International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:12, No:6, 2018

Web Proxy Detection via Bipartite Graphs and One-Mode Projections

Zhipeng Chen, Peng Zhang, Qingyun Liu, Li Guo

***Abstract*—**With the Internet becoming the dominant channel for business and life, many IPs are increasingly masked using web proxies for illegal purposes such as propagating malware, impersonate phishing pages to steal sensitive data or redirect victims to other malicious targets. Moreover, as Internet traffic continues to grow in size and complexity, it has become an increasingly challenging task to detect the proxy service due to their dynamic update and high anonymity. In this paper, we present an approach based on behavioral graph analysis to study the behavior similarity of web proxy users. Specifically, we use bipartite graphs to model host communications from network traffic and build one-mode projections of bipartite graphs for discovering social-behavior similarity of web proxy users. Based on the similarity matrices of end-users from the derived one-mode projection graphs, we apply a simple yet effective spectral clustering algorithm to discover the inherent web proxy users behavior clusters. The web proxy URL may vary from time to time. Still, the inherent interest would not. So, based on the intuition, by dint of our private tools implemented by WebDriver, we examine whether the top URLs visited by the web proxy users are web proxies. Our experiment results based on real datasets show that the behavior clusters not only reduce the number of URLs analysis but also provide an effective way to detect the web proxies, especially for the unknown web proxies.

***Keywords*—**Bipartite graph, clustering, one-mode projection, web proxy detection.

I. Introduction

TODAY proxies, one of malicious websites, provide new opportunities to criminals who are rapidly industrializing their dark business over the Web [1]. And they are gradually becoming a cornerstone of Internet criminal activities supporting criminal enterprises such as spam-advertised commerce, financial fraud, and as a vector for propagating malware (e.g., so-called “drive-by downloads”) [2]. As the global web index (GWI) social report [18] shows, it is Indonesia and Vietnam which lead the way (22% each), followed by China (20%), as shown in Fig. 1. This trend is more and more pronounced in fast-growth markets. Over 90 million online adults in China have used one to access restricted social platforms. Many proxy servers steal and track users’ information for the sake of profits.

By proxy servers, attackers could anonymously surf the internet without revealing their own IP addresses. Hence, stepping stone detection is much vital as well as other malicious attack detection because it is quite flexible and can be used to perform any kind of attacks such phishing attacks, Denial-of-Service (DoS) attacks etc., which increasingly becomes a thorny issue. However, an increasingly large number of web users, a wide diversity of web proxies, and massive traffic data pose significant changes for web proxy detection for backbone networks or enterprise networks.

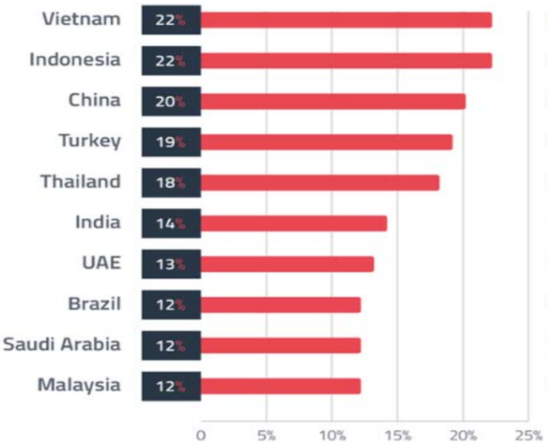


Fig. 1 Proxy/VPN users based on the Internet users aged 16-64 [18]

Certainly, there are many different types of proxies in different perspectives, such as the Web proxy, HTTP proxy, VPN, Tor and so on. Just as shown in Table I, we presented the different proxy protocols and their corresponding proxy products in chronological order. Staniford and Heberlein first demonstrated stepping stones detection in [3]. This approach is based on the packet’s content and vulnerable to encrypted stepping stones. Generally, mainstream methods utilize common features for the stepping stone detection. However, they still have their respective limitations. For examples, content analysis often introduces non-trivial overhead and needs update dynamically, rendering it impractical for large-scale network. Some utilizing instrumented browsers [19] or JavaScript engines [20] to visit limited websites for detecting the proxy may also be blocked due to fingerprinting techniques [21]. Thus, understanding the intrinsic properties of proxy and interactions between the proxy server and users is often critical in building an effective detection system.

In this paper, based on the discernible communication patterns, we present a system, ProxyHunter, to automatically detect the web proxies. The advantage about using web proxy is that it is free. This means one can enjoy all the benefits offered by the proxy server without having to incur any costs. Just as shown in Fig. 2, users could but type the URLs which they want

Zhipeng Chen is with School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China (e-mail: chenzhipeng@iie.ac.cn).

Peng Zhang, is with Institute of Information Engineering, Beijing, China (corresponding author, e-mail: pengzhang@ iie.ac.cn).

Qingyun Liu and Li Guo are with Institute of Information Engineering, Beijing, China (e-mail: liuqingyun@ iie.ac.cn, guoli@ iie.ac.cn).

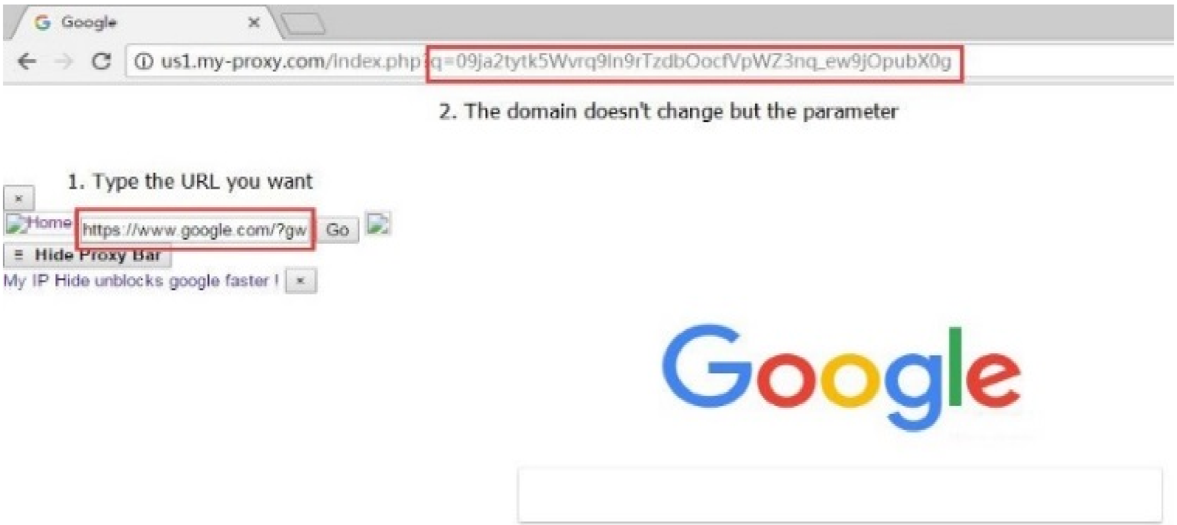
to visit, they will bypass the censorship to surf any resource unlimitedly.

TABLE I
TYPES OF PROXY

Tools	Time	HTTP Proxy	Web Proxy	VPN	Socks	Distributed Host
Freenet	1999	√				
TriangleBoy	2000	√				
Garden	2000	√				
FreeGate	2001		√			
Anoymizer	2002	√				
DynaWeb	2002	√				
UltraSurf	2002	√				
Circumventor	2003		√			
Tor	2004					√
Coral	2004					√
Hamachi	2004			√		
Psiphon	2004	√	√	√	√	
Firephoenix	2006			√		
GPass	2006			√		
Gtunnel	2007			√		
JAP	2007					√
Shadowsocks	2012				√	
Lantern	2013					√

In this paper, we study the communication patterns of users from a novel perspective, i.e. proxy users could access the limited services which are blocked. When the proxy does not work, the proxy user would seek some other alternative proxies. And the services frequently accessed would be stable. To obtain a comprehensive understanding of the communication patterns, we investigate 230,000 logs from 10,000 proxy users and 5,000 non-proxy users collected from one institute. Our key findings include: 1) Proxy users tend to seek more efficient proxy service and on average every user has two proxy service, and the top popular 300 proxies provide the 80% of stepping stone services approximately. 2) Unlike non-proxy users, proxy users have more stable communication patterns. They prefer to access those limited services.

Motivated by these findings, we propose a new approach of detecting web proxy traffic behavior by identifying and analyzing clusters of users that exhibit similar communication patterns. With the proxy cluster abstracting behavior patterns of a plurality of web users, the cost of traffic analysis is significantly reduced. Just as shown in Fig. 3, we could not understand users' destinations where we are at either the A side (between the Client and Proxy) or the B side (between the Proxy and Destination). So, we first set up one proxy honeypot capturing the network traffic logs, which we could know users' patterns behind proxies. And then, we use bipartite graphs to model network traffic between the web users and their actual visiting destinations, namely, the bipartite nodes between Client and Destination. As one-mode projections can effectively extract hidden relationships between nodes within the same vertex sets of bipartite graphs, we subsequently construct one-mode projections of bipartite graphs to connect Client hosts that communicate the same destination hosts. The derived one-mode projection graphs enable us to further build similarity matrices of web users, with similarity being characterized by the shared number of destinations between two hosts. Based on the similarity matrices, we apply a simple yet effective spectral clustering algorithm to discover the inherent web proxy user clusters. The behavior clusters not only reduce the number of analysis traffic, but also reveal detailed behavior patterns. Finally, we monitor the top domains visited by proxy user clusters and take them to our private web proxy checking tools implemented by WebDriver [10], a remote control interface that enables to remotely instruct the behavior of web browsers to exam whether the limited URL could be visited successfully. In success, the URL could be regarded as one web proxy.


Google
us1.my-proxy.com/index.php?q=09ja2tytk5Wvrq9In9rTzdbOocfVpWZ3nq_ew9jOpubX0g
Home
https://www.google.com/?gw
Go
≡ Hide Proxy Bar
My IP Hide unblocks google faster !
Google

2. The domain doesn't change but the parameter

1. Type the URL you want

Fig. 2 Using web proxy

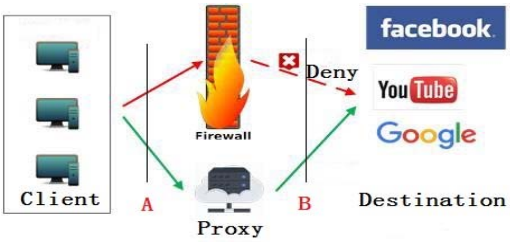
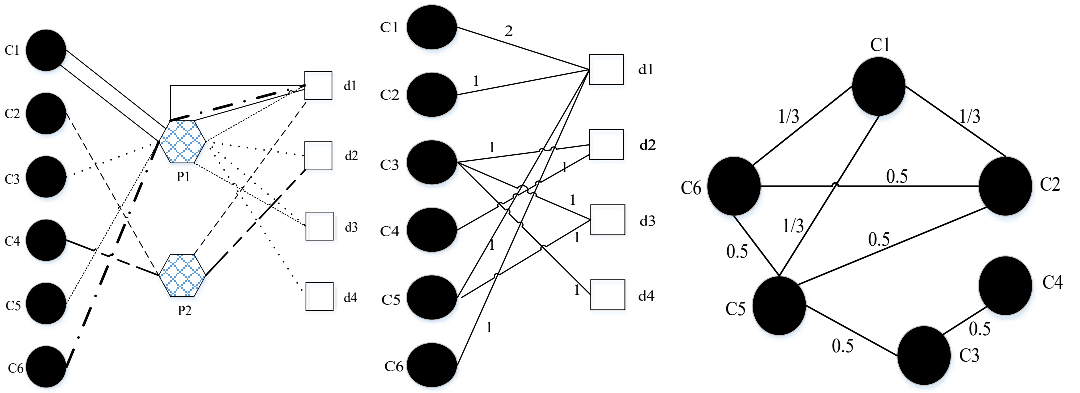


Fig. 3 Proxy model illustration

Challenges: Due to the great progress in evasion mechanisms, such as cloaking, fast fluxing, and domain generation, etc., detecting the web proxies is becoming more and more challenge. Unlike the normal servers, a large number of stepping stone hosts tend to be invisible, especially for the high anonymity mode. In the other words, traffic coming through a high anonymity stepping-stone host will look just the same as the traffic not using any proxy. In this case, the client can completely hide its identity from the censored institution. What's more, characteristic-based method is to identify traffic characteristics that are invariant or at least highly correlated across stepping stones. That's also vulnerable in varying traffic. On one hand, it is time-consuming on selecting the appropriate features. On the other hand, the features being trained change frequently over time. The classifier should thus be able to learn the evolution of these variations. In machine learning parlance, we need a classifier that is adaptive to "concept drift" [11]. What's more, some active measurements need employ additional packets to measure the inter-arrival times or the delays on the network. However, such mechanism would not work in most of the proxy servers with default configurations. In the following sections, we will discuss how we cope with these challenges in details.



(a) Example of web user visiting via proxy (b) Bipartite graphs building (c) One-mode projection graph of client hosts

Fig. 4 Modeling host communication using bipartite graphs and one-mode projection graphs

Contributions. Our main contribution is to provide the first solution for web proxy detection based on clustering the web users hidden behind a web proxy. We believe our solutions are significant since we deploy them within an existing network. In particular, other relevant contributions of the paper are summarized as follows:

- We build one proxy server honeypot and use bipartite graphs to represent proxy user communication patterns between users and destination hosts, and construct one-mode projection graphs to capture behavior similarity.
- We explore behavior similarity of web users using clustering algorithms and discover the inherent features of the proxy clusters. By monitoring the URLs visited by proxy clusters, we demonstrate practical benefits of exploring behavior similarity in detecting the web proxies through traffic traces.
- This methodology is based only on Client IP addresses and Destination URLs, does not require any information about HTTP heads (which are occasionally obfuscated) or packets (which are often encrypted or unavailable).

The rest of this paper proceeds as follows. In Section II, we present the problem. Section III shows the measurement of the communication patterns of the proxy users. Section IV gives the proposed method in detail. In Section V, the experiment and evaluation are shown. Section VI summarizes the related work. Finally, we conclude this paper in Section VII.

II. Problem Formulation

A. Goal

Our goal is to improve the efficiency of the web proxy detection. More precisely, we have proxy honeypot that allows us to acquire the web users' traffic logs. Based on the idea of one-mode projection of bipartite graphs, we discover the inherent web proxy user clusters. Then, we take the top domains visited by proxy user clusters to our private web proxy checking tools for verifying the truth of the web proxies. In short, we build the bipartite graph between Client and Destination, as shown in Fig. 4, clustering the similarity of web proxy users and figure out newly potential web proxies by observing the bipartite graph between Client and Proxy.

B. Stepping Stone

In this paper, we use the terms stepping stone and the proxy interchangeably. Both refer to the intermediary host of a connection chain which provides relay service.

We study the web proxy detection at a large scale networks based on records collected from our proxy honeypot and some non-proxy logs. More formally, we define a function that takes as input network traffic logs and outputs a bipartite graph containing 3-tuple of attributes in a certain window time.

Definition: A function Fin is defined as $\mathrm{Fin} : \mathrm{logs} -> \mathrm{G(srcIP, dst, e)}$, where the logs mean the records from our datasets, the set srcIP denotes the set of client IP, the set dst is the destination URLs, and the e is the edge set ($e \in \mathrm{srcIP} \times \mathrm{dst}$). Specifically, srcIP and dst are two disjoint vertex sets, and the weight of edge is the number of the same edges generated in a certain period.

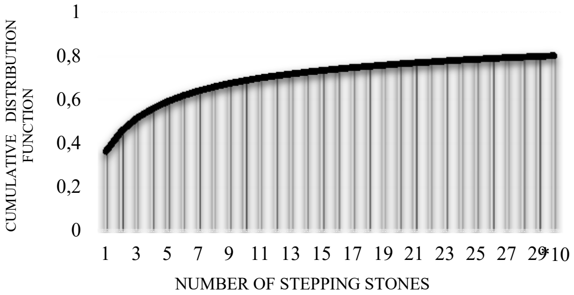


Fig. 5 CDF of the popular stepping stones

To study the social-behavior similarity of end-hosts in network traffic, we leverage one-mode projection graphs of bipartite graphs that are used to extract hidden information or relationships between nodes within the similar behavior patterns. Fig. 4 (a) illustrates an example of a simple graph that shows data communication between six client IP addresses (C1 – C6) and four destination URLs (d1-d4) via two kinds of web proxies (P1, P2). Fig. 4 (b) shows a corresponding bipartite graph, and Fig. 4 (c) is the one-mode projection of the bipartite graph on the vertex set of the client hosts (C1 – C6). Two nodes are connected by an edge in the one-mode projection if and only if the two nodes have connections to at least one same node in the bipartite graph [24]. We leverage one-mode projection graphs to explore the social-behavior similarity of client hosts. However, our interest is not creating a clustering of web proxy users, but rather our focus is to single out new potential web proxies even they maybe flux dynamically.

III. Measurement Communication Patterns

Our hypothesis is that the behaviors of users using proxies are similar and they have some inherent communication patterns. That is to say, users tend to seek more efficient proxy and they prefer to access those limited services. When the proxy does not work, the proxy user would seek some other alternative proxies.

To validate our hypothesis, we conduct a comprehensive measurement study on the communication patterns of proxy users. First, we captured real network traffic from a large institute where deployed our proxy honeypot in the backbone router spanning from January 03 to July 28, 2017 (PList1). Then, we analyzed and found the top 300 domains provide the 80% of stepping stone services approximately, as the cumulative distribution shows in Fig. 5. That means we could save enough resources for just monitoring these stepping stones frequently used. Next, we measured the stability of the communication patterns of proxy users. Specifically, we recorded the websites' categories frequently visited by each proxy user when we collected the datasets (January, 2017) and then re-checked their pattern status every two months. The results are summarized in Table II. We can see that proxy user pattern was relatively stable. A little changed during seven months on average. What's more, we randomly select 100 proxy users from PList1, and test how many proxies are used. We find that 78% of users just frequently use one kind of proxy and others used at least two proxies. Further investigation showed that users changed into other new proxy service primarily because the old one did not work, or they got one better. Hence, motivated by this finding, we could detect the unknown web proxies by focusing on the destinations hosts of users.

TABLE II
Stability of Communication Pattern

time	Category(Percentage)				
	social networks	video	filesharing	advertising	porn
January,2017	22.3%	19.6%	16.1%	14.4%	12.9%
March,2017	20.5%	21.4%	13.8%	15.6%	14.1%
May, 2017	19.3%	18.9%	15.7%	13.3%	13.6%
July, 2017	20.8%	22.7%	14.8%	12.9%	14.7%
Average	20.73%	20.65%	15.10%	14.05%	13.8%
SD	0.0107	0.0149	0.0089	0.0105	0.006

IV. Proposed Methodology

In this section, we give a detailed description of our proposed methodology. Web proxy URLs have a short lifespan since they are used only for a limited duration. Still, the inherent interests of users hidden behind the proxy would not flux frequently. Hence, we study the web proxy user community detection for singling out new potential web proxy URLs. Fig. 6 shows the schematic progress of our methodology. This methodology is defined in the following steps.

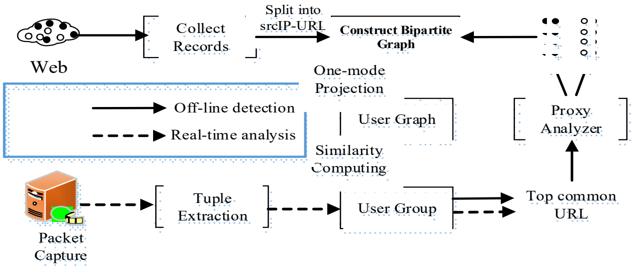


Fig. 6 Process framework of the method

A. Graph Cut via Partitioning Similarity Matrix with Spectral Clustering Algorithm

In this paper, the similarity measure $S_{u,v}$ is presented by the weighted edges between two client hosts u and v in the one-mode projection graph [24], because the weighted edges quantify the social-behavior similarity of communication patterns in traffic. Let $\mathrm{N(u)}$ and $\mathrm{N(v)}$ represent the numbers of Internet hosts with which two clients u and v have communicated, respectively. We then use $W_{u,v}$ to denote the weight for the edge between u and v in the one-mode projection:

$$W_{u,v} = \frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|} \qquad (1)$$

where $|\mathrm{N(u)} \cap \mathrm{N(v)}|$denotes the total number of the shared destination URLs in the bipartite graph between the two clients u and v, and $|\mathrm{N(u)} \cup \mathrm{N(v)}|$ denotes the total number of the uniquely combined destinations of u and v. Note that $\mathrm{u} \neq \mathrm{v}$.

```
Algorithm 1: Algorithm of discovering proxy users behavior clusters using an augmented spectral clustering algorithm
Input: network flow traces from honeypot and other non-proxy logs
Output: the clusters $C_{1,} C_2$, where (1: web proxy; 0: benign web sites)
1: Construct bipartite graphs of host communications from flow traces
2: Generate the one-mode projection of bipartite graphs and its weighted adjacency matrix, and then obtain the similarity matrix S.
3: Compute the diagonal matrix A where$\mathrm{A(i,i)} = \sum_{j=1}^{n} s_{i,j}$ .
4: Compute the Laplacian matrix $\mathrm{L} = A^{-1/2} S A^{1/2}$ and find the second smallest K eigenvalues.
5: For $\mathrm{i} = 1,2,\dots,\mathrm{n}$, let $y_i \in \boldsymbol{R^k}$ be the vector corresponding to the i-th row of S.
6: Clustering the points with the K-means into clusters $C_1, C_2$.
```

One interesting observation of the one-mode projection graphs for host communications lies in the clustered patterns in the weighted adjacency matrix. The scatter plots in Fig. 7 visualize the one-mode projection graphs for two different clusters, namely, the destinations of the web proxy users and non-proxy users. This observation motivates us to further explore clustering techniques and graph partitioning algorithm to uncover the patterns of web proxy users and so to detect the web proxy. Our study applies a simple spectral clustering algorithm illustrated in [17] where k=2(web proxy client cluster and non-proxy cluster). Algorithm 1 outlines the major steps of the proposed approach. The input of the algorithm is the network flows containing one 3-tuple, client IP, destination URLs, and the weight (the visiting times) during a given time window (we set $\mathrm{t} = 5\mathrm{min}$). The first step is to construct the weighted bipartite graphs and then generate the one-mode projection of bipartite graphs and obtain the similarity matrix S. So, we should find a partition of the graph such that the edges between different groups have a very low weight (which means nodes in different clusters are dissimilar) and the edges within a group have high weight. So, the simplest and more direct way is to adopt mincut strategy. Unfortunately, due to the unbalancing conditions of the mincut solutions, we introduce the Laplacian matrix $\mathrm{L} = A^{-1/2} S A^{1/2}$, where A is the diagonal matrix with $\mathrm{A(i,i)} = \sum_{j=1}^{n} s_{i,j}$ and $\mathrm{i} = 1,2,\dots,\mathrm{n}$, and then by the Rayleigh-Ritz theorem we compute the eigenvector and find the second smallest eigenvalue of L. The simplest way is to use the sign of eigenvalue as indicator function [17]. The output of this algorithm is the client IP, and each IP address is assigned to a cluster, having similar social relationship with the destinations.

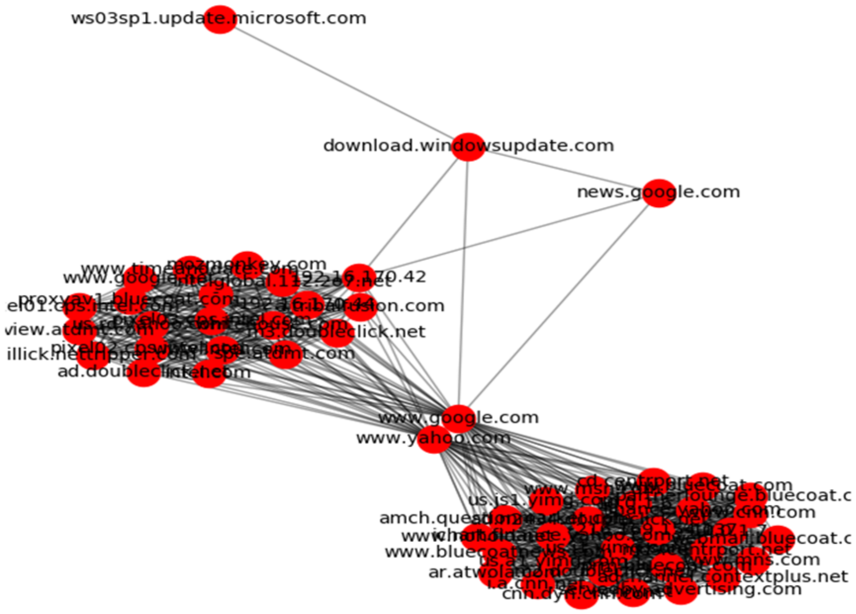
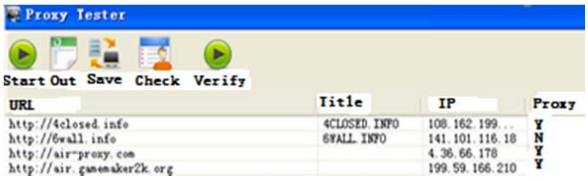


Fig. 7 Visualization for the one-mode projection of bipartite graphs

B. Web Proxy Detection

After acquiring the web proxy cluster, we keep a real-time list of these client IPs and extract all frequently visited domains among them. Just as shown in Fig. 2, the web proxy's domain does not change when we surf using the proxies. In a way, the most effective and simple method to examine if one URL is a web proxy is to check whether we could successfully visit some websites limited by firewall via the proxy URL. Hence, we developed one tools implemented by WebDriver [10]. The running result is shown in Fig. 8. We could import URLs in txt or excel format when we click the “start” button, and then we could click the “Verify” button, the result will be shown in the form of list. In the column of “Proxy”, “Y” means the corresponding URL is a web proxy, if “N” indicates not. For privacy reasons, the column “IP” has been mapped and anonymized.



URL	Title	IP	Proxy
http://4closed.info	4CLOSED.INFO	108.162.199...	Y
http://6wall.info	6WALL.INFO	141.101.116.18	N
http://air-proxy.com		4.36.66.178	Y
http://air.gamemaker2k.org		199.59.166.210	Y

Fig. 8 Web proxy detection by batch verification

V. Evaluation

In this section, we evaluate the effectiveness of the proposed method. Specifically, we want to answer the following questions: For the short lifespan of the web proxy, does the proposed method seek some unknown web proxies? Does using clustering web users lead to more effective web proxy detection? In our experiment, we evaluate our proposed method on a rich corpus with thousands of web pages from our proxy honeypot and some logs acquired from one backbone router. The experiment runs on a machine with Intel(R) Xeon(R) CPU E5-2682 v4 @ 2.50 GHz processors. Table III presents the statistical results during four different periods. The Client IP Num. is the number of source IP in proxy user cluster. The Proxy means the number of the true proxy in our experiment. The Detection Number indicates the number of the top URLs visited by proxy user cluster. The Verified Proxy represents the number of the true proxy after the verifications. Recall is the proxy detection proportion.

TABLE III
Detection Results

Periods	Client IP Num.	Proxy	Detection Number	Verified Proxy	Recall (%)	Time (μs)
1	106	1794	2173	1548	86.29	1302.3
2	101	1244	1666	1062	85.37	1011.2
3	83	1345	5009	1200	89.22	941
4	38	237	3336	207	87.34	363

From Table III, we could find the most significant time overhead is constructing the bipartite graphs among the Clients. But recall rate is irrelative to the Client Num. on the surface. To illustrate the results, we performed experiments using another two key metrics: density and expansion [23] shown in Fig. 9. The Density is the proxy detection rate on verifications. Higher values of Density imply that the resources needed to analyze a web proxy are used more efficiently. That means the more density rate, the less workload in the verification. The Expansion presents the average number of new proxies that our method finds for every client IP. For example, when there’re 100 top client IPs in web proxy clusters, the method then identifies 130 web proxies, then expansion of the system is 1.3. So, a higher expansion indicates that for every client IP a larger number of web proxies are found. Many factors could result in the varying expansion rate, such as the new evasive ways, the selected URLs from the proxy cluster, etc. When the expansion rate is becoming lower and lower, perhaps it is time for us to actively update our clustering mechanisms. Furthermore, we do not concern the overheads in constructing the bipartite graphs, because we do not need to build the graphs every day due to the stability of proxy users’ communication patterns.

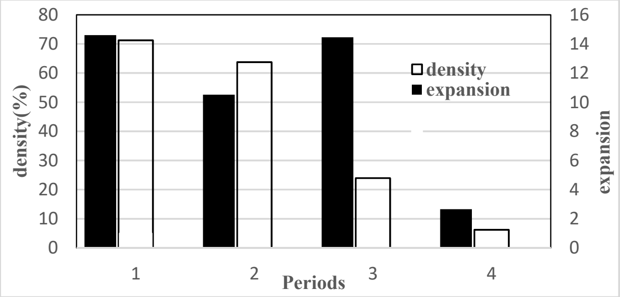


Fig. 9 Density vs. expansion

We also compared with existing work. As shown in Fig. 10, we compared the detection precision and time consumption between ProxyHunter and ProxyDetector [22]. We can find ProxyHunter outperforms ProxyDetector on the prediction time consumption, while their detection precisions seem almost alike. But, further investigation presents ProxyHunter could detect some new stepping stones.

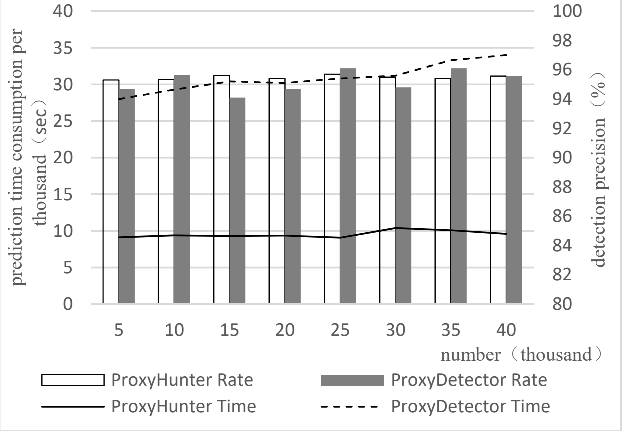


Fig. 10 ProxyMiner vs. ProxyDetector on time and rate

VI. Related Work

We present the related work on stepping stone detection from two points. One is academic and the other one is industrial.

In present academic study, signature-based and characteristic-based methods are two main stepping stones detection mechanism. The former is based on content, such as thumbprints [3] and watermarks [5], etc. Thumbprint creates a signature by matching some attributes of the packets or packet flows to detect the stepping stones, which are not generally effective at preventing new or unknown stepping stones. Watermark scheme injects a watermark in the incoming flow at a host connecting to server and checks if it exists on the outgoing flow, if yes this indicates that is a stepping stone host else a normal host. However, that is challenging on the encrypted traffic. The latter approaches are based on analyzing the packet transmission characteristics. Specially, Vahid [6] uses a machine learning based approach on different types of traffic logs to identify the incoming stepping stones base traffic on the server side. Liu et al. [7] proposes a server-based scheme to detect whether a host establishes a TCP connection to the server is a stepping stone or not by analyzing RTT (Round-Trip Time). But, the RTT is sensitive to network fluctuation and will differ between local traffic and traffic that traverses the WAN (Wide Area Network). There are certain characteristics of network traffic such as packet size, packet timestamp, ON/OFF periods, inter-packet delay, etc., which can help to detect stepping stone hosts [8], [9], [12], [13].

In industry, there are also some commercial solutions for the stepping stones detection. Lots of examples and a comparison of what methods are used are presented in Table IV. These methods include URL list, IP filters, Packet analysis, HTTP head filters, pre-defined rules and IP geo-location. IP2Proxy [14] analyses the HTTP header X-Forwarded-For for spotting proxy traffic. However, this is an optional header. Snort [4] extracts the heuristic rules from the blacklists for the detection. CIPAFilter [15] compares URLs with a list of known proxy websites and then blocks. The method needs to be updating over the time. MaxMind [16] uses the IP list to offer the detection service. This however runs into the same problem as using a URL list. ProxyDetector [22] consumes much time on the feature buildings for the machine learning prediction.

TABLE IV
VARIOUS DETECTION MECHANISMS

method name	URL blacklist	IP filter	Packet analysis	HTTP header filter	Pre-defined rules	IP geo-location
IP2Proxy				√		
Snort		√		√	√	
MaxMind		√				
CIPAFilter	√					
ProxyDetector			√		√	√

VII. CONCLUSION

In this paper, we proposed a novel method for automatically detecting the web proxies by clustering the web proxy users. A particular challenge in this domain is that web proxies are constantly evolving in a dynamic landscape. To prevail in this contest, we find users would seek other available proxy when the former proxy is down. So, the highlight of our method is extracting the inherent features of proxy users instead of seeking directly the short lifespan of web proxies, namely, using potential and stable communication patterns to detect the web proxy varying frequently. Through our experiments, we show that the method can correctly detecting the web proxy, and that it outperforms a previously proposed characteristic-based approach especially for detecting the unknown web proxies.

The future work includes how to dig more available features and applies into other proxy type detections. What's more, some false negative cases should be considered (i.e., one user may first login on one local Content Distribution Network, then connect to the proxy server).

ACKNOWLEDGMENT

The research work is supported by Supported by Strategic Priority Research Program of the Chinese Academy of Sciences under Grant (No.XDA06030602), National Natural Science Foundation under Grant (No.61402464, No.61402474, No.61602467) and National Key R&D Program 2016, 2016YFB0801304.

REFERENCES

[1] Li Z, Alrwais S, Xie Y, et al. Finding the linchpins of the dark web: a study on topologically dedicated hosts on malicious web infrastructures(C)//Security and Privacy (SP), 2013 IEEE Symposium on. IEEE, 2013: 112-126.

[2] Ma J, Saul L K, Savage S, et al. Beyond blacklists: learning to Detect Malicious Websites from Suspicious URLs (C)//Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2009: 1245-1254.

[3] Staniford-Chen S, Heberlein L T. Holding Intruders Accountable on the Internet(C)// Security and Privacy, 1995. Proceedings. 1995 IEEE Symposium on. IEEE, 1995:39-49.

[4] Snort. https://www.snort.org/, accessed on:15/11/2017.

[5] Peng P, Ning P, Reeves D S. On the secrecy of timing-based active watermarking trace-back techniques(C)// Security and Privacy, 2006 IEEE Symposium on. IEEE, 2006:15 pp.-349.

[6] Aghaei-Foroushani V, Zincir-Heywood / N. A Proxy Identifier Based on Patterns in Traffic Flows(M). IEEE, 2015.

[7] Lin R M, Chou Y C, Chen K T. Stepping stone detection at the server side(C)// Computer Communications Workshops. 2011:964 - 969.

[8] Kumar R, Gupta B B. Stepping Stone Detection Techniques: Classification and State-of-the-Art(M)// Proceedings of the International Conference on Recent Cognizance in Wireless Communication & Image Processing. Springer India, 2016.

[9] Shullich R, Chu J, Ji P, et al. A Survey Of Research In Stepping-Stone Detection (J). International Journal of Electronic Commerce Studies, 2011, 2(2).

[10] SeleniumWebDriver.http://docs.seleniumhq.org/projects/webdriver/, accessed on:16/11/2017.

[11] Gama J, Žliobaitė I, Bifet A, et al. A survey on concept drift adaptation(J). ACM Computing Surveys (CSUR), 2014, 46(4): 44.

[12] J. Brozycki. Detecting and preventing anonymous proxy usage, SANS Inst, 2008.

[13] Miller S, Curran K, Lunney T. Traffic Classification for the Detection of Anonymous Web Proxy Routing(J). International Journal for Information Security Research, 2015, 5(1): 538-545.

[14] IP2Proxy, http://www.fraudlabs.com/ip2proxy.aspx, accessed

on:13/11/2017.

[15] CIPAFilter, https://cipafilter.com/, accessed on: 10/11/2017.

[16] MaxMind, https://www.maxmind.com/, accessed on: 18/11/2017.

[17] Luxburg U. A tutorial on spectral clustering (M). Kluwer Academic Publishers, 2007.

[18] Global Web Index Q4,2013-Q3,2014 based on the Internet users aged 16-64,http://insight.globalwebindex.net/chart-of-the-day-90-million-vpn-users-in-china-have-accessed-restricted-social-networks?ecid= .

[19] Seifert, C., Welch, I. and Komisarczuk, P., 2007. Honeyc-the low-interaction client honeypot. Proceedings of the 2007 NZCSRCS, Waikato University, Hamilton, New Zealand, 6.

[20] Cova, Marco, Christopher Kruegel, and Giovanni Vigna. "Detection and analysis of drive-by-download attacks and malicious JavaScript code." Proceedings of the 19th international conference on World wide web. ACM, 2010.

[21] De Maio, Giancarlo, et al. "Pexy: The other side of exploit kits." International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment. Springer, Cham, 2014.

[22] Chen Z, et al. ProxyDetector: A Guided Approach to Finding Web Proxies.(C) The 42nd IEEE Conference on Local Computer Networks (LCN), 2017.

[23] Invernizzi L, Benvenuti S, Cova M, et al. EvilSeed: A Guided Approach to Finding Malicious Web Pages(C)// Security and Privacy. IEEE, 2012:428-442.

[24] Xu, Kuai, Feng Wang, and Lin Gu. "Behavior analysis of internet traffic via bipartite graphs and one-mode projections." IEEE/ACM Transactions on Networking (TON) 22.3 (2014): 931-942.

Zhipeng Chen received his bachelor degree in Dalian University, in 2011 and master degree in Beijing university of Posts and Telecommunications, in 2014. Now he is pursuing his Phd. Degree in School of Cyber Security, University of Chinese Academy of Sciences. His interests include big data analysis, artificial intelligence and network security.

Peng Zhang received his Phd. Degree in Institute of Computer Technology, Chinese Academy of Sciences, 2013 and now worked in Institute of Information Engineering, Chinese Academy of Sciences. His interests include service computing and network security.

Qinyun Liu received his Phd. Degree in Institute of Computer Technology, Chinese Academy of Sciences, 2015 and now worked in Institute of Information Engineering, Chinese Academy of Sciences. His interests include network traffic analysis and network security.

Li Guo graduated from Xiangtan university in 1993 and now worked in Institute of Information Engineering, Chinese Academy of Sciences. She is also a professor of Beijing university of Posts and Telecommunications. Her interests include machine learning and network security.