CONCURRENCY AND COMPUTATION: PRACTICE AND EXPERIENCE
Concurrency Computat.: Pract. Exper. 2017; **29**:e3891
Published online 30 June 2016 in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/cpe.3891

SPECIAL ISSUE PAPER

A novel attack to track users based on the behavior patterns

Xiaodan Gu[1,*,†], Ming Yang[1], Congcong Shi[2], Zhen Ling[1] and Junzhou Luo[1]

[1]*School of Computer Science and Engineering, Southeast University, Nanjing, P.R. China*
[2]*Global Energy Interconnection Research Institute*

SUMMARY

Currently, people around the world daily use the Internet to access various services, such as e-mail and online shopping. However, the behavior-based tracking attacks have posed a considerable threat to users' privacy. Relying on characteristic patterns within the Internet activities, this attack can link a user's multiple sessions. In this paper, we investigate the behavior-based tracking attack and propose some countermeasures to mitigate the threat. We preprocess the raw traffic data and then extract features ranging from lower layer network packets to high-level application-related traffic. Specifically, we focus on four types of application-level traffic to infer users' habits, including HTTP, IM, e-mail, and P2P. In addition, we extract the web queries entered into shopping websites and classify them to infer users' preferences. Then, we construct the preference models and propose an improved method. For evaluation, we collect traffic in the real-world environment to construct a large-scale dataset. Five hundred and nine users are selected in terms of the user's active degree. When the term frequency–inverse document frequency transformation is used, the improved method can identify an average of 93.79% instances correctly. Our extensive empirical experiments demonstrate the effectiveness and efficiency of our approaches. Finally, we discuss and evaluate several countermeasures. Copyright © 2016 John Wiley & Sons, Ltd.

Received 19 February 2016; Revised 15 April 2016; Accepted 15 May 2016

KEY WORDS: behavior-based tracking; privacy; traffic analysis; pattern recognition; preference model

1. INTRODUCTION

Benefiting from the dramatic improvement of data transmission technologies, the World Wide Web is ubiquitous. More and more websites are developed to meet the growing needs of people. One person can even live his life only relying on various categories of websites, including shopping, food ordering, socializing, entertainment, financing, and so on. However, the usages of these websites may disclose the user's privacy. For example, when a user shops online, the contact information, shipping address, and credit card number are all disclosed to the website. And the rising popularity of social networks makes users disclose many private information spontaneously, which can be used to portray life status [1] or even predict future activities [2].

In the aspect of the content providers, it is no longer news that they track their users with cookies. When a user visits some website, the provider may mark him with a unique ID without any consent. In the next time, through identifying the ID, the provider links multiple sessions of the same user and creates the activities profile, which may be sold to the advertisers through real-time bidding [3]. The researchers even have demonstrated that the average price of a user's profile is less than $0.0005.

Besides, the third-party web tracking [4] technologies make the situation worse. On the one hand, many users are unaware of the presences of third-party websites, so they have no consciousness to

*Correspondence to: Xiaodan Gu, School of Computer Science and Engineering Southeast University, Nanjing, P.R. China.
†E-mail: guxiaodan@seu.edu.cn

take countermeasures. On the other hand, by using super cookies and device fingerprinting methods, the third-party service providers can correlate users' activities across websites.

Furthermore, recent work [5–9] finds that the user can be tracked according to his web behavior pattern and propose the so-called 'behavior-based tracking' techniques. They believe that the websites visited by an individual reflect his interests and habits to some degree; this behavior may be repeated regularly in a period of time. On account of the users' personalized interests and habits, web behavior is a stable and discriminational feature to classify different users. Compared with the first two tracking methods, the behavior-based tracking attack poses a more serious threat for two main reasons. First, the attacker just needs to passively sniff the traffic generated by the victim instead of interacting with him, like storing information in his computer or running some scripts on his browser. As a result, the victim has no idea that he is tracked. Second, it is more difficult to resist the behavior-based tracking attack. With regard to the attacks based on cookies, the victim can disable or clear cookies frequently in the browser. For super cookies, he needs to search folders and delete them manually. Disabling Flash and JavaScript is a good way to reduce the victim's fingerprintable surface. To protect against the behavior-based tracking attack, he needs not only encrypt the content but also hide the communication relationship. Using some anonymous tools like Tor may be a convenient solution. However, Tor itself faces many security threats [10].

When researchers introduce the behavior-based tracking techniques, they demonstrate the effectiveness by performing experiments on the real-world traffic datasets. However, the scale of these datasets is always too small to prove the scalability, which is criticized by some people.

Our contribution. In this paper, we propose an improved behavior-based tracking attack by introducing the users' preference models, in which we extract web queries entered into the online shopping websites and classify them to infer the users' habits. In addition, to get the users' real behavioral intentions, we filter the HTTP traffic and pick out all domains that the users really want to visit in the data preprocessing. We carry out the experiments on the real-world dataset, which contains 509 active users; the improved method can identify an average of 93.79% instances correctly. The results show that the behavior-based tracking attack is still feasible in the large-scale scenario. We also design a browser extension to add noise, following some probability distribution to web traffic, which can effectively mitigate the threat.

The rest of this paper is organized as follows. Section 2 presents the related work. In Section 3, we describe the threat model and define some important concepts. In Section 4, we propose a basic behavior-based tracking method. We improve the basic method in Section 5. We provide the results of our experiments in Section 6. We also discuss and evaluate the countermeasures in Section 7. Section 8 concludes this paper and discusses the future work.

2. RELATED WORK

In the biometric field, the behavioral biometric is a mature technique based on users' skills, styles, preferences, knowledge, or strategies. For example, researchers propose varieties of methods to identify users according to some subtle differences in patterns of mouse movements [11, 12] or keystroke dynamics [13].

In the network application field, Padmanabha *et al*. [14] find that humans have unique clickprints when they browse the same website. They extract the duration, number of pages viewed, average time spent per page, the starting time, and the starting day of the week per session to construct the clickprint pattern from the real web browsing data. On this basis, Yang [15] proposes two profiling techniques and three additional criteria based on the concepts of support and lift. When profiles are built, she evaluates the similarity by using the Euclidean distance. Besides, researchers believe that the e-mail data also can be used to identify users. Vel *et al*. [16] use 170 style marker attributes and 21 structural attributes in the classification, including the greeting acknowledgment, farewell acknowledgment, signature text, number of attachments, and so on. To solve the problem of high-dimensional features, Lackner *et al*. [17] apply the concept of activation patterns to the e-mail header data, which is based on the artificial intelligence and machine learning techniques.

DOI: 10.1002/cpe

Unlike the scenarios mentioned in the preceding texts, in the traffic analysis field, the behavior-based tracking techniques are carried out by passively sniffing traffic and extracting behavioral features to link multiple sessions of the same user. Kumpost *et al.* [5] believe that the websites visited by the user and the corresponding frequencies reflect his habits. They store the destination IP address, source IP address, and the number of connections in a two-dimensional matrix based on the NetFlow logs. It means that the cell (i, j) contains the number of connections initiated from the source IP address i to the destination IP address j. Then, they employ the inverse document frequency (IDF) transformation and the cosine similarity metric to get better results. The accuracies for SSH, HTTP, and HTTPS are 61.512, 23.571, and 26.561%, respectively. Similarly, Herrmann *et al.* [6] apply the multinomial naive Bayes classifier to the destination host access frequencies. They evaluate their method on a real-world dataset from 28 users, and the accuracy is up to 73%. To further study the scalability in a real-world setting, they [7] implement an evaluation with the MapReduce framework on a large-scale dataset which contains more than 2100 concurrent users' DNS requests. By resolving ambiguous predictions with cosine similarity, 88.2% of all instances are linked correctly. They [8] also evaluate two classification techniques and the pattern mining approach based on the criteria support and lift. Abt *et al.* [9] propose a new approach to generate highly predictive user profiles based on behavior templates derived from statistics of IP addresses and port numbers. Their results show that it is enough to identify eight individuals by monitoring only 5 min of network traffic.

As we can observe, most works of behavior-based tracking techniques perform experiments on the small scale datasets, which cannot prove the scalability. Although Herrmann *et al.* get a high detection rate on a large-scale dataset, we think that it is not reasonable to discard test instances during resolving the ambiguous predictions. Besides, the accuracy can be improved further. To tackle these problems, we propose our behavior-based tracking methods and carry out experiments on a larger dataset in the real-world environment.

3. THREAT MODEL

In this section, we describe how the behavior-based tracking attack works as well as the assumptions of the threat model in detail. In addition, some important concepts are defined to make the description of our tracking methodology more clear.

3.1. Behavior-based tracking attack

As the name suggests, the behavior-based tracking attack is a tracking method to re-identify the target user according to his web behavior pattern. Different from the explicit identifier-based tracking technologies, like the cookie, account ID, and other unique identifiers, the behavior-based tracking attack needs to extract implicit identifiers from the victim's traffic and profile his behavior pattern. In the re-identification stage, the attacker utilizes the similarity algorithms to match the behavior patterns.

Figure 1 illustrates the attack scenario of the behavior-based tracking method we use throughout this paper. In it, the attacker implements the behavior-based tracking solely relying on sniffing the network traffic. We assume that an attacker can passively observe and record a set of users' traffic without knowing their real identities. For example, the attacker may be the internet service provider or the administrator of local area networks, so that he can capture all the traffic through the egress routers. Considering the high requirements for the capability, the attacker also can set up a free virtual private network or hotspot to attract traffic as an alternative. However, he cannot modulate the users' traffic or send probe packets.

If one user does not establish encrypted connections to transfer data, the attacker can obtain the contents of the application layer. In fact, it is a fairly common situation. While browsing websites, most people have no consciousness to use the anonymous tools to encrypt their data; the majority of websites still use the HTTP protocol to interact with the user, except during the authentication stage. So in our model, the attacker can get the application data if they are in plaintext. But the explicit

DOI: 10.1002/cpe

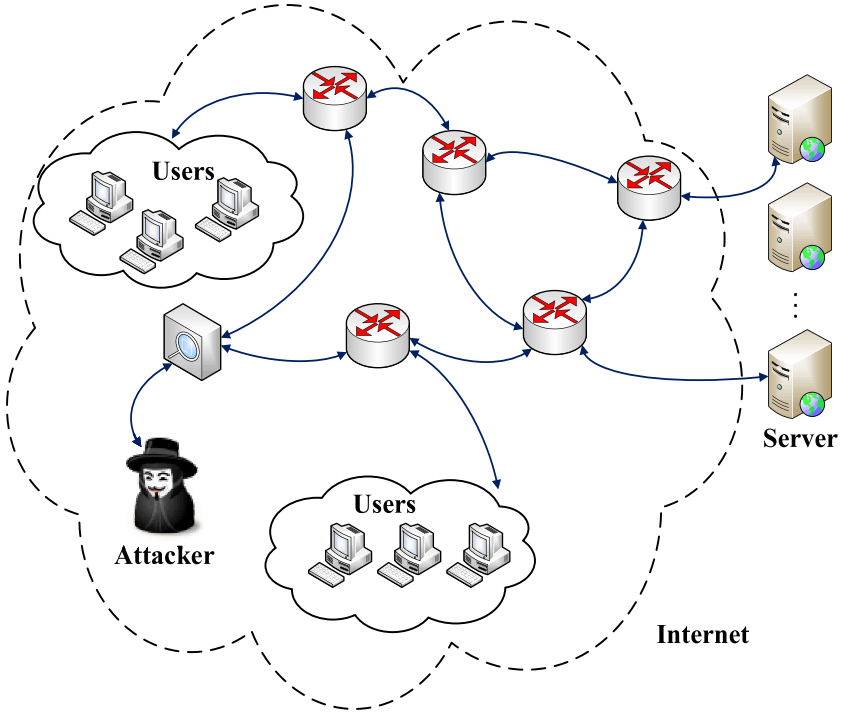

Figure 1. Threat model of the behavior-based tracking attack.

identifiers are not under consideration, that is, the QQ numbers, usernames of mailboxes, and social network websites.

3.2. Real user behavior

Related works [5–8] believe that the websites visited by a user can express his web behavior and habits. They always extract the destination IP address (or domain name) and number of connections from traffic to create profiles. The number of connections means the access frequency of the corresponding website. But nowadays, most websites display multiple advertisements (Ads) that belong to other domains in their pages; one Ad may appear in many different websites. When a user visits some websites, his or her browser establishes multiple TCP connections to request all objects embed. During the HTML parsing, the browser does not distinguish the Ad elements from other objects. Therefore, the generated traffic contains the Ad domains, which are not in line with the user's subjectively determined demand. These Ad domains are obfuscated data, which may decrease the accuracy of the classification algorithm. Consider the following case scenario. *UserA* and *UserB* visit websites in the light of lists L and M, respectively. There is no intersection between the lists L and M. but most of the Ads displayed by L and M are the same. In common sense, the behavior patterns of *UserA* and *UserB* are completely different. Unfortunately, if we extract all domains according to the existing methods, including the Ad domains, there is a possibility that the classifier identifies *UserA* and *UserB* as the same class. Moreover, the browser creates multiple TCP connections to the server to accelerate the data transmission in an interaction; the specific number of concurrent connections is decided by the web server and parameter settings of the user's browser. It is not equal to the access frequency. If we extract all the destination domains and corresponding connections as features, we will make the profiles ambiguous and get lower accuracy in classification.

As a result, we need to filter the victim's traffic and extract the real behavioral intentions. For instance, a user visits *Sina.com* one time and generates a series of packets, which contain the *Sina* and other domains, like *JD* and *Taobao*. But we just represent the traffic as a triplet <*Sina*, 1>, consisting of the real destination domain *Sina* and the corresponding access frequency. In order to

DOI: 10.1002/cpe

distinguish the real destination from other Ads, we use some method to preprocess the dataset before feature extraction. In addition, a few software and browser extensions automatically contact the servers via the HTTP protocol. We also need to filter traffic of this type.

3.3. Session

Similar to the related work [5–8], we assume that the IP address of each user is not changed in a fixed period of time; this period is called session in this paper. So all traffic with the same IP address in a session can be aggregated as one record. Obviously, if the session is set too small, we cannot classify the user's identity with high accuracy in a large scale. In the remainder of this paper, we set the session to 24 h by convention, beginning at 0:00 AM each day. The attacker can extract features from one session's traffic to create an instance.

3.4. Access frequency

We believe that the access frequency of a specific domain is a relatively ambiguous concept. As described in the preceding texts, the number of established connections during the page loading is obviously unreasonable. If we use the page request frequency as an alternative, the value will be vulnerable to the network jitter. The longer the page loading time is, the more frequently the user refreshes. In addition, the widely used AJAX frameworks may affect the frequency calculator. Hence, we regard the access frequency of a specific domain in a fixed-length time slot as 1. That is to say, we will only add 1 to the frequency even though the user loads the page repeatedly in one time slot. When the interval between two visits exceeds one time slot, we add 1 to the frequency.

4. THE BASIC METHOD

In this section, we describe our basic tracking methodology. In it, an observer can execute traffic analysis attacks to profile users' behavior and link multiple sessions originating from the same user on the basis of characteristic patterns. We first implement the data preprocessing to filter the traffic. Then, we extract appropriate features to create profiles. Finally, we convert the session linkage problem into pattern matching with the multinomial naive Bayes (MNB) classifier [18].

4.1. Data preprocessing

Because of the high revenue, many websites embed Ads in their pages. Even a few websites maintain operations relying on advertising. When a user visits some websites, his traffic may contain many Ads, which are not the real behavioral intentions. It is very common that the operating system components and some software, like the antivirus and browser extensions, automatically contact the servers via the HTTP protocol at high frequencies. Therefore, we decide to preprocess the traffic to get the user's real behavior.

In the data preprocessing, we first need to pick out all domains that the users really want to visit. As shown in Figure 2, according to the HTTP protocol [19], just the first GET request retrieves the target HTML document when a user opens a new page in his browser. All the subsequent requests refer to the objects embedded in this page. They may contain some resources in other domains, for example, the Ad images and videos. Therefore, we can pick out all GET requests for the HTML documents to represent users' accesses. Unfortunately, no explicit identifier can indicate the appearance of a HTML document request. To address this problem, we try to analyze the media types of resource based on the URL strings. But the result is frustrating. When we match the URL strings with some keywords, for example, 'html' and 'jsp', we discard many useful information. We find that most websites apply the URL-rewriting techniques to improve the usability and search friendliness, which remove the type extensions of HTML documents. As an alternative, we filter HTTP traffic and retain GET requests whose 'accept request-header' field is filled with 'text/html'. As for the

DOI: 10.1002/cpe

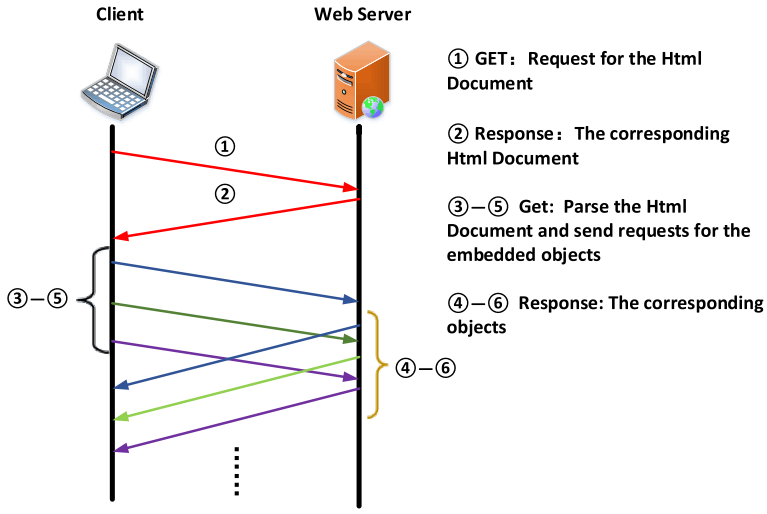

Figure 2. Interactive process of HTTP protocol.

automatically generated traffic, the values of 'user-agent' field are obviously different from the browser.

Then, we use the geolocation tool to obtain the approximate geographic location of the user based on his IP address. The result can help us infer the user's physical activity areas. In case that multiple sessions are linked, we even can depict the user's location trajectory. Finally, because a domain name may have multiple IP addresses, we convert the destination IP address to the domain name based on DNS A records. If the host field exits in the URL string, we can use the value as the destination domain; the DNS CNAME records can help us redirect multiple domain names to the same website. To mitigate the curse of dimensionality, we also replace all domains with the corresponding top-level domain names by using regular expressions.

4.2. Feature extraction

In data mining and pattern classification, it is very crucial to extract features which can reflect the true characteristics, because the selected features have a direct and significant impact on the classification accuracy. Previous works only extract domains and frequency from the traffic, ignoring the high-level application related information. In this paper, we primarily focus on four applications to mine details about users' habits, including HTTP, IM, e-mail, and P2P. We also extract some additional features that help to improve the detection rate in the following.

- Destination domain: As mentioned before, we regard the GET request referring to the 'text/html' resource as a website visit. So we filter the HTTP GET packets with the 'accept request-header' field and collect different values of the primary domain names. Suppose that there are D different primary domain names in total, we can write them as a |D|-dimensional vector. And the *i*th component of the vector means the access frequency of the corresponding domain.
- Access frequency of each domain: We regard the access frequency of a specific domain in a fixed-length time slot as 1. In this paper, we assume that 10 min is a time slot. If one interaction between the user and the website lasts within one time slot, we set the access frequency of the corresponding website to 1, and the value is added 1 for every extra time slot. At last, we count the total number for each domain by aggregating all corresponding interactions in a session.
- Geographic location of the user: By using the open geolocation API, we can get the approximate geographic location of the user based on his IP address. This feature is useful in the scenario of setting up a free virtual private network to attract traffic.
- Mail domain: The e-mail service is so widespread that many service providers offer free mailboxes to their users. Besides, most companies and universities also offer the internal mail services.

DOI: 10.1002/cpe

We believe that a user may have multiple e-mail mailboxes and select one based on the different communication purpose each time. For example, a user uses the company mailbox to report the work progress to his leader. When he or she wants to greet his or her friend, he or she may select some private mailbox according to his or her preferences. So, we extract the combination of different e-mail domains for each user.

- Search engine: Nowadays, varieties of search engines are available and have their own advantages, respectively. We believe that the user may make a choice based on the content type he or she wants to search. For instance, he or she likes to use the Google Scholar to find scholarly articles. When he or she wants to search some Chinese keywords, he or she prefers to use Baidu. So, we extract the combination of search engines used by the user in a session.
- User-Agent strings of browsers: By analyzing the URL strings, we can get the details about the user's system version, browser version, and the language preference. If a user has installed several operating systems and browsers, we even can identify him just with the user-agent strings.
- Usage frequency of IM: We count the number of connections established by QQ as the usage frequency of IM.
- Usage frequency of P2P: We count the number of connections established by eDonkey as the usage frequency of P2P.

In summary, a user's traffic in a session can be aggregated and represented as a behavior vector $\rightharpoonup\mathbf{x}$:

$$\vec{\mathbf{x}} = < x_1{}^{f_{x_1}}, x_2{}^{f_{x_2}} \ldots x_n{}^{f_{x_n}} > \tag{1}$$

In the preceding texts, x_i represents the ith feature, and f_{xi} represents the corresponding frequency.

4.3. Classification

It is very important to select an appropriate classifier for the classification problem. In terms of the classification accuracy, many researchers apply the support vector machine, which is well known for its high performance. However, the high dimensional features have a devastating impact on the effectiveness of support vector machine [20]. As a result, we apply the MNB classifier to identify users like recent work.

The MNB is a special model which implements the naive Bayes algorithm for multinomial distributed data. It also assumes that each feature is independent from one another. The MNB classifier estimates the probability of a vector $\vec{\mathbf{x}} = < x_1{}^{f_{x_1}}, x_2{}^{f_{x_2}} \ldots x_n{}^{f_{x_n}} >$ belonging to a particular class C_i as:

$$\begin{aligned} p_{C_i}(x) &= p(C_i|x) \\ &= \frac{p(C_i)p(x|C_i)}{p(x)} \propto p(c_i)\prod_{j=1}^{n} p\left(x_j|C_i\right)^{f_{x_j}} \end{aligned} \tag{2}$$

As we can see from Equation (2), the value of the frequency f_{xj} will have a great impact on the classification result. If we use the raw data to calculate the probability of the instance belonging to one particular class, the decision of the MNB classifier may be biased toward classes which contain features with high frequencies. Taking this reason into account, we apply the term frequency (TF) transformation and IDF transformation to deal with the raw data which are widely used in information retrieval and text mining. The TF counts the number of occurrences of the target term; the value is always standardized. The IDF is used to evaluate the weight of the term. It can diminish the weights of terms that occur very frequently and increase the weights of terms that occur rarely in the document set. By using TF–IDF transformation, the higher the value is, the more important the corresponding term is. In other words, if one feature occurs many times in an instance, it is an important component. But if this feature occurs many times in most instances, it is a low discrimination power feature and needs to diminish the weight.

DOI: 10.1002/cpe

The calculations of TF, IDF, and TF–IDF transformation are described in Equations (3–5):

$$f_{\mathrm{x}}^{tf} = 1 + \log(f_{\mathrm{x}}) \tag{3}$$

$$f_{\mathrm{x}}^{idf} = \log\left(\frac{N}{n_{\mathrm{x}} + 1}\right) \tag{4}$$

$$f_{\mathrm{x}}^{tf-idf} = f_{\mathrm{x}}^{tf} * f_{\mathrm{x}}^{idf} \tag{5}$$

In the preceding texts, f_{x} represents the frequency of the feature x in one instance, and n_{x} represents the number of instances containing the feature x, while N represents the total number of all instances. To avoid division by zero errors, the denominator n_{x} adds to 1.

5. THE IMPROVED METHOD

In this section, we mainly analyze the application layer data generated by the users. We first extract the web queries embedded into the top seven online shopping websites. After decoding these queries, we perform the web query classification to get the categories of products the users want to buy. Finally, we construct the preference models and improve the basic method to link multiple sessions of the same user with high accuracy.

5.1. Web query extraction and decoding

Compared with the web contents browsed by users, the web queries the users entered can reflect their individual preferences more intuitionally. Because the major search engine websites like Google and Baidu force users to utilize HTTPS protocol to encrypt the data, we cannot get these queries. As an alternative, we analyze the search queries in shopping websites. As shown in Table I, we select the top seven online shopping websites as the target.

HTTP protocol sends the parameters in the form of 'key = value' pairs attached to the URL request. So, we can collect the search queries by using string matching. Table I indicates the corresponding web query matching patterns of selected shopping websites. It is noteworthy that all the web servers encode the URL to replace unsafe ASCII characters. Nearly all the Chinese websites use the UTF-8 (UCS Transformation Format—8-bit) or GBK to encode the URL. UTF-8 is a variable-width encoding that can represent every character in the Unicode character set. The ASCII characters use 1 byte, while the Chinese characters cost 3–4 bytes. GBK is an extension of the GB2312 character set, and all the characters use 2 bytes. If we have no idea about the encoding standard, the text may appear garbled or as question marks or boxes after decoding. We need to carry out the encoding detection before decoding. In the beginning, we write the following regular expression according to the encoding rules to distinguish between UTF-8 and GBK: '^(?:[\x00-\x7f]|[\xe0-\xef][\x80-\xbf]{2})+$'. However, because of the overlap between two encoding spaces, our regular expression cannot correctly identify all the

Table I. Web query matching patterns of top seven shopping websites.

Top seven online shopping websites	Web query matching pattern
Taobao(Tmall)	search?q=(search_product.htm?q=)
JD	Search?keyword=
Amazon	field-keywords=
Suning	search.suning.com/
PaiPai	?keyword=
Yhd	search.yhd.com/c0-0/k
Dangdang	?key=

DOI: 10.1002/cpe

queries. To address this problem, we perform the automatic character encoding detection. We first assume that the phrase X is encoded in GBK. Then, we convert it to UTF-8. The output will be reversed to GBK as a new phrase X'. If X' is exactly equal to X, the assumption is right, and we implement the GBK decoding. Otherwise, the assumption is wrong, and we implement the UTF-8 decoding.

5.2. Web query classification

When the web queries are extracted and decoded correctly, we need to assign them to one predefined category, respectively. On this basis, we can construct the preference models by evaluating the same user's web queries in one session. Nowadays, there have been rapid developments in web query classification, which greatly impact the search engine optimization. Some related works [21] study the classification algorithms based on the open source corpuses. Because most people in our experiments enter web queries in Chinese, the corpuses of English is useless to us, and we are lacking open source Chinese corpuses. If we want to build a Chinese corpus, it will be costly, which we cannot bear. When we search some products in JD.com, we find that the HTML document returned from the server lists multiple possible categories in a descending order. Therefore, we decide to classify the web queries by real-time searching in JD.com and analyzing the response documents. The specific steps are listed as follows.

- Predefined categories: By analyzing the category structures of several Chinese online shopping websites, we have predefined 10 categories: Home Electronics; Consumer Electronics; Clothing, Shoes, and Bags; Food and Drink; Books Audio, and Video; Health, Kids, and Mother; Beauty and Clean; and Automotive. All the web queries initiated by users will be classified into one category.
- Chinese word segmentation: We find that the server of JD does not return the possible categories if the web query is too long. So we apply the IKAnalyzer [22] to segment a long Chinese query into several tokens, which is an open source and lightweight java-based library. In order to avoid making the segmentation too fine-grained, we modify the configuration file and set the IKAnalyzer in the smart mode to support the maximum word length segmentation. Then, we introduce a simple model called bag-of-words to represent these tokes. It means that we do not consider the order, semantic, or grammatical structures between tokens. If the web query is short, we do not segment it and regard it as a token.
- Real-time searching queries: For each token in a web query, we construct a real-time request to the JD.com and parse the returned HTML document by using the HtmlParse library. We create a filter to only extract the content attribute of the meta 'description' tag. The string matching algorithm is applied to get multiple possible categories, which are ranked in descending order of the weight. Because the category structure of JD is not exactly equal to ours, we map the returned results onto the predefined categories.
- Query classification: For each query, when we obtain categories with weights of all tokens, we can sum up the weights of the same categories. At last, the highest weight category is assigned to the query. If it happens that several categories get the same weight, we will choose the one which has fewer tokens.

5.3. Preference model

By counting up categories of the same user's web queries in a session, we can construct his preference model and represent it as a preference vector $\vec{\mathbf{y}}$:

$$\vec{\mathbf{y}} =< d_1, d_2 \ldots d_7, c_1, c_2 \ldots c_{10} > \tag{6}$$

In the preceding texts, d_1 to d_7 represent the top seven online shopping websites. If the user has submitted queries to the ith website in a session, the value of d_i is set to 1 and 0 otherwise. c_1 to c_{10}

DOI: 10.1002/cpe

represent 10 predefined categories. Similarly, the value of c_i is set to 1 if the user has initiated queries of the ith category.

5.4. *Overall process*

As shown in Figure 3, we improve the basic method by introducing the users' shopping preference models based on their web queries. The overall process is depicted as follows.

- Data preprocessing module: The captured unknown traffic is input into the data preprocessing module, which performs a series of operations, including converting IP addresses to the domain names, extracting the primary domain names, converting IP addresses to the geographic locations, and filtering HTTP requests.
- Feature extraction module: The preprocessed data are passed to the feature extraction module. In this module, eight features are extracted from the traffic generated by the same user in a session and represented as a behavior vector. This vector is passed to the classification module. In addition, the URL requests containing queries to the top seven online shopping websites are passed to the preference modelling module during the packet parsing.
- Preference modelling module: When the preference modelling module receives URL strings, it extracts and decodes the embedded web queries by using the automatic character encoding detection algorithm. Then, the module will construct and send a series of requests to JD.com to identify the queries' categories. On this basis, the preference model of one user is constructed and represented as a preference vector, which will be passed to the classification module.
- Classification module: We identify the unknown users in three phases. First, for each received behavior vector, we apply the MNB classifier to obtain the probabilities of it belonging to every class. We select the top N classes with their probabilities as a candidate list. Second, in the range of the N candidate classes, we perform classification on the corresponding preference vector by using naive Bayes classifier. The probability of each candidate class is also recorded. Third, for the same candidate class, we sum up the two probabilities. Finally, the unknown user is identified as the class which has the maximum value.

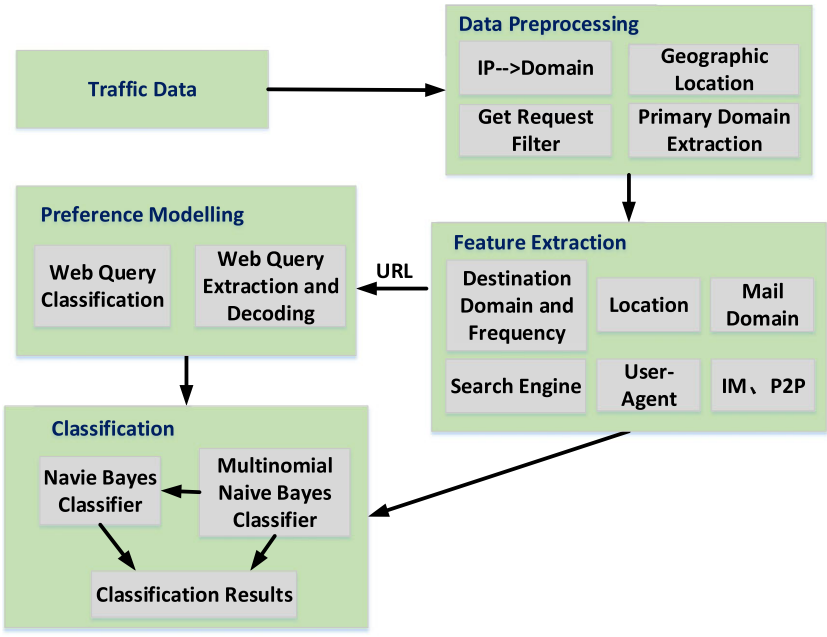

Figure 3. Overall process of the improved method.

DOI: 10.1002/cpe

6. EVALUATION

In this section, we mainly describe the experiment methodology used in the real-world environment to evaluate the performance of our methods. According to the previous work, we use the detection rate (accuracy) as the evaluation criteria.

6.1. Data collection

In order to evaluate the feasibility and scalability of our methods, we collect large amounts of traffic in a real-world setting. Considering the high-speed network traffic, we capture packets with PFQ [23], which is highly optimized for the multi-core architecture, as well as for network devices equipped with multiple hardware queues. We construct two datasets of different scale and content, including the *header* dataset and the *payload* dataset. As shown in Table II, the two datasets only contain IPV4 traffic.

The *header* dataset is generated by mirroring all traffic passing through the egress switch of our laboratory to a sniffing computer for 5 weeks. Fifty-five users' traffic are captured, but only 66 bytes of each packet are recorded. The *payload* dataset is obtained by sniffing all incoming and outgoing traffic of our college in cooperation with the network center of our university between 19 May 2015 and 11 June 2015. The packets are completely recorded, including the payload of the transport layer. In total, we get an over 25 TB dataset which contains 1200 individuals' traffic.

Considering that the recorded traffic data can disclose users' privacy, we inform the target users and get their permissions before traffic sniffing. Besides, we also get permission from the network center of our university. We promise that all the traffic are only used for scientific purposes. In the experiments, we just extract the features mentioned in the preceding texts and store them with the anonymous class labels in the database. Once the data analysis is finished, the raw data will be deleted.

6.2. Experiment results

Because the *header* dataset does not contain application data, we only extract destination domains, access frequencies, and e-mail domains as features to identify 55 users. In the classification phase, we use Weka [24] toolkit to implement the MNB classifier, and all instances are labeled with the corresponding anonymous class label. The *header* dataset is divided into two parts: the training set and the testing set. The training set contains traffic of the first, third, and fifth week, and the testing set contains the rest. There is no intersection between the training set and the testing set. When we identify users on the destination domains and access frequencies, the detection rate is up to 82.6%. By considering the e-mail domains, we increase the detection rate to 100%. The satisfying results have preliminarily demonstrated the effectiveness of the behavior-based tracking attack. However, there is a significant deficiency that the scale of the *header* dataset is too small. To address this problem, we implement more experiments on the *payload* dataset to evaluate the scalability of our approaches.

As described in the preceding texts, we have collected 1200 individuals' traffic by sniffing all incoming and outgoing traffic of our college. But the total number of staff and students in our collage does not exceed 500. The excess part may be experiment machines or movable equipment. To maintain high data quality, we select 509 active individuals whose total number of access frequencies in 24 days is more than 500. The number of different domains visited by these active users is up to 25 639. It means that the performance of our classifier will be seriously impacted by the high-dimensional features. To evade this issue, we try to select a subset of domains. Figure 4

Table II. Collected datasets in our experiments.

	Summary			
Datasets	Protocol	Containing users	Content	Duration
Header	IPV4	55	Only 66 bytes of each packet	5 weeks
Payload	IPV4	1200	The full packet	24 days

DOI: 10.1002/cpe

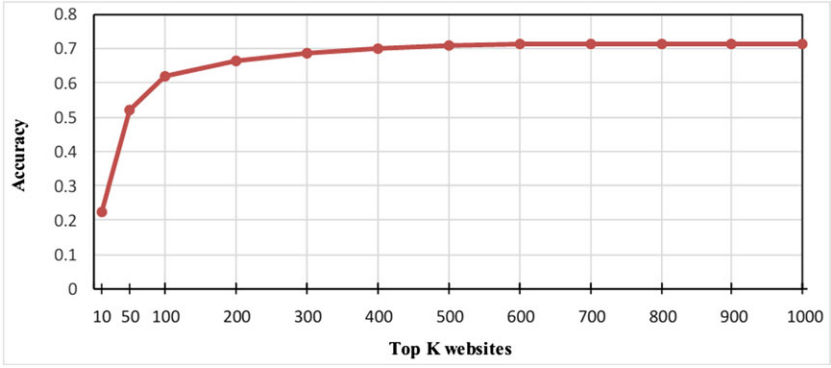

Figure 4. Detection rates of different values of *K*.

illustrates the detection rates of 10-fold cross validation when we use top *K* most popular domains. As we can see from Figure 4, when *K* varies from 10 to 50, there is an increase by 30%. But when *K* exceeds 800, there is a slightly drop in accuracy. We can conclude that redundant features will confuse the classifier and cause a decrease in accuracy. In subsequent experiments, we use top 800 domains as features.

To evaluate other features, we introduce the concepts of 'surprisal' and entropy like Eckersley's work [25]:

$$S(i_{n,f}) = -\log_2(P(i_{n,f})) \tag{7}$$

$$H(I_f) = -\sum_{n=1}^{N} P(i_{n,f}) * \log_2(P(i_{n,f})) \tag{8}$$

In the preceding texts, i represents an instance, and f represents a feature. $P(i_{n,f})$ is a discrete probability density function of the instance $i_{n,f}$. The surprisal S represents the amount of information associated with the value of a discrete instance, which is measured here in units of bits. $H(I_f)$ represents the entropy of the feature f in the overall sample space. Figure 5 illustrates the entropy of several features which are computed according to the equations mentioned in the preceding texts. Among them, the Mac address is used as a control. From Figure 5, we can see that the user-agent is a discriminational feature, especially when the user has multiple browsers.

Because the power of features cannot be easily predicted, we empirically tested a set of features. In order to avoid overfitting, we divide the *payload* dataset into training and testing set. The training set contains traffic of the first 15 days, and the rest constitutes the testing set. In the subsequent experiments, we always use these two subsets to evaluate the performance unless otherwise noted.

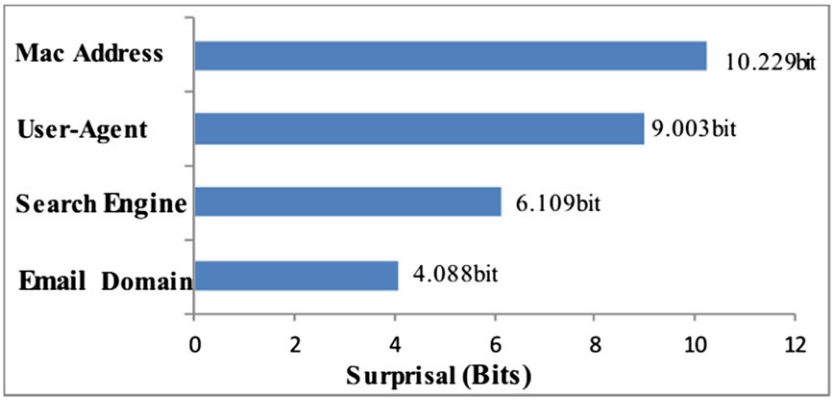

Figure 5. The entropy of several features.

DOI: 10.1002/cpe

The results are illustrated by Figure 6, which provides a visual impression of our recognition results and gives an overview of the features' impact on the detection rates.

The 'fundamental subset' means that we extract destination domains, frequencies, geographic locations, and usage frequencies of IM and P2P to create profiles. We plot other three histograms in the same figure by adding a specific feature successively. When all features are used, we can identify an average of 78.93% instances correctly.

Figure 7 illustrates the detection rates of three patterns of the basic method with G guesses. As the name suggests, the pattern called 'Raw' means that we identify users on the raw data. The 'TF' and 'TF–IDF' patterns use TF transformation and TF–IDF transformation to the raw data before the classification, respectively. The value of G is increased from 1 to 10, which represents the size of the ranking list of candidates. In other words, if the real identity of a user is in the ranking list of candidates, we consider the classifier successful. As we can see from the figure, the average accuracy of our basic method is raised to 85.61% when we use the TF–IDF transformation. When G is set to 10, the accuracy exceeds 90%.

Figure 8 illustrates the detection rates of the improved method using TF–IDF, with different lengths of the candidate classes list. When the length is set to 1, the improved method degenerates to the basic method. So, we can regard this point as the datum point. According to the trend line in the figure, if the length is not big enough, we get a small increase in accuracy. Because in that case, the accuracy is limited by the first classification result. In other words, the real class is not in the candidate list. However, when the length exceeds a certain value, which is 30 in the figure, the accuracy is no longer improved or even decreased, because the accuracy is affected by the result of the second

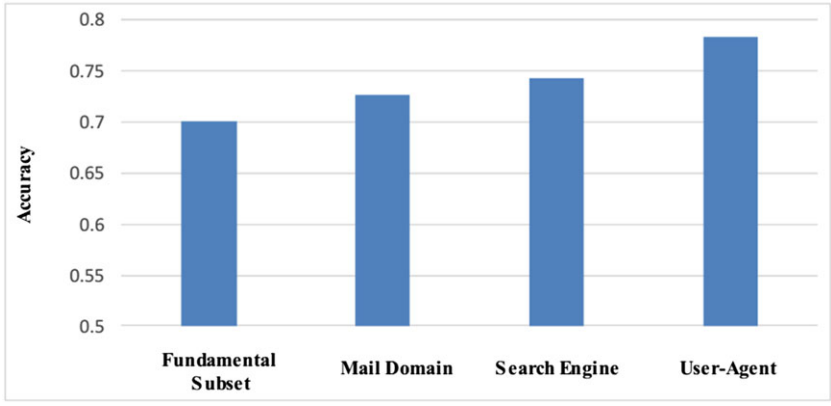

Figure 6. Influence of additional features on the detection rates.

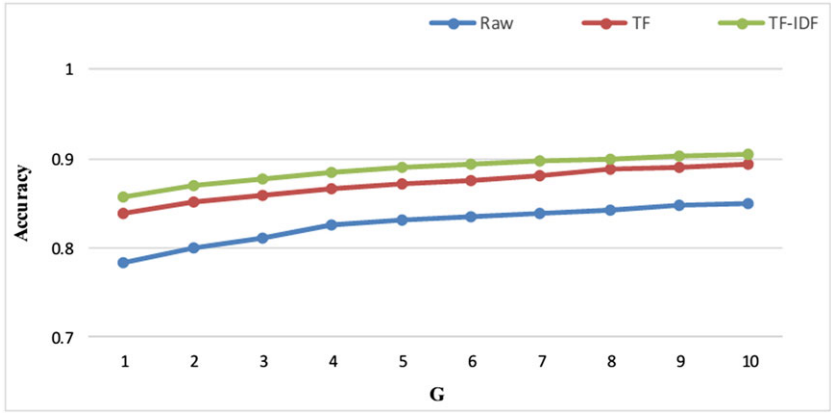

Figure 7. Detection rates of the basic method with different values of G.

DOI: 10.1002/cpe

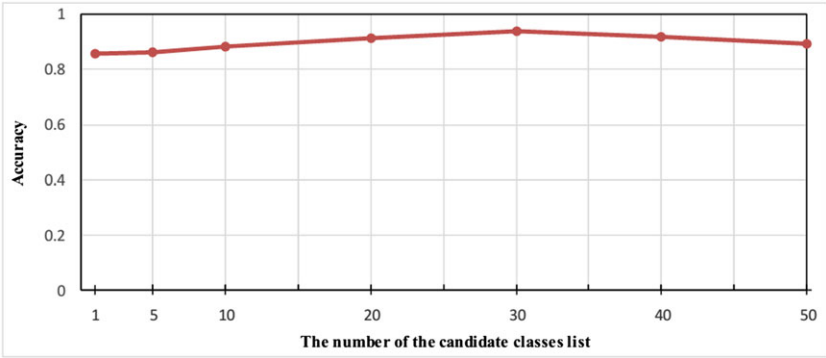

Figure 8. Detection rates of the improved method with different lengths of the candidate class list.

classification. In summary, it is appropriate to set the length of the candidate class list to 30 in this dataset.

Figure 9 illustrates the detection rates of three patterns of our two attack methods. As we can see from the figure, the user's preference models can help to improve the accuracy; the TF–IDF technology is important to the MNB classifier. When the TF–IDF transformation is used, the improved method can identify an average of 93.79% instances correctly. All these results can prove the effectiveness and scalability of our approaches in the large scale scenario.

7. COUNTERMEASURES

In this section, we mainly discuss and evaluate the feasibility and effectiveness of various countermeasures to resist the behavior-based tracking attack.

Herrmann *et al.* [7, 8] carry out the behavior-based tracking attack based on DNS queries and propose four possible countermeasures to mitigate the effectiveness, including using anonymizers, changing IP address frequently, DNS cache, and range queries. Although their threat model is different from us, the first two methods can be applied to our attack. We also add noise to the web traffic to cover users' behavior patterns.

7.1. Using anonymizers

Herrmann *et al*. believe that the anonymizers like Tor can hide the users' communication patterns. Indeed, Tor encrypts and forwards data through at least three onion routers to provide users with anonymity services. But there are still some weaknesses. He *et al*. [26] have proven that Tor anonymous communication traffic can be identified based on the TLS fingerprint or packet-size

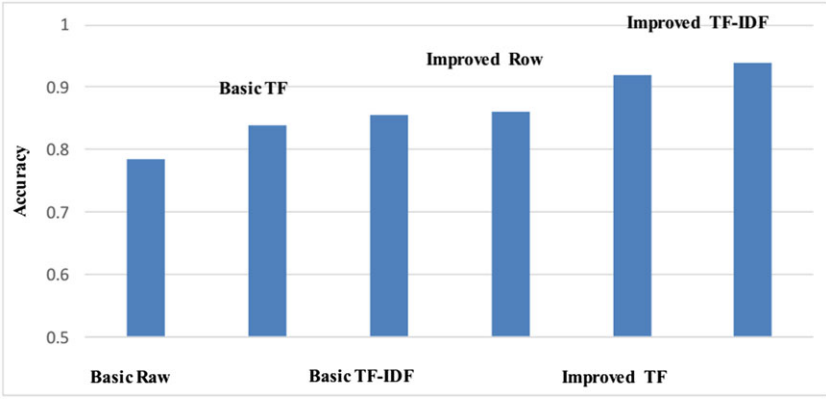

Figure 9. Detection rates of the two attack methods.

DOI: 10.1002/cpe

distributions. It means that we can disclose that someone is using Tor. Considering that Tor is blocked by internet service provider in China, it is possible that only a few people use Tor. As a result, we can infer the identity of the user who uses Tor with few candidates. In addition, the user needs to connect to a bridge to access Tor network. In a period of time, he can just get three different bridges. By matching the IP addresses of bridges, we can further filter candidates. Apart from inferring the user's identity, we also can deduce the communication content by launching the website fingerprint attacks [27, 28].

7.2. *Changing IP address frequently*

In the threat model, we assume that the user does not change his IP address in a session. The length of a session is set to 24 h by convention when we implement our attack. Even though the results of Abt *et al*. [6] indicate that they can identify eight individuals by monitoring only 5 min of network traffic, we think short-term traffic is insufficient to identify users in the large-scale dataset. Although the situation is bad, it is not as bad as what Abt *et al*. have said. We evaluate this countermeasure by varying the length of a session in our attack. As we can see from Figure 10, when the session is set to 8 h, the accuracy of the improved method is 76.49% which still poses a great threat. However, when the session is set to 4 h, it has decreased by 18%; the accuracy is further decreased to 18.89% when the session is set to 1 h. We can reasonably infer that the more frequently the user changes his IP address, the lower accuracy the attacker gets. Furthermore, we find that the accuracy of the improved method declines faster than the basic method. It means that the preference model is time sensitive. When the length of the session is too short, we cannot construct proper preference models which will lower the accuracy.

From this view, when the IP address is dynamically assigned, changing it frequently is a feasible solution. However, a practical problem is that the lease time configured by the DHCP server will restrict the frequency of changing IP address. During the DHCP lease time, we cannot get a new IP address unless modifying the MAC address. But changing the Mac address frequently in a large scale may cause the MAC address conflicts. Besides, this method will seriously affect the user's experience and increase the burden on the DHCP server.

7.3. *Adding noise to web traffic*

Although the range queries proposed by Herrmann *et al*. cannot be directly applied to our threat model, we get the idea from their method. We also try to add noise to the user's traffic. Considering that the HTTP traffic makes up a large proportion of our dataset, we only add dummy HTTP traffic in this paper. If we forge HTTP GET requests to the servers manually, it is very easy to distinguish the noise from the user's browser traffic. So, we design a browser extension to visit random websites automatically. In that case, the attacker finds it difficult to decide whether a website is visited by the program.

When the user opens his or her browser, the extension runs automatically. It first loads a local file of the top 1000 websites from Alexa and randomly choose M websites to build the dummy list. Then, it randomly selects a website following some probability distribution and opens the corresponding URL

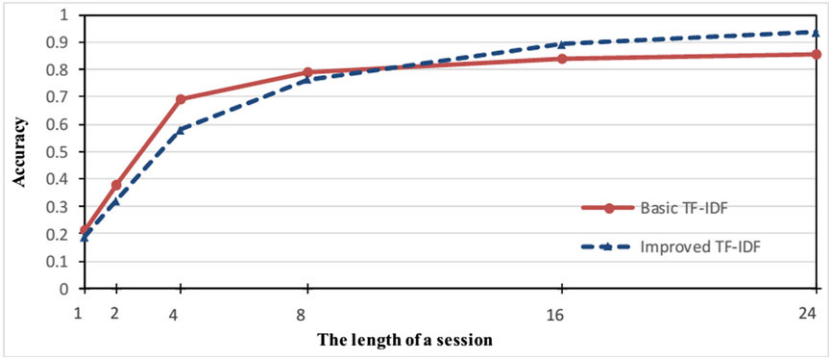

Figure 10. Detection rates of different lengths of a session.

DOI: 10.1002/cpe

in a new tab in the background repeatedly. The value of M and the probability distribution function are configured by the user. When the tab created lasts 10 s, it is closed automatically. To guarantee the user's browsing experiences, the new tab will not get focus in its life cycle, and at any time, there is only one dummy tab created by the extension. When the user closes his or her browser, the extension is no longer working.

To evaluate the effectiveness of this method, we need to run the extension for a period of time and record the generated traffic. It will cost us a great amount of time and money, and the results on the new dataset are unable to be validly compared with our previous results. Considering the reasons mentioned in the preceding texts, we implement a simple simulation by adding dummy records to the *payload* dataset. When the dummy list is constructed, it is import to simulate the random website selection with some probability distribution.

We utilize the pseudorandom number generation algorithm to realize the random selection. For example, when simulating the uniform distribution, we just repeatedly generate random integers within the range of 1 to M and then add records of the corresponding websites. Simulating the Zipf-like distribution is relatively more complex. In the initialization, we divide the dummy list into two subgroups, A and B, according to the parameter. For instance, when M is 100, and the ratio is set to 2:8, the subgroup A contains the first 20 websites, and B contains the remaining. To ensure that websites in the subgroup A own 80% of the dummy records, we carry out the two-stage randomization. In the first randomization, we generate a random integer within the range of 1 to 10. If the generated number is greater than 8, the subgroup B is selected. Otherwise, the subgroup A is selected. Then, we uniformly generate a new random integer in the range of the selected subgroup. By repeating this process, we can realize the random selection with the Zipf-like distribution.

Because the volume of dummy traffic is related to the runtime of the user's browser, we need to count the total runtime in a session. We assume that the browser is not closed if the time interval between two adjacent HTTP requests is within 10 min.

Table III shows the statistics of the top 10 active users' web behavior in a session in our dataset. As we can see, users visit a few favorite websites very frequently while most domains have a little traffic. The average ratio of the domains whose frequency less than 4 to the total number is about 75%. On the whole, each user's access frequencies of domains exhibit a Zipf-like distribution. So in the experiments, we select websites from the list, following the uniform distribution and the Zipf-like distribution, respectively. When we construct the Zipf-like distribution, we make that 25% of the websites in the list is selected with 75%.

Figure 11 illustrates the detection rates of different lengths of the dummy website list. To more directly reflect the effectiveness of this countermeasure, we only select the domain names and access frequency as features in classification; the baseline is generated by using TF–IDF transformation. We find that the performance of randomly selecting websites following the Zipf-like distribution is better than the uniform distribution. It is not difficult to understand. Because the TF–IDF

Table III. Statistics of the users' web behavior in a session.

	Statistics			
Top 10 active users	Means (different domains)	Means (different domains whose frequency exceeds 19)	Means (different domains whose frequency is less than 4)	The highest frequency
User 1	206.43	14.86	139.58	234.0
User 2	117.14	6.43	89.14	88.42
User 3	106.86	3.71	75.86	71.86
User 4	107.57	5.71	74.43	151.29
User 5	129.43	5.43	99.0	88.43
User 6	111.86	3.29	97.58	69.71
User 7	94.43	5.71	66.29	95.71
User 8	93.29	4.86	70.43	142.14
User 9	102.71	7.43	73.58	95.29
User 10	78.14	2.86	61.14	50.72

DOI: 10.1002/cpe

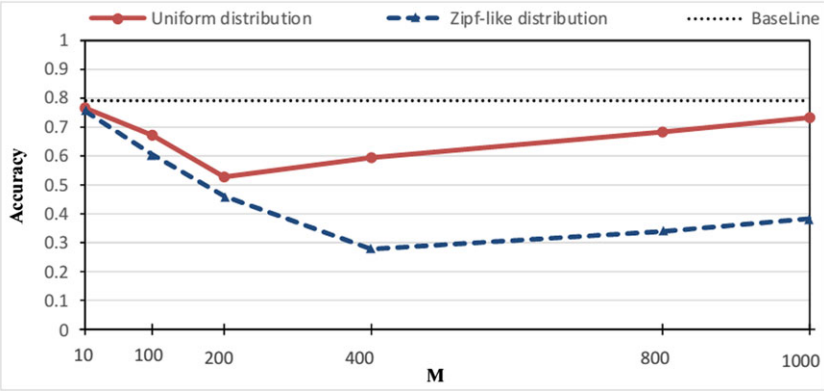

Figure 11. Detection rates of different lengths of the dummy website list.

transformation may reduce the impact of the dummy traffic, following the uniform distribution. When the length is set to 400, adding noise to web traffic following the Zipf-like distribution can reduce the accuracy to 27.78%, which has proven the effectiveness of this countermeasure.

8. CONCLUSIONS

Recent works of the behavior-based tracking techniques always perform experiments on the small-scale datasets, which cannot fully prove the feasibility of these methods. To address this problem, we propose a basic behavior-based tracking attack and extract features ranging from lower layer network packets to high-level application-related traffic. Furthermore, we improve the basic method by analyzing the user's search queries in shopping websites and constructing the preference models. We execute experiments on a real-world traffic dataset containing 509 active users. When using the TF–IDF transformation, we can identify 93.79% instances correctly with the improved method, while the basic is 85.61%. Based on these results, we believe that the behavior-based tracking attack is feasible in the large-scale scenario and should be carefully considered without contempt. To resist this attack, we also discuss and evaluate the feasibility and effectiveness of various countermeasures. For future research, we intend to construct a spark streaming framework to analyze the real-time traffic.

ACKNOWLEDGEMENTS

This work is supported by National Natural Science Foundation of China under grant nos. 61272054, 61572130, 61320106007, 61502100 and 61532013, and 61402104, Jiangsu Provincial Natural Science Foundation under grants BK20140648 and BK20150637, National High Technology Research and Development Program of China under grant 2013AA013503, the Fundamental Research Funds for the Central Universities under grant 2242014R30010, Science and technology project of China State Grid in 2015 (SGRIXTKJ [2015]-614), Jiangsu Provincial Key Technology R&D Program under grant BE2014603, Jiangsu Provincial Key Laboratory of Network and Information Security under grant BM2003201, Qing Lan Project of Jiangsu Province, and Key Laboratory of Computer Network and Information Integration of Ministry of Education of China under grant 93K-9.

REFERENCES

1. Tang J, Liu Z, Sun M, Liu J. Portraying user life status from microblogging posts. *Tsinghua Science and Technology* 2013; **18**(2):182–195.
2. Gong J, Tang J, Fong A. ACTPred: activity prediction in mobile social networks. *Tsinghua Science and Technology* 2014; **19**(3):265–274.
3. Olejnik L, Tran M, Castelluccia C. Selling off privacy at auction. Proceedings of the 14th Annual Network and Distributed System Security Symposium, San Diego, CA, USA, 2014; 1–15.
4. Mayer J, Mitchell J. Third-party web tracking: policy and technology. Proceedings of the 33rd IEEE Symposium on Security and Privacy, San Francisco, California, USA, 2012; 413–427.

DOI: 10.1002/cpe

5. Kumpošt M, Matyáš V. User profiling and re-identification: case of university-wide network analysis. Proceedings of the 6th international conference on trust, privacy and security in digital business, vol 5695, Berlin, Heidelberg, 2009; 1–10.
6. Herrmann D, Gerber C, Banse C, Federrath H. Analyzing characteristic host access patterns for re-identification of web user sessions. Proceedings of the 15th Nordic conference on secure IT systems, Vol 7127, Espoo, Finland, 2010; 136–154.
7. Banse C, Herrmann D, Federrath H. Tracking users on the internet with behavioral patterns: evaluation of its practical feasibility. Proceedings of the 27th IFIP TC-11 International Information Security and Privacy Conference, Crete, Greece, 2012; 235–248.
8. Banse C, Herrmann D, Federrath H. Behavior-based tracking: exploiting characteristic patterns in DNS traffic. *Computers & Security* 2013; **39**(A):17–33.
9. Abt S, Gartner S, Baier H. A small data approach to identification of individuals on the transport layer using statistical behaviour templates. Proceedings of the 7th International Conference on Security of Information and Networks, Glasgow, Scotland UK, 2014; 25–32.
10. Yang M, Luo J, Ling Z, Fu X, Yu W. De-anonymizing and countermeasures in anonymous communication networks. *IEEE Communications Magazine* 2015; **53**(4):60–66.
11. Pusara M, Brodley C. User re-authentication via mouse movements. Proceedings of CCS Workshop on Visualization and Data Mining for Computer Security, Washington, DC, USA, 2004; 1–8.
12. Zheng N, Paloski A, Wang H. An efficient user verification system viamouse movements, Proceedings of the 18th ACM Conference on Computer and Communications Security, Chicago, Illinois, USA, 2011; **139–150**.
13. Banerjee S, Woodard D. Biometric authentication and identification using keystroke dynamics: a survey. *Journal of Pattern Recognition Research* 2012; **7**(1):116–139.
14. Clickprints on the web: are there signatures in web browsing data. http://ssrn.com/abstract=931057 [27 October 2006].
15. Yang Y. Web user behavioral profiling for user identification. *Decision Support Systems* 2010; **49**(3):261–271.
16. Vel O, Anderson A, Corney M, Mohay G. Mining e-mail content for author identification forensics. *ACM SIGMOD Record* 2001; **30**(4):55–64.
17. Lackner G, Teufl P, Weinberger R. User tracking based on behavioral fingerprints. Proceedings of the 9th International Conference on Cryptology And Network Security, Vol 6467, Kuala Lumpur, Malaysia, 2010; 76–95.
18. Manning C, Raghavan P, Schütze H. *Introduction to Information Retrieval*. Cambridge University Press: New York, 2008; 258–263.
19. Hypertext Transfer Protocol (HTTP/1.1): message syntax and routing. https://tools.ietf.org/html/rfc7230 [June 2014].
20. Boser B, Guyon I, Vapnik V. A training algorithm for optimal margin classifiers. Proceedings of the 5th annual workshop on Computational learning theory, NY, USA, 1992; 144–152.
21. Broder A, Fontoura M, Gabrilovich E, Joshi A, Josifovski V, Zhang T. Robust classification of rare queries using web knowledge. Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, Amsterdam, The Netherlands, 2007; 231–238.
22. IK-Analyzer. https://code.google.com/archive/p/ik-analyzer [21 December 2012].
23. Bonelli N, Pietro A, Giordano S, Procissi G. On multi-gigabit packet capturing with multi-core commodity hardware. Proceedings of the 13th international conference on Passive and Active Measurement, Vienna, Austria, 2012; 64–73.
24. Witten I, Frank E. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Publishers: San Francisco, 2005.
25. Eckersley P. How unique is your web browser. Proceedings of the 10th Privacy Enhancing Technologies Symposium, Vol. 6205, Berlin, Germany, 2010; 1–18.
26. He G, Yang M, Luo J, Zhang L. Online identification of Tor anonymous communication traffic, *Ruanjian Xuebao Journal of Software* 2013; **24**(3), 540−556.
27. Cai X, Zhang X, Joshi B, Johnson R. Touching from a distance: website fingerprinting attacks and defenses. Proceedings of the 19th ACM Conference on Computer and Communications Security, Raleigh, NC, USA, 2012; 605–616.
28. Gu X, Yang M, Luo J. A novel website fingerprinting attack against multi-tab browsing behavior. Proceedings of the 19th IEEE International Conference on Computer Supported Cooperative Work in Design, Calabria, Italy, 2015; 234–239.

DOI: 10.1002/cpe