doi:10.11959/j.issn.1000-436x.2015293

高效的基于段模式的恶意 URL 检测方法

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：提出一种高效的基于段模式的检测恶意 URL 的方法，该方法首先解析已标注的恶意 URL 中的域名、路径名和文件名 3 个语义段，然后通过建立以三元组为词项的倒排索引快速计算恶意 URL 每个语义段的模式，最后基于倒排索引查找到的段模式来判定给定的 URL 是否是恶意 URL。不仅如此，该方法还支持基于 Jaccard 的随机域名识别技术来判定包含随机域名的恶意 URL。实验结果表明，与当前先进的基准方法相比，该方法具有较好的性能和可扩展性。

关键词：恶意 URL；段模式；三元组；倒排索引；随机域名
中图分类号：TP319 **文献标识码**：A

Efficient segment pattern based method for malicious URL detection

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: An efficient segment based method for detecting malicious URL was proposed. Firstly it analyzed the annotated malicious URLs in terms of three semantic segments, i.e., domain segment, path segment and file segment. Secondly it quickly calculated the common pattern of each semantic segment exploiting the tri-gram model based inverted index. Finally it decided whether a given URL was malicious based on the segment patterns returned by searching the inverted index. Moreover, this method also supported the Jaccard based random domain name identification technique for deciding malicious URLs with random domain name. Experimental results show that proposed method outperforms the state-of-the-art baseline methods, and can achieve good efficiency and scalability on malicious URL detection.

Key words: malicious URL; segment pattern; tri-gram; inverted index; random name

1 引言

随着互联网技术的飞速发展，网络犯罪手段层出不穷，网络威胁形式越来越多，导致识别网络威胁的难度大大增加，成本呈上升趋势。2014 年，美国波莱蒙研究所（Ponemon institute）发布的报告[1]表明，2014 年网络攻击导致美国的大公司损失 1 270 万美元，同比增长 9.7%。该报告显示，网络犯罪给大公司造成的损失当中，业务中断和信息丢失占了近 75%；而一般企业或组织平均每周遭受到 1.7 次成功的网络攻击，平均修复一次网络攻击的周期为 31 天，所需要花费的代价约为 64 万美元，这与 2013 年相比，修复周期延长了 4 天，花费代价增长了 23%。尽管网络防御水平在不断提高，但是网络犯罪集团也在不断增强其适应力，因此需要研究有效的方法来识别网络威胁，预防网络犯罪的发生。

Wikipedia 将任何一种使用万维网方便网络犯

收稿日期：2015-10-25
基金项目：国家高技术研究发展计划（“863”计划）基金资助项目（Y370041101）；国家自然科学基金资助项目（61174152, 61303056, 61402464, 61502478）
Foundation Items: The National High Technology Research and Development Program of China (863 Program) (Y370041101); The National Natural Science Foundation of China (61174152, 61303056, 61402464, 61502478)

罪的威胁都称之为网络威胁[注1]。网络威胁使用不同类型的恶意软件和诈骗手段，它们的共同点是使用 HTTP 或 HTTPS 协议，或是使用其他类型的协议或组件访问 Web。因此，通过检测恶意 URL 来判定网络威胁（如钓鱼网站等）是可行的。然而，恶意 URL 为了减少被检测到的可能，可能会采用各种手段来隐藏自己。例如，Porras 等[2, 3]使用当前日期和时间作为种子每小时随机生成 250 到 50 000 个域名，包含这种随机域名的恶意 URL 难以被检测到。除此之外，与网站正确的 URL 具有较高相似度的恶意 URL 也很难被检测到，这种 URL 也很容易误导用户，例如将 login 篡改为 log1n 或将 index 篡改为 lndex 等，用户很可能误入这些网址导致信息泄露。所以，为了避免信息泄露，预防网络犯罪，这些恶意 URL 需要在被访问之前检测出来。为此，有效的检测恶意 URL 方法应满足如下要求。

1) 实时性。检测方法应能在短时间内检测出恶意 URL。在用户访问一个恶意 URL 时需要请求服务器，检测方法应能在恶意网页返回给用户之前提示用户该 URL 具有不良目的，并将恶意网页内容阻止在客户端外。

2) 扩展性。检测方法应能有效地检测出新的恶意 URL。攻击者为了躲避正常的检测，会使用算法来生成随机域名增加检测的难度，检测方法应该能够检测出那些新的、不常见的恶意 URL。

3) 准确性。检测方法应具有较高的精度。目前恶意 URL 的数量远不及网站正确的 URL 的数量，要能精确地检测出恶意 URL 具有一定的挑战；另外，有的恶意 URL 只篡改了网站正确的 URL 的一些关键词，给恶意 URL 和正确 URL 的区分带来了很大的挑战。

为此，本文提出一种高效的基于段模式的检测恶意 URL 的方法，该方法首先解析已标注的恶意 URL 中的域名、路径名和文件名 3 个语义段，然后通过建立以三元组为词项的倒排索引计算恶意 URL 段的公共模式（TCP, tri-gram inverted index based common pattern computing），因此，将该方法简记为 TCP 方法。TCP 方法直接从组成恶意 URL 的字符串中提取恶意 URL 的段模式。TCP 方法由于只使用 URL 的词汇特征，不需要额外特征，节省了计算开销。根据 URL 的标准规范[4]，URL 字符串只包含字母、数字和一些特定的符号，例如“/”、“?”、“.”、“=”、“-”、“_”等，所以恶意 URL 段模式的提取过程只是对字符串进行处理。在目前的条件下，URL 处理的速度可以达到每秒百万量级。不仅如此，TCP 基于倒排索引查找到的段模式，使用有限状态自动机[5]来判定给定的 URL 是否是恶意 URL，避免了不存在公共模式的 URL 对之间的计算，提高了恶意 URL 判定的效率。

2 相关工作

恶意 URL 检测方法根据使用的信息不同，大致可以分为 3 类：基于黑名单的方法、基于网页内容的方法和基于 URL 的方法。接下来，介绍这几类工作的典型代表。

基于黑名单的方法[6]主要是通过查找 URL 黑名单来判断给定的 URL 是否为恶意 URL，如果命中，则该 URL 为恶意 URL，否则为正确的 URL。如 Google Safe Browsing、Netcraft Toolbar、eBay Toolbar 等浏览器的黑名单机制都属于这类方法[7]。这种方法主要通过人工标记、蜜罐、用户反馈、爬虫等方法来维护 URL 黑名单。通过分析可以看出，基于黑名单的方法简单、直接、准确率高。然而，这种方法只能检测已出现过的恶意 URL，对于新的和包含随机域名的恶意 URL 无法检测出来。

考虑到恶意 URL 的网页内容具有某种特殊的目的或意义，因此另一种典型的检测恶意 URL 的方法是基于网页内容的方法，该方法借助网页包含的信息，如网页标签、文本等，判定给定的 URL 是否是恶意 URL。Provos 等[8]提出了一种利用网页标签特征检测恶意 URL 的方法，例如某些特定 JavaScript 是否出现，iframe 标签是否越界等。Moshchuk 等[9]提出一种利用反间谍软件工具来分析 URL 网页内容中是否包含木马可执行文件，以此来判定恶意 URL。

Zhang 等[10]通过计算网页中每个词语的 TF-IDF 值，从中选择 TF-IDF 值最高的几个词语组成查询，利用搜索引擎返回的检索结果，判定待检测网页的合法性。许杰[11]提出一种基于 TF-IDF 余弦定理算法对网页进行特征匹配的方法，对用户正在访问的恶意 URL 进行检测与拦截。通过对相关工作的分析可以看出，基于网页内容的方法首先需要获取网页的内容，然后对网页内容进行分析，这样会带来显著延迟，不适合高速的在线检测。

另一种比较流行的检测恶意 URL 的方法是基于

注1 https://en.wikipedia.org/wiki/Web_threat

URL的方法，目前已有的工作基本都是通过提取URL中的特征，例如 URL 的长度信息、服务器地理位置信息、服务器 IP 信息等，训练分类器对 URL 进行分类，从而判定给定的 URL 是否是恶意 URL。例如，Garera 等[12]通过分析钓鱼网站的 URL 结构，总结出 4 种类型的 URL 结构，通过特征选择算法，选取页面特征、域名特征、类型特征和词汇特征等 18 种特征，基于逻辑斯谛回归（logistic）模型训练分类器，对恶意 URL 进行检测和拦截。

Ma 等[13~15]则提出一种基于可疑 URL 的词汇特征和主机属性训练分类器，对恶意 URL 进行检测的方法，该方法利用词袋模型（BOW, bag of word）获得成千上万的特征。其中，词汇特征包括主机名长度、URL 长度、URL 中点号数等信息以及 URL 中主机和路径中每一个词汇符号信息；在主机属性中考虑 IP 地址属性、WHOIS 属性、域名属性和地理位置属性。通过分析可以看出，基于分类模型的恶意 URL 检测方法，需要对大量的特征进行提取和计算，而特征选取的质量直接决定方法的有效性。

与现有的方法相比，本文提出的 TCP 方法只使用基于 URL 字符串提取的段模式作为特征检测 URL 是否是恶意 URL，大大减少了特征计算量；而且，该方法引入了信息检索中的倒排索引技术，提高了 URL 模式的计算和匹配速度，进而提高 TCP 对恶意 URL 的检测速度；不仅如此，该方法还支持基于 Jaccard 的随机域名识别技术来判定包含随机域名的恶意 URL。

3　TCP 方法的原理

本节将详细介绍 TCP 方法的原理。为此，首先给出符号定义和恶意 URL 检测问题的形式化定义，然后介绍 TCP 方法的框架。

3.1　符号和问题定义

根据 URL 的标准规范[4]，URL 字符串包含 3 个不同的语义段：域名、路径名和文件名，因此可以将 URL 解析成这 3 个语义段的形式，然后逐个段考虑。为了简化运算，本文将 URL 中的字母、数字和特定的符号，例如“?”、“=”、“-”、“_”等，都当作常规字符对待，字符“/”作为段连接符和路径名分隔符，“.”作为域名和文件名分隔符，在提取语义段的公共模式时常规字符不区分考虑。下面则给出段的公共模式的定义。

定义 1　(段公共模式)。段公共模式（简称为段模式）是由常规字符组成的字符串，记为 $s=c_1,\cdots,c_\ell$，其中，ℓ 是公共模式的长度；$c_i\,(1\leqslant i\leqslant \ell)$ 是常规字符或是通配符“*”，通配符能匹配任意长度的常规字符串，但不包含“/”和“.”。对于任意的 $i\,(1\leqslant i<\ell)$，满足以下条件：若 $c_i=*$，则 $c_{i+1}\neq *$。规定只包含通配符的段模式是非法的。

基于段模式的定义，接下来将详细介绍段模式与语义段匹配的判定规则。

规则 1　已知段模式 $s=c_1,\cdots,c_\ell$，给定一个语义段 $u=c'_1,\cdots,c'_m$，如果存在一个函数 $f:[1,m]\to[1,\ell]$，满足 $\ell\leqslant m$，对于 $\forall j\in[1,m)$，都有 $f(j)\leqslant f(j+1)$；且对于 $\forall i\in[1,\ell)$，若 $c_{i+1}\neq *$，存在唯一的 $j\in[1,m)$，使 $f(j)=i, c'_j=c_i$，则称段模式 s 与语义段 u 匹配。

本文分别使用 s_d、s_p、s_f 表示域名、路径名、文件名的段模式，由于域名、路径名和文件名采用相同的常规字符集表示，因此，这 3 个语义段都可以基于规则 1 进行匹配判断。

由于 URL 可以解析为域名、路径名和文件名 3 个语义段的形式，因此 URL 公共模式可以通过这 3 个语义段的模式来表示，基于段模式的定义，下面给出 URL 公共模式的定义。

定义 2　(URL 公共模式)。URL 公共模式(简称为 URL 模式)是由对应的域名、路径名和文件名 3 个段的模式通过字段连接符连接而成，记为 $P=s_d/s_p/s_f$。

通过定义 2 可以看出，URL 模式匹配问题可以转化为段模式匹配的问题进行求解，接下来介绍 URL 公共模式和 URL 匹配的判定规则。

规则 2　已知 URL 模式 $P=s_d/s_p/s_f$，给定一个 URL，若 P 的每个段模式与都与待检测 URL 的对应段匹配，则称该 URL 模式与待检测 URL 匹配。

下面将详细论述 TCP 方法的处理流程。

3.2　TCP 方法框架

TCP 方法检测恶意 URL 的处理框架如图 1 所示。

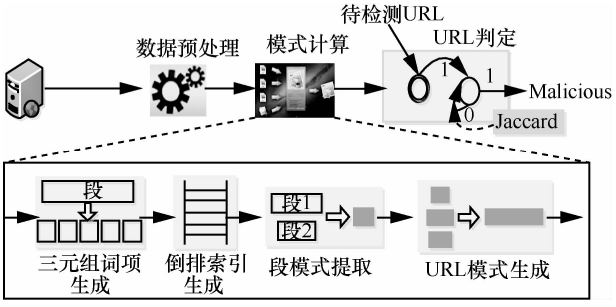


图 1　TCP 方法框架

该框架主要包括以下 3 个模块。

1) 数据预处理模块：该模块解析 URL 中的域名、路径名和文件名 3 个语义段。

2) 模式计算模块：该模块首先将每个语义段基于三元组模型（tri-gram）转化为词项集合表示，接下来基于词项建立倒排索引，然后通过建立以三元组为词项的倒排索引提取 URL 每个语义段的模式，最后基于段模式通过字段连接符连接得到URL 公共模式。

3) URL 判定模块：该模块基于恶意 URL 的公共模式判定被检测 URL 是否是恶意 URL。

下面则详细介绍这 3 个模块的处理流程。

3.2.1 数据预处理

考虑到 URL 采用的协议一般都是 HTTP 或 HTTPS，因此，在解析 URL 时不考虑 URL 中的协议部分，在解析之前先将 URL 包含的“http(s)：//”部分从 URL 中分离；然后根据 URL 字符串的特点，解析 URL 的剩余部分，从中提取域名、路径名和文件名 3 个语义段，以如下 URL 为例。

URL="walmartmegablackout.com/include/wordpress/login.htm"，对该 URL 进行解析，可以从中解析出：域名="walmartmegablackout.com"、路径名="include/wordpress"、文件名="login.htm"。

3.2.2 模式计算

本节将详细介绍如何基于数据预处理模块获取的 URL 的 3 个语义段，生成 URL 的公共模式。URL 公共模式计算主要分为以下几个步骤。

1) 语义段三元组词项表示。TCP 基于自然语言处理中定义的三元组模型（tri-gram）将语义段表示成三元组集合的形式。由于 URL 中的域名、路径名和文件名采用相同的常规字符集合表示，因此这 3 个段的三元组词项表示可以采用相同的方式生成。本节以域名段的词项表示为例，将域名段表示为多个三元组的算法如算法 1 所示。

在该算法中，*Count* 表示域名段中包含的三元组个数；*TrigramArray* 是一个用于存储域名段中三元组及其位置数组。算法首先根据域名分隔符“.”将域名段分解成子串的形式，然后将每个子串分别表示成三元组，这样保证三元组中的字符来自同级域名。对于文件名，它的三元组表示方法与域名一致，而路径名由于以“/”将不同级路径分开，因此，在对路径名进行三元组词项表示时，需将域名表示算法中的域名分隔符“.”替换成路径分隔符“/”。

值得注意的是，本文之所以采用 tri-gram 模型表示语义段，原因在于：据统计任意 2 个 URL 串之间具有相同二元组、三元组和四元组的概率分别为 95.7%、75.8%和 33.6%[16]。因此，为了降低模式计算的时间复杂度和词项倒排索引存储的空间复杂度、保证语义段切分的区分性和模式提取的合理性，本文采用 tri-gram 模型。在本文中，TCP 规定 2 个 URL 之间至少有一个相同的 tri-gram 时，才计算它们之间的公共模式。

算法 1 SplitDomainIntoTrigrams

Input: *Domain*, *Count*

Output: an array of tri-grams of *Domain*

1) Set *TrigramArray*← ø

2) *CurrentPos* = 0

3) Split *Domain* into *SubStrings*

4) while *SubString* != ø do

5) if strlen(*SubString*) ≥ 3 then

6) Split *SubString* into successive *Trigrams*

7) *TrigramArray*←*Trigram* and *Position*

8) else

9) *SubString* regard as *Trigram*

10) *TrigramArray*←*Trigram* and *Position*

11) end if

12) *Count*++

13) end while

14) return *TrigramArray* and write back *Count*

2) 倒排索引创建。TCP 根据 URL 语义段的三元组词项表示，基于三元组为每个语义段创建倒排索引。以域名段为例，创建的域名段的倒排索引（记为 DomainInvertedIndex）如图 2 所示。

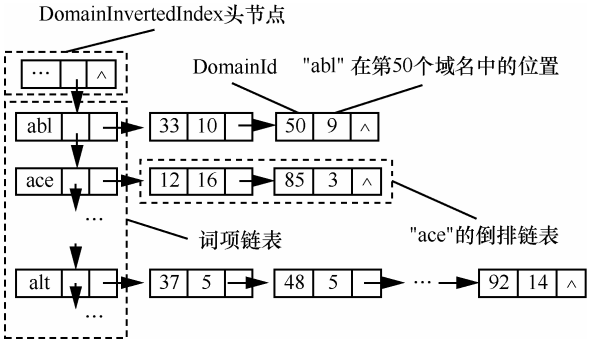


图 2 TCP 方法域名段倒排索引数据结构

在索引中，词项链表中的每个节点包含 3 个字段：tri-gram、指向下一个 tri-gram 的指针和指向包含该 tri-gram 的倒排列表的指针。每个词项对应的

倒排链表的每个节点包含该 tri-gram 所属的域名的编号、在域名中的位置和指向下一个包含该 tri-gram 的域名节点的指针。其中，倒排链表按域名编号递增的方式存储。与此相同，可以建立路径名和文件名 2 个语义段的倒排索引。

3) 段模式提取。以域名段的段模式生成为例，为了生成域名段的段模式，TCP 需要遍历域名的倒排索引，查找可能存在公共模式的域名。具体地，对出现在同一倒排链表中的所有词项进行比较，计算它们之间的公共模式，通过遍历 2 个域名段对应的词项列表提取 2 个域名段的公共模式，具体如算法 2 所示。

算法 2　ExtractDomainCommonPattern

Input: $DomainId_1$，$DomainId_2$

Output: the common pattern of $Domain_1$ and $Domain_2$

1) Set *CommStr*←ø

2) Traverse TrigramArrays of $Domain_1$ and $Domain_2$

3) if $TrigramArray_1[i]$ =T$rigramArray_2[j]$ then

4)　if *CommStr* = ø then

5)　*Judge*($Position_{1i}$, $Position_{2j}$)

6)　*CommStr* = *Trigram* | ".*"+*Trigram* | "*"+*Trigram*

7)　else

8)　*Judge*($Position_{1i}$, $Position_{2j}$)

9)　Append(*CommStr*, the last letter of *Trigram* | "*."+*Trigram* | "*"+*Trigram* | *Trigram*)

10)　end if

11)　$Position_1 = Position_{1i}$; $Position_2 = Position_{2j}$;

12) end if

13) if any position is not the last Trigram of two Domain then

14)　*CommStr* = *CommStr* +"*"

15) end if

16) return *CommStr*

在算法 2 中，对 2 个域名段，本文只提取它们之间的一个公共模式。据实验统计，2 个域名存在多于一个段模式的概率不超过 2%[16]，因此 2 个域名之间只保留一个段模式是可行的。

段模式的提取是根据 2 个域名包含的 tri-gram 在域名中的位置进行计算，根据 tri-gram 在域名中出现的位置不同来判定是否将 tri-gram 或 tri-gram 中的字符或通配符写入段的公共模式。例如，域名 "walmartmegablackout.com" 与 "adamant-cable.ru" 提取的段模式为"*abl*"。同样，按照该方法生成路径名和文件名 2 个语义段的段模式。

4) URL 公共模式生成。基于 URL 中的域名、路径名和文件名 3 个段的段模式，使用段连接符"/"将 3 个语义段的模式进行拼接，根据定义 2 生成 URL 的公共模式。例如，给定 2 个 URL，"walmartmegablackout.com/include/wordpress/login.htm" 和 "adamant-cable.ru/include/world/index.html"，它们的公共模式为"*abl*/include/wor*/*.htm*"。

将生成的所有的 URL 模式加载到 0-1 状态有限自动机中，下面将详细介绍如何利用该自动机判断待检测的 URL 是否是恶意 URL。

3.2.3　恶意 URL 判定

恶意 URL 判定本质上是一个分类问题：给定一个待检测 URL，需要判断其是属于恶意 URL，还是属于网站正确的 URL。因此，恶意 URL 判定可以通过一个简单的线性分类器来实现[17]。本文使用一个有限状态自动机来实现恶意 URL 的判定。

具体地，TCP 准备一个标注的 URL 训练数据集，该数据集中恶意 URL 和正确的 URL 按 1:2 的规模组成。通过数据预处理和模式计算 2 个模块对该数据集中的所有 URL 进行 URL 模式提取，然后将所有的 URL 模式加载到自动机中，从而完成恶意 URL 判定的准备工作。

接下来，基于 3.1 节中定义的规则 1 和规则 2 将待检测的 URL 与自动机中所有的 URL 模式进行匹配，记自动机中恶意 URL 模式与该待检测 URL 匹配的个数为 M_{num}，正确 URL 模式与该 URL 匹配的个数为 N_{num}，根据标注的训练数据的特点，若满足条件：$2M_{\text{num}} \geqslant N_{\text{num}}$，则判定该 URL 为恶意 URL，否则为正确的 URL。

值得注意的是，在本节介绍的都是包含固定域名的恶意 URL 检测方法，接下来将介绍包含随机域名的恶意 URL 判定方法。

3.3　包含随机域名的恶意 URL

目前，有很多恶意 URL 通过随机生成域名的方式来躲避检测与拦截[2, 3]。因此，为了进一步提高恶意 URL 检测的准确率，TCP 在 URL 判定中引入随机域名识别机制。

在随机域名识别方面，Yadav 等[18]提出了一种

通过计算域名的一元组和二元组的 KL 距离、Jaccard 系数和编辑距离判定产生的随机域名是否是恶意的方法，并且通过实验说明基于 Jaccard 系数的方式判定效果最好。因此，TCP 引入 Jaccard 系数来处理包含随机域名的恶意 URL 的判定。

鉴于在计算 URL 公共模式时使用 tri-gram 作为词项，对于随机域名的表示继续使用 tri-gram 表示。TCP 针对无法用有限自动机判定的 URL，则通过计算恶意（正确）URL 的域名段模式与该 URL 的域名段的 Jaccard 系数进行判定，计算方式如下。

$$sim = \frac{|A \cap B|}{|A \cup B|}$$

其中，A 表示被检测 URL 的域名段的 tri-gram 集合；B 表示由 TCP 中所有恶意（正确）URL 的域名段模式中的 tri-gram 组成的集合；$A \cap B$ 表示 A 与 B 之间相同的 tri-gram 集合；$A \cup B$ 表示 A 与 B 包含的所有的 tri-gram 集合；$|\cdot|$ 表示集合的大小。sim 表示集合 A 与 B 的 Jaccard 相似度。

记恶意 URL 的域名段模式与该 URL 的域名段的 Jaccard 相似度为 JM_{sim}，正确 URL 的域名段模式与该 URL 的域名段的 Jaccard 相似度为 JN_{sim}，若满足：$2JM_{sim} \geqslant JN_{sim}$，则判定该 URL 为恶意 URL，否则为正确的 URL。

4 实验与分析

为了验证本文提出的基于段模式的恶意 URL 检测方法（TCP）的有效性，本节将对 TCP 的有效性和扩展能力进行实验分析。首先，测试 TCP 方法检测恶意 URL 的准确率；然后，测试 TCP 方法的运行效率；最后测试 TCP 方法的扩展能力。本节所有实验都是在一台服务器上完成的，配置如下：64 bit Linux OS，16 core 2 GHz AMD Opteron(tm) 6128 处理器，32 GB RAM。

4.1 实验设置

1) 数据集：实验中使用的恶意 URL 数据和正确的 URL 数据都来自网上的公开数据集，是通过使用开源软件 Larbin [注2]从网站上抓取、去重获得的。其中，恶意 URL 数据集由从 2 个著名的恶意 URL 汇总网站 Phish Tank [注3]和 Malware Patrol [注4]网站上爬取的；正确的 URL 数据集是从 Google 和 DMOZ 网站上爬取的。数据集的分布情况如表 1 所示。

表 1 数据集组成情况

URL 类型	训练数据规模/万	测试数据规模/万
恶意 URL	400	80
正确 URL	800	120
共 计	1 200	200

2) 基准方法：为了验证 TCP 方法对恶意 URL 检测的有效性，在实验中采用以下 2 种典型的方法作为基准方法。

黑名单方法：经典的检测恶意 URL 的方法[6]，主要是通过查找 URL 黑名单来判断给定的 URL 是否为恶意 URL。

CW 方法：CW 方法[14]是一种在线学习的方法，它基于置信度加权（CW, confidence weighted）算法判断恶意 URL。

基于上述实验设置，首先测试各个方法在恶意 URL 检测上的准确率，然后进一步测试各个方法的运行效率，最后测试 TCP 方法的扩展能力。

4.2 实验结果

4.2.1 准确率测试

为了测试 TCP 方法检测恶意 URL 的准确率，本节分别比较 TCP 方法和 CW 方法对测试数据数据中恶意 URL 检测的误判数和漏判数。其中，误判数是指将 URL 恶意性判断错误的数量，漏判数是指方法没有判断出来的恶意 URL 的数量。

误判数和漏判数的实验结果如图 3 和图 4 所示。

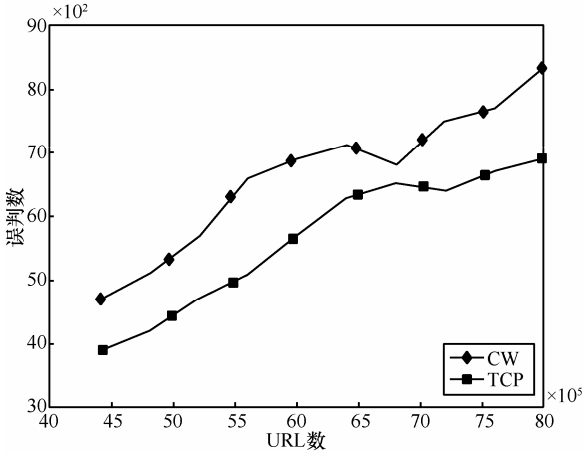


图 3 TCP 与 CW 误判数比较

注2 http://larbin.sourceforge.net/index-eng.htm

注3 http://www.phishtank.com/

注4 http://www.malware.com.br/

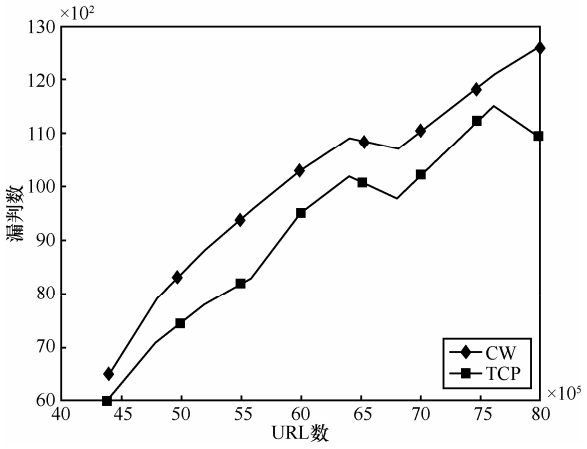


图 4　TCP 与 CW 漏判数比较

从图 3 和图 4 中可以看出，与基准方法 CW 相比，TCP 方法的漏判数和误判数都明显低于 CW，这说明 TCP 方法能够有效检测恶意 URL。

4.2.2　运行效率测试

本节评估 TCP 方法与基准方法 CW 在恶意 URL 检测上的运行效率。

在实验中，通过比较 TCP 和 CW 在相同数据集下检测恶意 URL 的时间开销，来评价这些方法的运行效率，实验结果如图 5 所示。

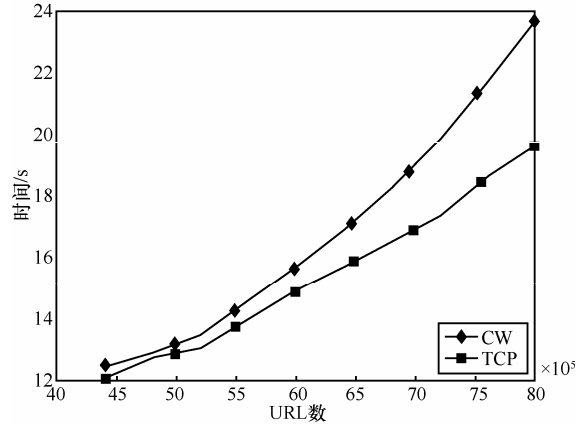


图 5　TCP 与 CW 运行效率比较

从图 5 中可以看出，与 CW 方法相比，TCP 方法的运行效率明显优于 CW 方法，且增长速度也小于 CW，这说明 TCP 方法在恶意 URL 判定的实时性好，原因在于，TCP 采用倒排索引避免了冗余的计算，并按 tri-gram 将 URL 模式进行排列，在检测 URL 时，根据倒排索引查找可能匹配的 URL 模式，减少了模式匹配的计算量。

4.2.3　扩展能力测试

本节将验证 TCP 方法的扩展能力。

在实验中，通过对比在检测相同数量的恶意 URL 时，TCP 方法所需的 URL 公共模式数量和黑名单方法所需的 URL 数量来评价这些方法的扩展能力，实验结果如图 6 所示。

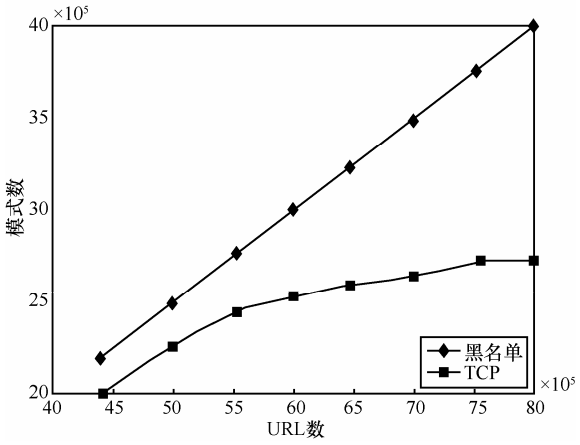


图 6　TCP 与黑名单方法扩展能力比较

从图 6 中可以看出，在检测相同规模的 URL 时，TCP 方法使用的 URL 模式数要远远小于黑名单方法使用的 URL 数。不仅如此，随着检测的 URL 数量的增加，TCP 方法所需的 URL 公共模式的数量近似以对数的速度增长，而黑名单方法所需的 URL 数量则呈现线性增长。通过该组实验可以说明，与黑名单方法相比，TCP 方法具有良好的扩展能力，这是因为 TCP 方法采用段模式的思想：一个 URL 公共模式能够有效匹配多个 URL，因此，在一定程度上，TCP 可以用少量的 URL 模式即可检测更多的恶意 URL。

基于以上实验分析可以看出，与基准测试相比，TCP 方法在检测恶意 URL 时，不仅可以获得更高的准确率，而且在实时性和扩展能力方面也能获得更好的效果，这些都表明 TCP 方法的有效性，这也说明在恶意 URL 检测中，采用段模式是一个非常有用的技术。

5　结束语

本文提出一种高效的基于段模式的恶意 URL 检测方法，通过建立以三元组 tri-gram 为词项的倒排索引快速计算恶意 URL 的段模式。不仅如此，该方法通过基于 Jaccard 的随机域名识别技术来判定随机域名产生的恶意 URL。通过与最新的 CW 方法和黑名单方法的大量的实验表明，该方法在检测恶意 URL 时具有较好的实时性、扩展性和有效性。然而，TCP

还存在一些问题。例如，只通过简单的 Jaccard 指数检测包含随机域名的恶意 URL 的方式还不够完善。因此，在下一步工作中将根据域名与 IP 地址之间的映射关系，检测包含随机域名的恶意 URL。

参考文献：

[1] Ponenom Institute. 2014 Global Report on the Cost of Cyber Crime[R]. 2014.

[2] PORRAS P, SAIDI H. YEGNESWARAN V. Conficker C P2P Protocol and Implementation[R]. SRI International Tech. Rep. 2009.

[3] PORRAS P, SAIDI H, YEGNESWARAN V.　An Analysis of Conficker's Logic and Rendezvous Points[R]. SRI International Tech. Rep, 2009.

[4] https://url.spec.whatwg.org/[EB/OL]. 2015.

[5] HENZINGE T A, RASKIN J C C O. The equivalence problem for finite automata: technical perspective[J]. Communications of the ACM, 2015, 58(2): 86-86.

[6] PRAKASH P, KUMAR M, KOMPELLA R R, *et al*. Phishnet: predictive blacklisting to detect phishing attacks[A]. Proceedings of IEEE International Conference on Computer Communications[C]. 2010. 1-5.

[7] LIKARISH P, JUNG E. Leveraging Google safe browsing to characterize Web-based attacks[A]. Association for Computing Machinery[C]. 2009.

[8] PROVOS N, MAVROMMATIC P, RAJAB M A, *et al*. All your iframes point to us[A]. Proceedings of the 17th Usenix Security Symposium[C]. 2008.1-16.

[9] MOSHCHUK A, BRAGIN T, GRIBBLE S D, *et al*. A crawler-based study of spyware in the Web[A]. Proceedings of the Network and Distributed System Security Symposium[C]. 2006.

[10] ZHANG Y, HONG J, CRANOR L. Cantina: a content-based approach to detecting phishing Web sites[A]. Proceedings of 16th International Conference on World Wide Web[C]. 2007. 639-648.

[11] 许杰. 云安全模式下恶意 URL 实时检测系统的设计与测试[D]. 北京: 北京邮电大学, 2014.
XU J. Design and Testing of Malicious URL Real-time Detecting System Working in the Mode of Cloud Security[D]. Beijing University of Posts and Telecommunications, 2014.

[12] GARERA S, PROVOS N, CHEW M. A framework for detection and measurement of phishing attacks[A]. Proceedings of 5th ACM Workshop on Recurring Malcode[C]. 2007. 1-8.

[13] MA J, SAUL L K, SAVAGE S, *et al*. Beyond blacklists: learning to detect malicious Web sites from suspicious URLs[A]. Proceedings of the 15th ACM SIGKDD Conference on Knowledge Discovery and Data Mining[C]. 2009.1245-1254.

[14] MA J, SAUL L K, SAVAGE S, *et al*. Identifying suspicious URLs: an application of large-scale online learning[A]. Proceedings of the 26th International Conference on Machine Learning[C]. 2009. 681-688.

[15] THOMAS K, GRIER C, MA J, *et al*. Design and evaluation of a real-time url spam filtering service[A]. Proceedings of the 2011 IEEE Symposium on Security and Privacy[C]. 2011. 447-462.

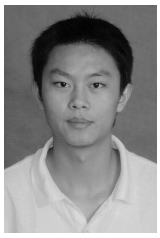
[16] HUANG D, XU K, PEI J. Malicious URL detection by dynamically mining patterns without pre-defined elements[J]. World Wide Web, 2014, 17(6): 1375-1394.

[17] HAN J W, KAMBER M, PEI J. Data Mining: Concepts and Techniques[M]. Beijing: China Machine Press.2012.

[18] YADAV S, REDDY A K, RANJAN S. Detecting algorithmically generated malicious domain names[A]. Proceedings of the 10th ACM SIGCOMM Conference on Internet Measurement[C]. 2010. 48-61.

作者简介：

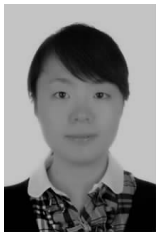
林海伦（1987-），女，山东临沂人，博士，中国科学院信息工程研究所助理研究员，主要研究方向为数据挖掘、知识图谱。



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。