 个）和低俗内容类（968 个）网页大多通过图片和镜像的方式；钓鱼网页类（2 653 个）和其他违法违规类（714 个）网页大多采用对源码进行修改的方式，通过提取网页的源码特征和网页快照特征，构建基准网页数据库。

4.1 测试环境

数据库版本为 Oracle 12c Relase 2。

实验环境如下。

CPU：Intel(R) Core(TM) i5-4210H CPU @ 2.90 GHz。

内存：3 GB。

操作系统：Red Hat Enterprise Linux Server release 7.2。

编程语言：C 语言、Python2.7.5。

4.2 实验步骤

步骤 1 采集基准网页信息，提取基准网页源码和网页快照特征，并存入数据库。

步骤 2 更改网卡配置，接入网关流量，进行 7 天流量检测。

步骤 3 根据第 3 节中介绍的流量数据解析流程，对接入的流量进行解析，得到 URL、HTTP 分组头域和 HTTP 实体内容，并以自定义数据分组的形式将解析出的网页信息传递到接收端。

步骤 4 接收端接收到数据分组后，对网页数据碎片进行拼接得到完整的网页数据，并对网页源码标准化处理。

步骤 5 利用网页源码特征提取模块提取网页的特征，并读取数据库中存储的基准网页数据进行相似性对比，采用海明距离判断网页源码的相似性。当相似性小于设定的阈值时，进行步骤 6，否则直接将其过滤。通过实验发现，当阈值取 3 时，能达到最佳的过滤效果。

步骤 6 通过网页快照相似性比对进一步提高实验结果准确率。

4.3 实验结果与分析

通过对测试数据经过 URL 初步过滤后，得到 194 714 个网页。将可疑网页与数据库中的基准网页进行源码和快照的相似性比对，查找出目标网页，即与基准网页对应的恶意镜像网页和近似恶意镜像网页。实验检测出的目标网页结果如图 9 所示。

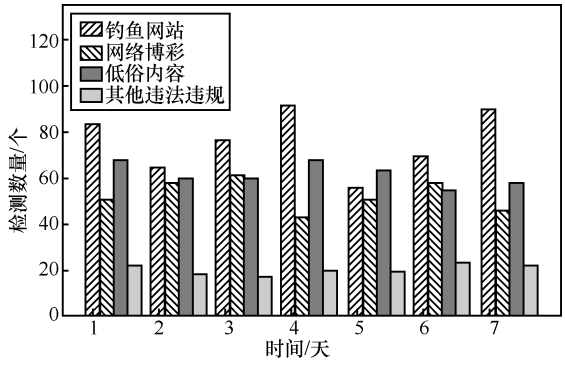

图 9 检测结果

图 9 中分别展示了每天所检测出的 4 种类型的恶意网页数，这 4 种类型分别与各自的基准网页对应，符合本文对恶意镜像网页的定义。通过人工评测，除了正确检测出的目标网页外，还存在一些误判的网页，其中，钓鱼网页类有 31 条，网络博彩类有 24 条，低俗内容类有 29 条，其他违法违规类有 19 条。产生误判是因为恶意镜像网页有时效性，大多数恶意镜像网页在很短时间内就会失效，而本

文是实时对网关流量进行监测，通过本文所提方法对网关流量中的网页数据进行还原并与基准网页进行比对，进行判断是否为恶意镜像网页；基准网页中的数据需要提前采集，由于公开的数据集较少，需要捕获网关流量通过自动与人工相结合的方式进行收集并更新，会花费一定的时间，因此在基准网页与实时网关流量中的网页数据之间会有一定的时间差，导致出现误判的情况。实验发现，本文所提恶意镜像网页检测算法总的准确率为 93.42%。由于预先无法得知测试数据中所有的恶意镜像网页与近似恶意镜像网页总数，通过人工查找得到总数为 1 623 条，由此计算该算法的召回率为 90.20%，F 值为 0.92。从而可以看出，本文所提出的大规模网络流下恶意镜像网页检测算法是有实际意义和有效性的。

本文在相同的实验环境以及硬件设备下，采用同一组基准网页，对基于 URL 相似度、基于内容相似度、基于视觉相似度等常用的镜像网页检测算法和本文所采用的方法进行对比，表 1 为实验测试结果。从实验结果可以看出，只对网页内容进行检测的准确率仅比基于 URL 的检测算法准确率高，这是因为存在一些恶意镜像网页的网页源码中并没有实质性的内容，而主要是一些 Javascript 链接，因此仅通过网页内容进行检测对该类网页不能有效地识别，降低了检测的准确率。基于 URL 相似度的检测算法的准确率相对较低，这类方法的不足之处在于当网页的入链较少时，或者当网页是全新的网页时，是很难检测出来的。基于网页视觉的检测算法比前 2 种算法的准确率都高，但比本文的检测算法的准确率略低，且需要通过主动采集的方式，时间花销较高，且计算复杂度较高。

表 1　　恶意网页检测算法对比

算法	准确率	召回率	F 值
基于 URL 相似度[9]	80.58%	75.32%	0.778 6
基于内容相似度[5]	87.64%	84.65%	0.861 2
基于视觉相似度[11]	91.71%	89.23%	0.904 5
IMM4HT	93.42%	90.20%	0.917 8

相较于几种常见的算法，本文所采用的主被动相结合的面向高速网络流量的恶意镜像网站识别方法（IMM4HT, an identification method of malicious mirror website for high-speed network traffic）在准确率和召回率都有较大的提升。分析其原因，是因为从网关实时捕获真实的流量并还原网页数据，能够得到最原始的网页数据；同时，计算网页源码的相似性和网页快照的相似性进行集成判断，能够有效地提升分类准确率。综合以上评价指标，本文所采用的方法取得了较好的实验结果，更适用于高速的网络流环境，在具有较高准确率的同时能够保证一定的性能。

下面考察 IMM4HT 在不同网页规模情况下的性能，实验效果如图 10 所示。图 10 表示了不同网页规模下执行所需的时间变化情况，其中，p 为大于 68 KB[1]的网页在实验数据集中所占比例。为了

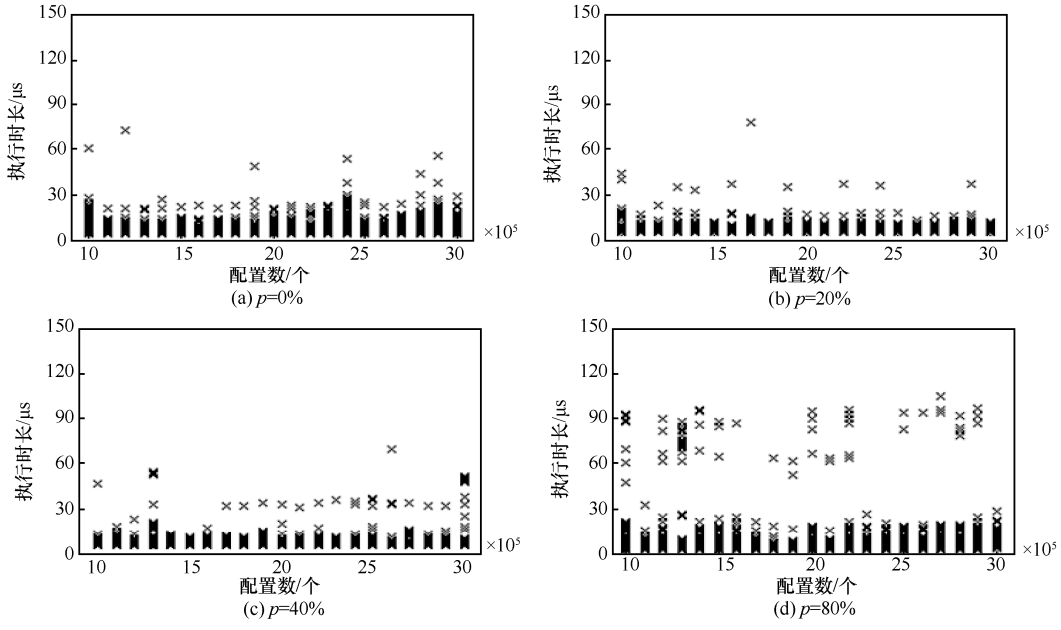

图 10　IMM4HT 执行时间随网页规模的变化趋势

全面检查 IMM4HT 的性能，对每种情况独立运行 10 000 次，以统计执行时间的分布情况。由图 10 可知，当网页的配置数小于 300 万的情况时，执行规则的时间一般在 20 μs 以下，且不随时间线性增长。IMM4HT 执行时间的分布随着 p 的增加而变得稀疏，p 越大，执行时间越长。但 IMM4HT 的平均执行时间基本不变。

5 结束语

本文提出了面向高速网络流量的恶意镜像网站识别方法，该方法首先接入网络流量，利用底层平台对流量进行 IP 碎片重组、TCP/UDP 流还原得到完整的分组段。然后对所需要的 HTTP 流量进行识别，得到完整的网页内容。接着对拼接的完整网页内容进行 Gumbo 标准化处理并分块，通过 simhash 计算出网页源码散列值。最后与基准网页进行比对，计算网页源码散列和网页正文散列与基准网页之间的海明距离。当其阈值在设定的范围内时，提取网页快照的 SIFT 特征点，通过聚类分析和映射处理得到网页快照的感知散列值。当被测网页与基准网页快照的感知散列值相似性大于一定阈值时，判定该网页是恶意镜像网页。实验表明该方法在准确率、召回率、处理时延这 3 项指标的综合评价最高。

由于恶意镜像网页的形式多种多样，一些恶意镜像网页与其基准网页在源码上存在较高的相似度，而网页快照则存在较大差异；另一些恶意镜像网页与其基准网页的网页快照存在较高相似度，而源码却截然不同。在下一步的研究与学习中，将针对这两方面对所提方法进行改进。

参考文献：

[1] CINIC. The 41st China statistical report on internet development [R]. Beijing: China Internet Network Information Center, 2018.

[2] QIN Z, YAN J, REN K, et al. SecSIFT: secure image SIFT feature extraction in cloud computing[J]. ACM Transactions on Multimedia Computing, Communications and Applications, 2016, 12(4s): 65.

[3] GOMEZ-NIETO E, SAN ROMAN F, PAGLIOSA P, et al. Similarity preserving snippet-based visualization of Web search results[J]. IEEE Transactions on Visualization and Computer Graphics, 2014, 20(3): 457-470.

[4] HOFMANN T. Probabilistic latent semantic indexing[C]//ACM SIGIR Forum. ACM, 2017: 211-218.

[5] SADOWSKI C, LEVIN G. Simhash: hash-based similarity detection[R]. Google, 2007.

[6] KOŁCZ A, CHOWDHURY A. Lexicon randomization for near-duplicate detection with I-match[J]. The Journal of Supercomputing, 2008, 45(3): 255-276.

[7] SHIVAKUMAR N, GARCIA-MOLINA H. Finding near-replicas of documents on the Web[C]//International Workshop on the World Wide Web and Databases. Springer , 1998: 204-212.

[8] KAPOOR A, ARORA V. Application of bloom filter for duplicate URL detection in a web crawler[C]// IEEE International Conference on Collaboration and Internet Computing. IEEE, 2016: 246-255.

[9] JIANG J, CHEN J, CHOO K K R, et al. A deep learning based online malicious URL and DNS detection scheme[C]//International Conference on Security and Privacy in Communication Systems. Springer, 2017: 438-448.

[10] LIU W, DENG X, HUANG G, et al. An antiphishing strategy based on visual similarity assessment[J]. IEEE Internet Computing, 2006, 10(2): 58-65.

[11] MAO J, TIAN W, LI P, et al. Phishing-alarm: robust and efficient phishing detection via page component similarity[J]. IEEE Access, 2017(5): 17020-17030.

[12] ZHANG H, LIU G, CHOW T W S, et al. Textual and visual content-based anti-phishing: a Bayesian approach[J]. IEEE Transactions on Neural Networks, 2011, 22(10): 1532-1546.

[13] CHEN Z, ZHANG P, ZHENG C, et al. CookieMiner: towards real-time reconstruction of web-downloading chains from network traces[C]// IEEE International Conference on Communications. IEEE, 2016: 1-6.

[14] BOBBARJUNG D R, JAGANNATHAN S, DUBNICKI C. Improving duplicate elimination in storage systems[J]. ACM Transactions on Storage, 2006, 2(4): 424-448.

[15] MIKOLAJCZYK K, SCHMID C. A performance evaluation of local descriptors[J]. IEEE transactions on pattern analysis and machine intelligence, 2005, 27(10): 1615-1630.

[作者简介]

张蕾（1996– ），女，四川广元人，中国科学院大学博士生，主要研究方向为信息过滤与内容计算及网络安全。

张鹏（1984– ），男，安徽淮南人，博士，中国科学院信息工程研究所研究员，主要研究方向为分布式系统和数据挖掘及网络安全。

孙伟（1980– ），男，山西宁武人，北京交通大学博士生，主要研究方向为计算机网络、信息安全和网络测量。

杨兴东（1994– ），男，河北张家口人，北京航空航天大学硕士生，主要研究方向为网络流数据处理及网络空间安全。

邢丽超（1993– ），男，黑龙江哈尔滨人，中国科学院大学硕士生，主要研究方向为信息过滤与内容计算。