Malicious URL Linkage Analysis and Common Pattern Discovery

Shin-Ying Huang
Institute for Information Industry
Taipei, Taiwan
shinyinghuang@iii.org.tw

Tzu-Hsien Chuang
Institute for Information Industry
Taipei, Taiwan
zxchuang@iii.org.tw

Shi-Meng Huang
Institute for Information Industry
Taipei, Taiwan
frankjhuang@iii.org.tw

Tao Ban
National Institute of Information and Communications Technology
Tokyo, Japan
bantao@nict.go.jp

***Abstract*— Malicious domain names are consistently changing. It is challenging to keep blacklists of malicious domain names up-to-date because of the time lag between its creation and detection. Even if a website is clean itself, it does not necessarily mean that it won't be used as a pivot point to redirect users to malicious destinations. To address this issue, this paper demonstrates how to use linkage analysis and open-source threat intelligence to visualize the relationship of malicious domain names whilst verifying their categories, i.e., drive-by download, unwanted software etc. Featured by a graph-based model that could present the inter-connectivity of malicious domain names in a dynamic fashion, the proposed approach proved to be helpful for revealing the group patterns of different kinds of malicious domain names. When applied to analyze a blacklisted set of URLs in a real enterprise network, it showed better effectiveness than traditional methods and yielded a clearer view of the common patterns in the data.**

Keywords- malicious URL; malicious domain name; drive-by download; linkage analysis

I. INTRODUCTION

Malware usually delivered through email, and malicious website, abnormal behavior can be discovered by analyzing the URL list in network traffic logs. For example, attackers might provide some information to lure a victim to a web page. The malicious code's trigger exploits vulnerabilities in web browsers and applications such as a Flash player or PDF reader. This kind of information so called backlist plays major roles in web security.

Because of the general difficulty in representing the infection and spread phases of malware, few researchers have focused on identifying the pattern of malicious website. This study focuses on the infection and communication phase of malicious domain name. In particular, we build a model of the connections between the malware infection phases of different, but related, malware infrastructures. By exploring the connections, we are able to identify potential new malicious websites related to malware infection or potential unwanted program, focusing on the most significant infection events. In brief, this paper presents a technique for analyzing and exploring the distribution patterns among drive-by malware. The proposed method is suitable for detecting unknown malicious domain names, and it complements the use of blacklists. Specifically, we make note of the server IP, host, URL, FQDN, domain name, parameter, path, and other features of an URL. We also adding verification mechanism for checking known malicious domain name. These features are then examined and joined together in a graph construct. We have demonstrated the exploration results of a set of malicious domain names by visualizing the malicious domain distribution, showing more suspicious domain name have been found with various attack patterns.

II. LITERATURE REVIEW

In this section, we give some background knowledge and describe related works in the areas of malicious domain name analysis as applied to drive-by download attacks, malware distribution infrastructures, and orchestrated behaviors of malware installations.

A. Drive-by-Downloads

Traditionally, malware will be only activated when a user opens an infected file. Unfortunately, cyber-attacks have become much more sophisticated over recent years, and this level of user-interaction is no longer required. Malware, used as a vehicle for hacking and other cybercrime, may be delivered by a website's hidden content, such as HTML tags, scripts, and advertisements. The simple act of visiting such a site is sufficient to result in compromise of the victim's system and then leakage of personal information.

Malicious downloads may be placed on websites that looks to be legitimate. You might receive a link in an email, a text message, or a social media post that tells you to read a content of interest on a website. When you open the page, the download starts in stealth and the malware is installed on your computer while you are enjoying the article or cartoon.

Security researchers detect drive-by downloads by keeping track of web addresses that they know have a history of malicious or suspicious behaviors [8], and by using crawlers to explore the web and visit different pages. If a web page initiates a download on a test computer, the site is marked as risky. Links in spam messages and other communications can also be used as source lists for malicious content.

To identify drive-by download web pages, researchers usually adopt three strategies: (i) visiting them with honeypots, (ii) statically (or dynamically) analyzing their content, and (iii) studying the set of malware distribution paths leading to malicious payloads.

978-1-7281-0858-2/19/$31.00 ©2019 IEEE

The PhoneyC [21] and CAPTURE-HPC [25] projects both used honeypots to detect malicious websites. In particular, researchers used vulnerable browsers running on virtual machines to visit web pages, looking for signs of infection in the guest system. The method generates few false positives, but it is vulnerable to fingerprinting and other evasions [11]. Moreover, it experiences a high number of false negatives.

Content analyzing of web pages is another efficient solution, sensing malicious patterns [9][18] or performing static [6] (or dynamic [12]) analysis of JavaScript code to identify malicious websites. However, these solutions become vulnerable as cyber-criminals escalate their code obfuscation and perfect their ability to fingerprint analysis platforms. Researchers need to consider these evolving evasions as a sign of growing maliciousness. For example, Kapravelos et al. [10] introduce a number of possible attacks that leverage weaknesses in the design of high-interaction honeyclients to evade their detection.

B. Malware Distribution Infrastructures

Malware distribution occurs via the infrastructure of a malware provider's storage facilities and transportation systems. Together, they convey malware and malicious services from malicious producers to websites or droppers (or both) [15][26]. On the other hand, legitimate distribution networks allow a company to deliver products from a manufacturer to a retailer. The approach of many malware distribution schemes is to counterfeit legitimate networks, and during the counterfeiting activity, different malware infrastructures often share resources to increase their exposure and lower their costs.

To overcome the shortcomings of the previous solutions, researchers have initiated the study [8] and detection of malicious paths in their entirety [24], from the initial landing pages to the infected page. Previous works have studied spam [2], infrastructures and economies of malvertising [23]. Infection paths have been passively detected, analyzing the redirection chains in network traffic [7][28], or actively detected through honeypots [16][28]. Other researchers have proposed to detect malicious downloads by reputation-based systems; for example, in Polonium [20], belief propagation in a tera-scale graph is used to compute the safety of billions of downloaded executable files from an initial seed of known benign and malicious files. The approach achieves CAMP (Content-Agnostic Malware Protection) achieved in a browser with minimal network requests [22].

Other researchers have proposed to leverage information about malware distribution traits, giving a more complete view of the attackers' systems and the lifecycles of the infected hosts. For example, the Nazca system [14] generates graphs containing all the heterogeneous entities involved in a malware distribution network instead of focusing on individual incidents. The Nazca system has done a good job in identifying malware distribution in large-scale networks. However, they did not consider Secure Sockets Layer (SSL) transmission or targeted attacks on an enterprise. A downloader-graph abstraction technique is proposed in [15], which captures the download activity on end hosts, and it explores the growth patterns of benign and malicious graphs. Their downloader graphs have the potential to expose a large amount of malware download activity, which may otherwise remain undetected. Through a crawling technique, Li et al. [17] studied malicious infrastructures by exploring the malicious neighborhoods of an initial set of dedicated malware-spreading hosts. The technique uses graph mining to classify hostname-IP clusters as topologically dedicated malicious hosts.

C. Orchestrated Behaviors of Malware Installations

A download action initiates a transfer of data from a remote system along a path from endpoint to endpoint. When we trace this path, called the entry path, it tracks back to the source of the download. After entry paths to malicious network infrastructures have been found, they can be used to further explore the networks, discovering their structure and interconnections [4], and thus help explore more malware sources. Researchers have done so by crawling in WEBCOP [27] and by exploiting search engine indexes in EVILSEED [13] to reach further into these networks and thereby obtain a more complete view.

The traditional solution is to refer to a blacklist to identify malicious downloads. However, using a blacklist is not enough to defend against a zero-day attack because there is always a time lag between the launch of the attack and the update of the blacklist. In addition, malicious downloads camouflage themselves to look like benign file downloads, or they use benign iframes [19]. Therefore, this study takes a macroscopic view of download behaviors and builds up a relation graph, also called a malware distribution graph. We link together the nodes of malware entry paths in order to reconfigure the malware distribution graph. Similar to orchestral music, the malware downloads and the supporting distribution infrastructure become more apparent when observing a larger part of the orchestrated network [5][19].

III. PROPOSED METHOD

The proposed malware distribution analysis mechanism is shown in Figure 1. Our proposed mechanism can automate the detection of malware distribution and reveal hidden threats in an enterprise network. The input of the system is a list of malicious domain names (black list). The system can be deployed inside an enterprise network to monitor the proxy logs in real-time mode.

Figure 2 shows the definition of nodes involved in malware distribution. We extended the categories beyond the usual categories defined in [14] in order to illustrate more in-depth relationships among different malicious URLs.

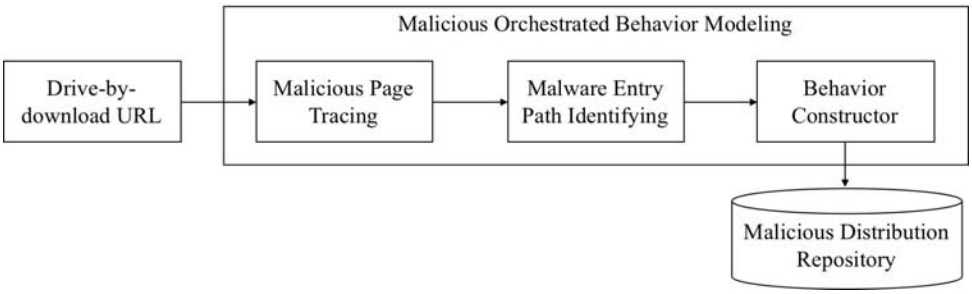

Figure 1. The system architecture.

A. *Modelling Malicious Orchestrated Behaviors*

The purpose of finding malicious orchestrated behaviors is to discover the relationship and reveal common pattern of a set of malicious or suspicious websites. As a starting point, we collect URLs from public and private blacklists, which include malicious files such as Trojans, ransomware, Java malware, Zeus malware, and Angler EK malware. Both blacklists update their URL lists daily, so we retrieve new seeds every day. We designed the heart of our tracing system, called the crawler, using a breadth-first search algorithm. Because attackers try to avoid detection by obfuscation and multilayer attacks, we used a web crawler written in Python called js-crawler [3], which in turn uses Webkit and executes Javascript code to enhance its performance. It can crawl all contents of a malicious URL, and save the files before moving to the next layer of URLs. We designed it to stop after three layers, which is a typical depth for a multilayer attack.

Next, based on each malicious website, we collect the proxy logs, which record all downloaded files. The system constructs entry paths containing URL information (i.e., the server IP, host, URL path, FQDN, domain name, etc.). Note that the enterprise proxy logs given to the system contain files from many benign domain names, such as google.com, yahoo.com, and yahoo.com, and they need to be filtered out. For this purpose, we use a list, published by Alexa [1], of the top 500 sites in Taiwan. This small list serves our purpose better than Alexa's larger list of one million websites because too much whitelist filtering may affect the system's efficiency.

Each constructed entry path might be redirect to other malicious websites each of which is a downloaded file. We focus on downloaded files because malware usually disguises itself or hides in downloaded PDF files, DOC files, Flash files, etc. When we get the malicious candidates, we analyze them individually for signs of suspicious behavior. Specifically, our system can connect malicious candidates and illustrate detail URL components, which helps identify candidates with similar patterns. In a final stage, the system compares malicious candidates within the same project node and thereby constructs collections of malicious orchestrated behavior; these collections are then sent to malicious domain repository.

For each explored domain name, we refer to the open source threat intelligence tool such as Virustotal and Google Safe Browsering to lookup each domain name and IP in order to help better labeling what types of malicious behavior it is. We use the identification result to help explain the malicious types of each domain name cluster.

All the known malicious domain name and the correlated domain name will be reasonably put into suspicious watch list. For example, if one malicious domain name is confirmed to be a malware category. Moreover, if another malicious domain name in the same group happened to be also known as malware category, we could infer that this group of domain names tend to have similar malicious pattern.

Figure 2. Definitions of nodes in malware distribution.

B. *Maintaining the Integrity of the Specifications*

The next task is to find additional related malicious activities. We use information extracted from the proxy logs. We consider whether any node in a collection has some relation with entities that are not yet in the collection. If so, then we add the corresponding entity. The following rules of the Nazca system [14] determine the relations used:

- URLs belonging to the same domain name or FQDN are related (resource reuse).
- Domain names/FQDNs being hosted on the same server are related (resource reuse).
- Files being downloaded from the same URL are related.
- URLs having the same path or file name are related (availability, ease of deployment).
- Files being fetched by the same clients are related.

We also add the following rules to connect other weak links: (1) FQDN include complicated long string, (2) files was named with simplified short string, and (3) path structure has the same part, and folder has regularly change.

IV. EXPERIMENTAL RESULTS

In this section, we describe our experimental design and results based on using real world data. The proposed mechanism is able to collect and extend the set of malicious URLs/domain name/IPs and the suspicious payloads. The following subsections present the details.

A. The Blacklist Dataset

The blacklisted malicious domain names are generated from a large-scale dynamic malware-analysis system hosted in a sandbox environment. Collected from May 2017 to March 2018, the dataset contains 4100 malicious domain names which are hard-coded or soft-coded in the malware programs under analysis. In other words, these blacklists are the soruce of the websites, and our poposed approach is designed to explore more related malicious or suspicious website.

B. Malicious domain name/URL Extension

The system analyzes the correlated URLs and their proxy records. First, a crawler impersonated a general user visiting malicious URLs, and we downloaded the malware up to three layers deep. We crawled the blacklist dataset, allowing it to grow. However, in the malicious websites, we also discovered benign web tools (e.g., Google Analytics, whosis web service), indicating that some malicious websites were trying to analyze what kind of visitors were tempted into the website. After filtering out the web analytic tools, 4,201,29 connectable malicious URLs are left for analysis.

In the second stage, the system constructed the malicious URL's logs and a malware hash (using md5) which were then correlated to model the malicious orchestrated behaviors. Each malicious orchestrated behavior model represents a relationship between malicious URLs. When the system modeled these online malicious logs and their malware hash, it found more than 3,526 groups, with each group displaying identifiable shared features such as having same malicious domain name category, using the same malware, having the same website path, or employing a shared server among multiple domain names [14]. This information can help to identify more malicious behaviors.

C. Graph Analysis

Figure 3 to Figure 5 show parts of the generated malware distributions. Note that the definition of node type is shown in Figure 1. We found that malicious domain name appeared in the same cluster tend to have similar malicious pattern. In Figure 4, a large cluster occurred because these websites often redirect user to other domain names, which increase the risk of downloading malware, download potential unwanted program or redirecting to phishing website. The marked domain names: wizoffer.gr, dryversdocumentsandcustomer.com were verified to be unsafe websites because of installing unwanted or malicious software on visitors' computers.

Such malicious websites tended to use various FQDN to increase the chance of victims visiting these malicious websites and ultimately downloading their malicious files. For example, in Figure 3, shows a graph example A with several different servers, IPs, and files. These separated clusters are connected through the redirection of domain names and FQDNs. Such malware distribution is like a maze with many landmines, and these domain names should be marked as high-risk websites despite not being on blacklists. Figure 4 shows the detail information in example A, we can verify that the correlated domain names are mostly malicious, too.

Some malicious websites tended to use various FQDN to increase the chance of victims visiting these malicious websites and ultimately downloading the malicious files.

Example B in Figure 5 shows a graph with several different domain names, IPs, and files. However, all the URLs derived from the domain names have common path pattern *FsMflooY*, which confirms that leveraging opened source website scan and lookup tools can help to explain the category of malicious websites. In Figure 6, all the domain names within the same group are all malicious in different levels

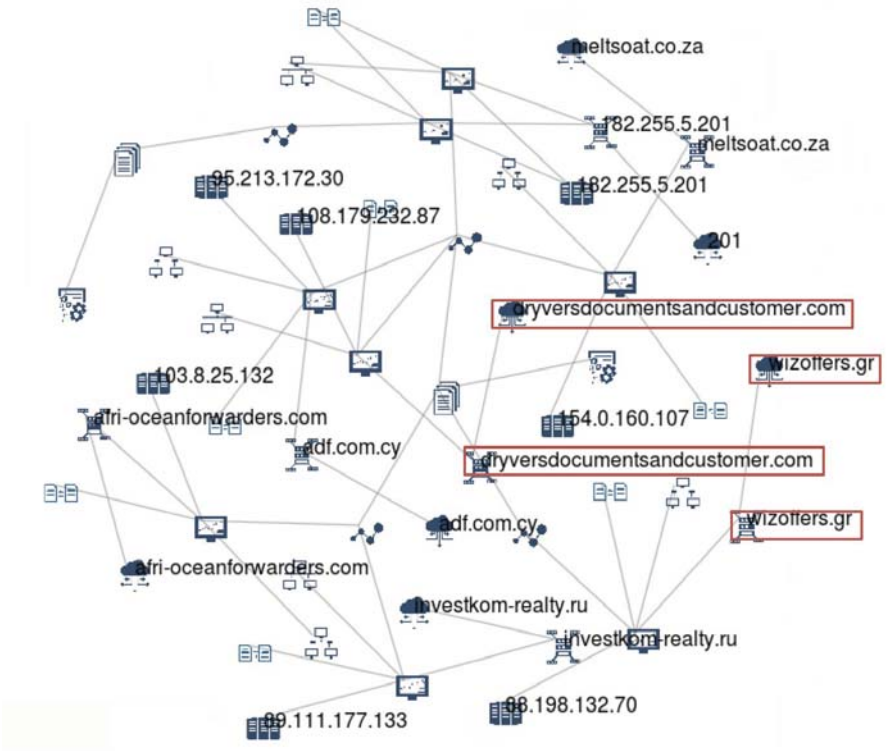

Figure 3. Example A: An example of malware distribution- unwanted softwar.

TYPE	ITEM	VIRUSTOTAL
Date	2017_04_24	
Domain	182.255.5.201 investkom-realty.ru dryversdocumentsandcustomer.com wizoffers.gr afri-oceanforwarders.com meltsoat.co.za adf.com.cy	virustotal ratio : 6 / 67 virustotal ratio : 4 / 65 virustotal ratio : 8 / 64 virustotal ratio : 5 / 64 virustotal ratio : 8 / 65 virustotal ratio : 3 / 65 virustotal ratio : 2 / 65
Server	103.8.25.132 108.179.232.87 182.255.5.201 88.198.132.70 154.0.160.107 95.213.172.30 89.111.177.133	
URL	http://dryversdocumentsandcustomer.com/cgi-sys/suspendedpage.cgi http://wizoffers.gr/cgi-sys/suspendedpage.cgi http://adf.com.cy/cgi-sys/suspendedpage.cgi http://182.255.5.201/~bemkmund/two/turbo.exe http://182.255.5.201/cgi-sys/suspendedpage.cgi http://154.0.160.107/cgi-sys/suspendedpage.cgi http://95.213.172.30/cgi-sys/suspendedpage.cgi http://meltsoat.co.za/cgi-sys/suspendedpage.cgi http://investkom-realty.ru/cgi-sys/suspendedpage.cgi http://108.179.232.87/cgi-sys/suspendedpage.cgi http://afri-oceanforwarders.com/cgi-sys/suspendedpage.cgi	
File Name	suspendedpage.cgi turbo.exe	
Path	/~bemkmund/two/turbo.exe /cgi-sys/suspendedpage.cg /cgi-sys/suspendedpage.c /cgi-sys/suspendedpage.cgi	

Figure 4. Example A: An example of malicious doamin report- - unwanted software.

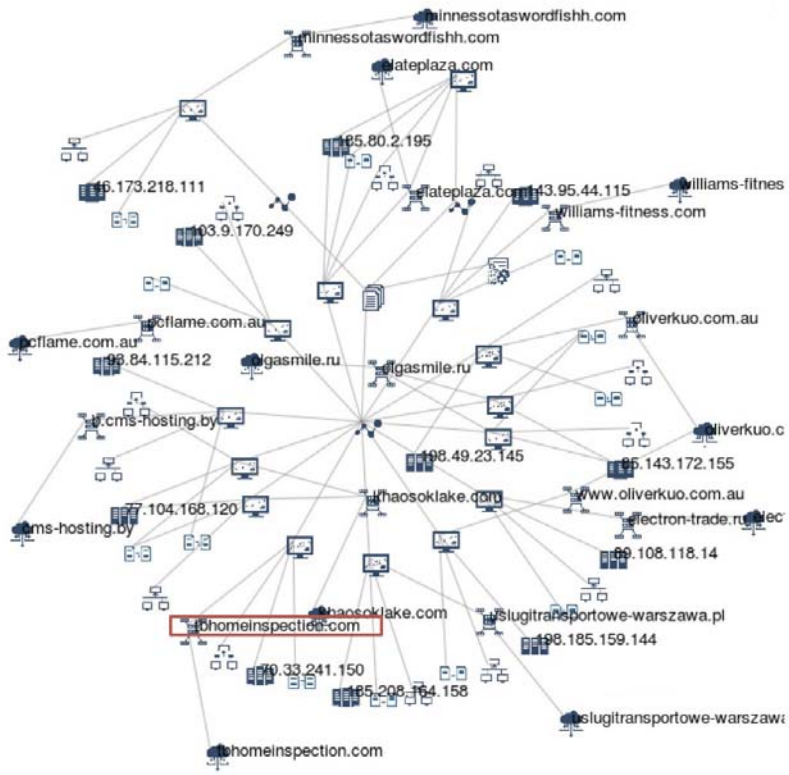

Figure 5. Example B: An example of malware distribution- send visitors to harmful websites.

TYPE	ITEM	VIRUSTOTAL
Date	2018_02_26	
Domain	tbhomeinspection.com	virustotal ratio : 11 / 67
	minnessotaswordfishh.com	virustotal ratio : 7 / 67
	williams-fitness.com	virustotal ratio : 6 / 66
	pcflame.com.au	virustotal ratio : 4 / 66
	uslugitransportowe-warszawa.pl	virustotal ratio : 4 / 66
	elateplaza.com	virustotal ratio : 9 / 66
	oliverkuo.com.au	virustotal ratio : 7 / 66
	olgasmile.ru	virustotal ratio : 1 / 67
	cms-hosting.by	virustotal ratio : 3 / 67
	electron-trade.ru	virustotal ratio : 5 / 67
	khaosoklake.com	virustotal ratio : 2 / 67
Server	70.33.241.150	
	103.9.170.249	
	89.108.118.14	
	198.49.23.145	
	77.104.168.120	
	93.84.115.212	
	85.143.172.155	
	46.173.218.111	
	143.95.44.115	
	185.80.2.195	
	198.185.159.144	
	185.208.164.158	
URL	http://uslugitransportowe-warszawa.pl/FsMflooY	
	http://elateplaza.com/FsMflooY	
	http://pcflame.com.au/FsMflooY	
	http://elateplaza.com/bg/FsMflooY	
	http://minnessotaswordfishh.com/at/FsMflooY	
	http://tbhomeinspection.com/FsMflooY	
Path	/FsMflooY	
	/at/FsMflooY	
	/bg/FsMflooY	

Figure 6. An example of malicious domain report- - send visitors to harmful websites.

Figure 7 shows example C with two domain names which were connected through the redirection of domain names and FQDNs. Through the visualization, we are able to understand that this group shares the same downloaded file (shit.exe). The detail information of example C is shown in Figure 8. Such malware distribution is like a maze with many landmines, and these domain names should be marked as high-risk websites.

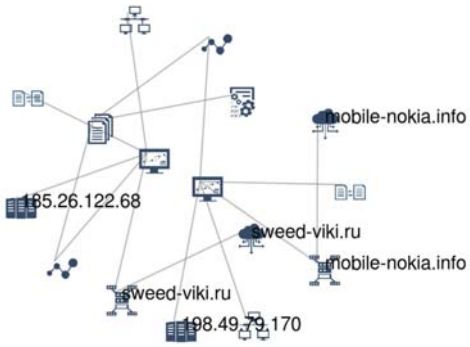

Figure 7. Example C: An example of malware distribution- Install unwanted or malicious software on visitors' computers.

TYPE	ITEM
Date	2017_05_31
Domain	mobile-nokia.info sweed-viki.ru
Server	198.49.79.170 185.26.122.68
URL	http://sweed-viki.ru/sweed/shit.exe http://mobile-nokia.info/wp-fmg/shit.exe
File Name	shit.exe
FQDN	

Figure 8. An example of malicious doamin report- install unwanted or malicious software on visitors' computers.

Based on the clustering results, we are able to extract the common features within each distinctive cluster. Through visualizing the delicate features extracted from the suspicious URLs, we can easily find the hub of malicious files, and can also visualize different malware attack pattern such as website redirection. We also measure the similarity of file name and FQDN and set up thresholds to adjust the similarity tolerance. This allows us to connect similar malicious files and similar domain names which might be machine generated. Comparing with the Nazca system, our system is able to identifies the dynamic malware distribution structure and constructs the malware distribution using less knowledge and samples. Also, our proposed system provides more information of node (like port, parameter, path and malicious type) and extend the connection rules to construct the relationship between the extended URLs and the files being downloaded.

V. CONCLUSION

The proposed method is designed for exploring more malicious domain name which often connected together in order to increase the attack success rate. We have proposed a malicious domain name analysis approach for observing different types of malicious behavior. The contribution of this paper is threefold: first, it actively visualizes the dynamic of attack pattern. Second, it constructs the ecosystem of malicious websites using less black lists. Third, it leverages open source threat intelligence to verify the threat level and the malicious category. Therefore, it is a useful verification and investigation tools for providing deeper information of malicious cluster and their common pattern. Such pattern like shared path and filename can be added as an Indicator of Compromise (IOC) of a particular threat. We have demonstrated how malicious domain name seeds can be extended and used to visualize different attack patterns. The topology of each domain clusters with detail entities are important asset and can be maintained and aggregated. In application stage, any unknown website or URL which is classified in a cluster conneted with malicious websites should be further investigated even though it is not detected by antivirus tools.

Our future work would be: (1) try other security visualization technique to better represent various malicious patterns; (2) design and implement button-up group aggregation mechanism; and (3) try clustering analysis or community analysis based on the generated clusters.

REFERENCES

[1] Alexa, the top 500 sites on the web, https://www.alexa.com/topsites

[2] D. S. Anderson, C. Fleizach, S. Savage, and G. M. Voelker, Spamscatter: Characterizing internet scam hosting infrastructure, Usenix Security, pp.1-14, 2007.

[3] I. Anton, js-crawler, https://github.com/antivanov/js-crawler.

[4] E. Bocchi, L. Grimaudo, M. Mellia, E. Baralis, S. Saha, S. Miskovic, G. Modelo-Howard, and S. J. Lee, "Network Connectivity Graph for Malicious Traffic Dissection," Proceedings of the 24th International Conference on Computer Communication and Networks (ICCCN), pp. 1-9, 2015.

[5] J. Caballero, C. Grier, C. Kreibich, and V. Paxson, , "Measuring Pay-per-Install: The Commoditization of Malware Distribution," Proceedings of Usenix security symposium, 2011.

[6] C. Curtsinger, B. Livshits, B. Zorn, and C. Seifert, "Zozzle: Low-overhead mostly static javascript malware detection," Proceedings of the Usenix Security Symposium, 2011.

[7] D. Gregory, "The everywhere war," The Geographical Journal, vol. 3, no. 177, pp. 238-250, 2011.

[8] C. Grier, L. Ballard, J. Caballero, N. Chachra, C. J. Dietrich, K. Levchenko, P. Mavrommatis, D. McCoy, A. Nappa, A. Pitsillidis, N. Provos , Z. M. Rafique , M. Rajab , C. Rossow, K. Thomas, V. Paxson, S. Savage, G. Voelker, "Manufacturing Compromise: The Emergence of Exploit-as-a-Service," Proceedings of the 2012 ACM conference on Computer and communications security, 2012, pp. 443-457.

[9] J. P. John, F. Yu, Y. Xie, A. Krishnamurthy, and M. Abadi, "deSEO: Combating Search-Result Poisoning," USENIX Security Symposium, 2011.

[10] A. Kapravelos, M. Cova, C. Kruegel, and G. Vigna, "Escape from monkey island: Evading high-interaction honeyclients," Proceedings of International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment 2011, pp. 124-143.

[11] A. Kapravelos, Y. Shoshitaishvili, M. Cova, C. Kruegel, and G. Vigna, "Revolver: An automated approach to the detection of evasive web-based malware," pp. 637-652, 2013.

[12] C. Kolbitsch, B. Livshits, B. Zorn, and C. Seifert, "Rozzle: De-cloaking internet malware," Proceedings of 2012 IEEE Symposium on Security and Privacy, pp. 443-457, 2012.

[13] L. Invernizzi and P. M. Comparetti, "Evilseed: A guided approach to finding malicious web pages," Proceedings of 2012 IEEE Symposium on Security and Privacy, pp. 428-442, 2012.

[14] L. Invernizzi, S. Miskovic, and R. Torres, C. Kruegel, S. Saha, G. Vigna, S.J. Lee, and M. Mellia, "Nazca: Detecting Malware Distribution in Large-Scale Networks," Proceedings of the Annual Network & Distributed System Security Symposium (NDSS), vol. 14, pp. 23-26, 2014.

[15] B. J. Kwon, J. Mondal, J. Jang, L. Bilge, and T. Dumitras, "The dropper effect: Insights into malware distribution with downloader graph analytics," Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 1118-1129, 2015.

[16] S. Lee and J. Kim, "WarningBird: Detecting Suspicious URLs in Twitter Stream," IEEE Transactions on Dependable and Secure Computing, vol. 12, no. 12, pp. 1-13, 2012.

[17] Z. Li, S. Alrwais, Y. Xie, F. Yu, and X. F. Wang, "Finding the linchpins of the dark web: a study on topologically dedicated hosts on malicious web infrastructures," Security and Privacy, pp. 112-126, 2013.

[18] J. Ma, L. K. Saul, S. Savage, and G. M. Voelker, "Beyond blacklists: learning to detect malicious web sites from suspicious URLs," Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining, 2009, pp. 1245-1254.

[19] N. P. P. Mavrommatis and M. A. R. F. Monrose, "All your iframes point to us," USENIX security symposium, pp. 1-16, 2008.

[20] C. Nachenberg, J. Wilhelm, A. Wright, and C. Faloutsos, "Polonium: Tera-scale graph mining and inference for malware detection," Proceedings of the 2011 SIAM International Conference on Data Mining, 2011.

[21] J. Nazario, "PhoneyC: A Virtual Client Honeypot," Proceedings of Workshop on Exploits, Malware, and Large-scale Trends (LEET), vol. 9, 2009, pp. 911-919.

[22] M. A. Rajab, L. Ballard, N. Lutz, P. Mavrommatis , and N. Provos, "CAMP: Content-Agnostic Malware Protection," Proceedings of the Annual Network & Distributed System Security Symposium (NDSS), 2013.

[23] A. Ranadive, S. Rizvi, and N. M. Daswani, Malicious advertisement detection and remediation, US Patent 8,516,590, 2013.

[24] C. Rossow, C. Dietrich, and H. Bos, "Large-scale analysis of malware downloaders," Proceedings of the International Conference on Detection of Intrusions and Malware, and Vulnerability Assessment, pp. 42-61, 2012.

[25] C. Seifert, and R. Steenson, Capture-honeypot client (capture-hpc), 2006, Available at: hhttps://projects.honeynet.org/capture-hpc.

[26] K. Shanthi, and D. Seenivasan, "Detection of botnet by analyzing network traffic flow characteristics using open source tools," Proceedings of 2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO), pp. 1-5, 2015.

[27] Stokes, J. W, R. Andersen, C. Seifert, and K. Chellapilla, "WebCop: Locating Neighborhoods of Malware on the Web," Proceedings of Workshop on Exploits, Malware, and Large-scale Trends (LEET), 2010.

[28] J. Zhang, C. Seifert, J. W. Stokes, and W. Lee, "Arrow: Generating signatures to detect drive-by downloads," Proceedings of the 20th international conference on World wide web, pp. 187-196, 2011.