Web Access Behaviour Model for Filtering Out HTTP Automated Software Accessed Domain

Manh Cong Tran
National Defense Academy
1-10-20 Hashirimizu, Yokosuka
Kanagawa, Japan, +81-46-841-3810
manhtc@gmail.com

Yasuhiro Nakamura
National Defense Academy
1-10-20 Hashirimizu, Yokosuka
Kanagawa, Japan, +81-46-841-3810
yas@nda.ac.jp

ABSTRACT

In many decades, due to fast growth of the World Wide Web, HTTP automated software/applications (auto-ware) are blooming for multiple purposes. Unfortunately, beside normal applications such as virus defining or operating system updating, auto-ware can also act as abnormal processes such as botnet, worms, virus, spywares, and advertising software (adware). Therefore, auto-ware, in a sense, consumes network bandwidth, and it might become internal security threats, auto-ware accessed domain/server also might be malicious one. Understanding about behaviour of HTTP auto-ware is beneficial for anomaly/malicious detection, the network management, traffic engineering and security. In this paper, HTTP auto-ware communication behaviour is analysed and modeled, from which a method in filtering out its domain/server is proposed. The filtered results can be used as a good resource for other security action purposes such as malicious domain/URL detection/filtering or investigation of HTTP malware from internal threats.

Categories and Subject Descriptors

K.6.5 [**Security and Protection**]: Invasive software

General Terms

Security, Experimentation, Algorithms

Keywords

HTTP automated software, cyber security, malicious detection, botnet, network security management

1. INTRODUCTION

Because of the fast growing of cyber security threats, normal users and also system administrators protect their networks by closing inward ports and permitting outgoing communication only over selected protocols such as HTTP. In many decades, Internet service developers have trend to develop their web services for reaching users. Therefore, HTTP automated softwares(auto-ware) are blooming for application update, advertising or collecting users' usage experience. Unfortunately, beside normal auto-ware, they can be also abnormal processes acting as fraudulent advertising software, virus, spyware, and malicious bots. In consequence, beside domain/server accessed by human, auto-ware accessed domain can be classified into three types: from normal software serves users' demand such as anti-virus updater, mail client, browser's toolbar; from greyware encompasses adware, spyware; from malware which is acting as HTTP-based botnet, worm or trojan horses. The distinction of normal and malicious activities and also domain/server from HTTP traffic is becoming tougher since the malicious requests merges adequately with legitimate HTTP traffic. Therefore, it is complicated to discriminate and impossible for network security devices to block malicious traffic or domain. Domains/servers which are accessed from HTTP auto-ware might be presented in a both of normal and malicious sides, such as: web services for OS updating purpose or C&C servers in transmitting command to a botnet. In consequence, filter-out auto-ware accessed domain/server are good resources for other multiple security purposes examination action such as malicious domain or URLs detection or investigation of HTTP malware from internal threats.

Web access behaviour in many researches are just considered for the improvement the effective of website design or architecture. However, understanding about auto-ware web request behaviour is also beneficial for anomaly/malicious detection, the network management, traffic engineering and security. In this paper, a model for HTTP auto-ware domain/server access behaviour is presented, based on that a network-based HTTP auto-ware accessed domain filtering is proposed. For simple, during the paper, "domain" will be used instead of "domain/server" in the meaning that is address on the Internet where user and software access by HTTP protocol.

2. RELATED WORK

In many decades, a lot of research is conducted relate to web environment or HTTP protocol. In the field of detection, Ashley has suggested a method for detecting potential HTTP C&C activity based on repeated HTTP connections to a C&C website [1]. According to this, an algorithm is proposed by for detecting HTTP polling activity. First, the enough time interval values of HTTP GET requests is determined. After that, k-means algorithm is applied to find a single cluster contains "most" data values. Finally, HTTP polling activity is pointed out by computing the standard de-

IMCOM '16, January 04-06, 2016, Danang, Viet Nam
© 2016 ACM. ISBN 978-1-4503-4142-4/16/01…$15.00
DOI: http://dx.doi.org/10.1145/2857546.2857614

viation of the cluster. If it is “small enough”, the HTTP GET requests are suspect of HTTP-based C&C traffic. Ashley has demonstrated that the method is able to detect automated polling activity to a website. However, the approach is just using the evaluation for periodic access. For that reason, the result is noticed with a caution of accuracy.

Using signature-based techniques, W. Lu et al. in [2] proposed a new hierarchical framework to automatically discover botnet on a large-scale Wi-Fi ISP network, in which the network traffic is classified into different application communities by using payload-signature. These signatures were used to separate known traffic from unknown traffic in order to decrease the false alarm rates. Like other signature-based techniques the proposed classifier is less effective as it is unable to identify new or encrypted patterns, and signature database need to be updated regularly [3].

At the side of identifying malicious domains/URLs such as by Anh Le et al in [4], lexical features are used to detect phishing and malicious URLs; or Y. Zhang et al[5], which uses a weighted sum of 8 features (4 content related, 3 lexical, and 1 WHOIS) to classify phishing URLs. However, there are many malicious auto-ware accessed domain or URLs look like as normal one and can not just use lexical features to detect. Therefore, results in this work can be used as a good supplementary method to reduce the false alarm in [4, 5]. In this paper, a method in filtering out HTTP auto-ware accessed servers/domains is proposed based on a modeling of web access behaviour. Experiment results show that it beneficial for network security management, and anomaly/malicious detection in reduce the false alarm.

3. MODEL OF ACCESS BEHAVIOUR

3.1 Observation Environment

With the goal of proposing a network-based method, for observation, analysis and experiment purposes, all web requests of a private network are collected. The network is served for about 2000 clients, and all the HTTP requests are collected through a proxy server. In the scope of experiment and evaluating the method, a set of domains which accessed by auto-ware are establish. In next sections, the features extraction and web access behaviour model is presented.

3.1.1 Feature Analysis and Extraction

For keeping the communication with their servers such as updating the information from servers or posting clients' data to servers, HTTP auto-ware repeatedly/periodically generate legal requests with the same pattern to their servers. Hence, there are many differences between auto-ware and user web access behaviour in some characteristics that are indicated as bellow:

- Malicious HTTP auto-ware will query URLs structured in a similar way, and in a similar sequence [6]. Besides that, HTTP-based malware, such as botnet, which follow the PULL style where they steadily periodically connect to their servers(e.g. C&C server) by requests with an interval in order to get the commands and updates [3]. Because of these characteristics, the length of URLs generated from HTTP malicious auto-ware and number of parameters in URLs are considered unchanged each times. Therefore, average length of URLs and average number of parameters in requests in a duration of time seem to be stable when access to their domains.

- Normal auto-ware (e.g. updater and downloader) transmits a similar periodic pattern of traffic that has been generated within a short period of time. A suspicious software does not generate bulk data transfer [7] but they still periodically sent a minor data in each request. Because of this reason, in a duration of time long enough(e.g. one day long), average amount of data sent by requests of auto-ware seem to be almost steady when access to their domains.

- On the contrary, normal users web access will not be steady each day. There is no periodically interval in requests of users to a domain.

According to these difference characteristics, a set of features to represent communication from a client of a private network to a domain is proposed. For each pair client/domain (c_i, S_j), one set $CS_{ij} = \{c_i, S_j, ul_{ij}, pa_{ij}, ds_{ij}\}$ is established to illustrate communication from client c_i to domain S_j. In that, the features are defined as bellow in a duration of dataset (e.g. one day): ul_{ij} is average length of URLs (without parameters); pa_{ij} is average number of parameters in requests; ds_{ij} is average amount of data sent by requests. Based on these features, a graph demonstrated access to S_j is established as following steps:

First, all HTTP requests connected to S_j from all clients of observed private network are collected.

Then, three above features from each client to each domain Sj are extracted. For each observed domain S_j has a set of $\{C, S_j, UL, PA, DS\}$. In that, C is set all of clients in the network which having accessed to domain S_j, UL, PA, DS is set of all values ul_{ij}, pa_{ij}, ds_{ij} respectively. A centroid point $Cen_j(aul, apa, ads)$ is defined from that C, S_j, UL, PA, S, each set values aul, apa, ads are average values of UL, PA, DS respectively.

At last, Euclidean distance is used to measure the distance between Cen_j to each $CSij = \{c_i, S_j, ul_{ij}, pa_{ij}, ds_{ij}\}$. A graph in ascending order of the sequence distance is generated which called *Access Variation Graph*(AVG).

An illustration of *Access Variation Graph* in five days data of a auto-ware's domain has been shown as in Figure 1. X axis illustrates the number of clients, and Y axis shows the distance between centroid Cen_j and $CS_{ij} = \{c_i, S_j, ul_{ij}, pa_{ij}, ds_{ij}\}$.

3.1.2 Web Access Behaviour Model

By observing in many domains in long term, AVGs of a domain which accessed by auto-ware are marked as similar each day, as can be seen in Figure 1. In the different form, domains which accessed by users, AVGs are marked difference each day. Therefore, AVG can illustrate the behaviour of a domain/server access. Based on these experiment results, the variation measurement of AVGs dissimilarity in each day is proposed as a parameter to distinguish between users and auto-ware accessed domain. The dissimilarity of AVGs(DSG) between any two days of a domain have been calculated based on the modified algorithm Hausdorff Distance, described in [8], details as bellow:

- Denoting that S_i is a domain, A and B are AVG of S_i in two days l and m respectively, in that $A =$

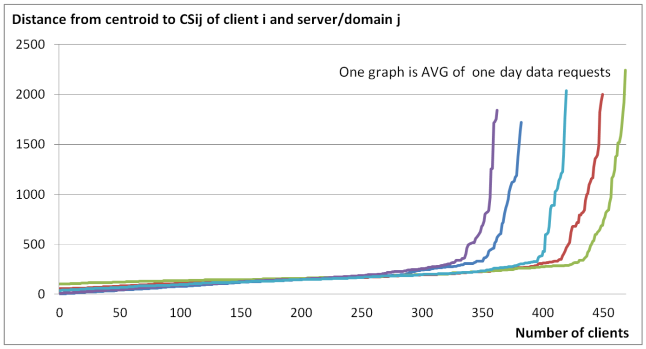

Figure 1. Access Variation Graph of an auto-ware accessed server/domain S_j in five days, one day data is presented in one graph.

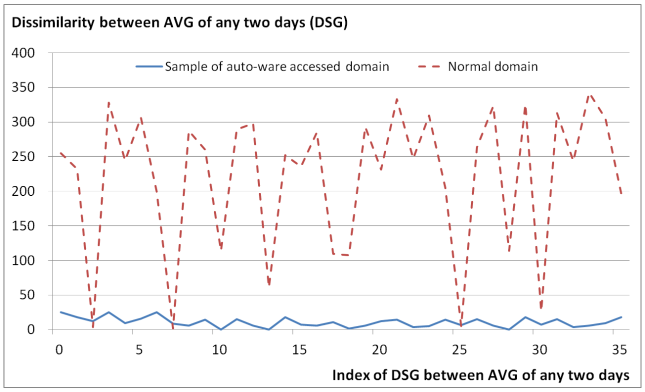

Figure 2. The dashed graph show DSGs of user accessed domain AVGs, the solid one presents DSGs of an auto-ware accessed domain AVGs.

$AVG(S_i, l) = \{x_1, ..., x_L\}$ and $B = AVG(S_i, m) = \{y_1, ..., y_M\}$.

- Defining that the distance between two points x and y is defined as Euclidean distance $d(x, y) = \|x - y\|$. From that, distance between a point x and an AVG is commonly defined as $d(x, AVG) = min_{y \in AVG|} \|x - y\|$.

- Based on the definition of distance from a point to an AVG, the distance between two graphs AVG_l and AVG_m is defined as in (1):

$$d(A, B) = \frac{1}{L} \sum_{x \in A} d(x, B) \tag{1}$$

- Based on Equation (1), dissimilarity measurement of AVG of S_i between two day l and m is determined as in Equation (2):

$$DSG(S_i, l, m) = max(d(A, B), d(A, B)) \tag{2}$$

With the goal of distinguishing between users and auto-ware accessed domain, an *Auto-ware Score* (AWS) is proposed to measure the stability of AVGs of a domain, AWS of each domain is calculated as standard deviation of that domain's DSGs. If observed data in N days, domain access

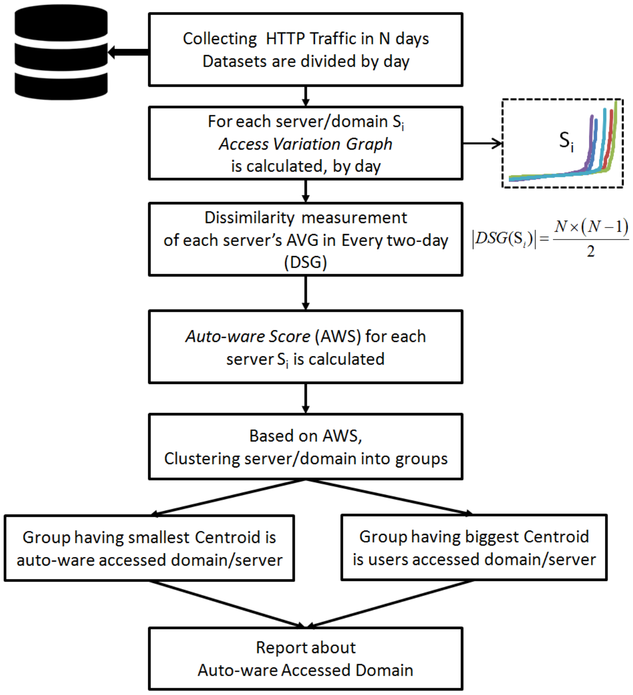
Collecting HTTP Traffic in N days
Datasets are divided by day
For each server/domain S_i
Access Variation Graph
is calculated, by day
S_i
Dissimilarity measurement
of each server's AVG in Every two-day
(DSG)
$|DSG(S_i)| = \frac{N \times (N-1)}{2}$
Auto-ware Score (AWS) for each
server S_i is calculated
Based on AWS,
Clustering server/domain into groups
Group having smallest Centroid is
auto-ware accessed domain/server
Group having biggest Centroid
is users accessed domain/server
Report about
Auto-ware Accessed Domain

Figure 3. Description of proposed method.

behaviour will present in N number of AVGs. The dissimilarity of AVGs(DSGs) between each two days is calculated and AWS is standard deviation of DSGs. A domain with a smaller AWS is higher probability of auto-wares' access domain. An example of comparison the variation in AVGs between a user and an auto-ware accessed domain is show in Figure 2. With a user accessed domain, as the dashed graph, the DSG graph shows a large flux variation, higher AWS, with many folding point. On the contrary, the DSG graph of an auto-ware accessed domain has just shown a little variation, lower AWS, as the solid graph.

4. PROPOSED METHOD

Based on the analysis of AVG and domain access behaviour in previous sections, a network-based auto-ware accessed domain filtering method is proposed. In this work, an auto-ware score is proposed for each domain, if a domain has smaller score, it will owns high probability of access by an HTTP auto-ware. General idea of method is to monitor the dissimilarity of AVG of each domain, and auto-ware score is calculated based on this dissimilarity. Steps of proposed method are illustrated in Figure 3:

First, the HTTP traffic is collected from network in N days and data will be divided in unit of day. For each domain, a set of AVG in N days will be established, based on these, $\frac{N(N-1)}{2}$ of DSGs of each domain are calculated as in Equation (2). Next, AWS of each domain is calculated as standard deviation of DSGs from previous step.

The classification of two types of domain (auto-ware and user accessed domain) is the final objective, therefore simple but effective K-Means clustering algorithm, which number of groups k = 2 is specified, is utilized. Group having smallest centroid is auto-ware accessed domain, in other side, group

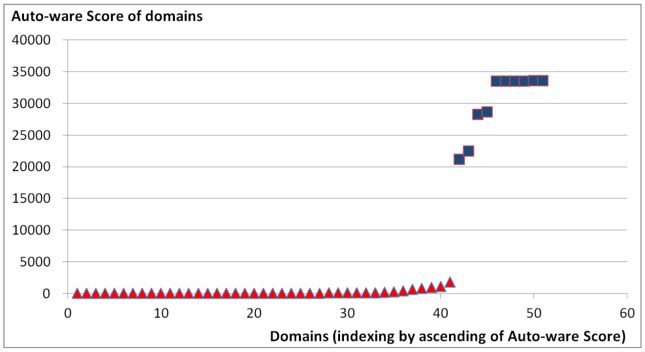

Figure 4. Domains Auto-ware Score distribution.

having biggest centroid is user accessed domain. Finally, filtering result is reported.

5. EXPERIMENT RESULTS

For experimental purpose, all requests of an organization network are captured in nine days from the 10th to 18th of December 2013. It counts about 77,567,000 requests from about 2000 clients. After using some techniques to eliminate necessary data such as a simple white list or statistic in requests, a set of 51 domains is used in applying the proposed method, in that 35 domains are in role of a C&C server, web advertising, toolbar's site etc, and they are accessed automatically from network.

By applying the proposed method in section 4, a set of AWS for 51 domains are calculated, score value is having large range from 0.18 to 33,550. The distribution of domains' AWS are demonstrated in Figure 4, in that, triangle-marker are almost auto-ware accessed domain, and rectangle-marker are belong to users accessed domain. K-means algorithm is implemented with number of groups k = 2. Clustered groups information are summarized in Table 1, accordingly, group 1 having a smaller centroid value, 171.52, so it marked as auto-ware accessed domain group, and group 2 with bigger centroid value 30183, is marked as users accessed domains group. Experiment results, overall accuracy and incorrect rate can be interpreted from Table 2. Out of 51 domains, 45 were correctly detected including 35 (group 1) as auto-ware and 10 (group 2) as user accessed domains. In total, overall accuracy rate reaches 88.24%. False detection number is 6 (group 1) out of 51 domains. They are user accessed domains but labeled as auto-ware accessed domain group, which constitutes the error rate of 11.76%.

HTTP threats not just come from malicious bot but also can be from other types of software such as bad (spyware), or greyware (adware or unauthorized peer-to-peer applications), therefore by filtering auto-ware accessed domain at network-level can be helpful for network security management since from this the non-human HTTP traffic can be measured. The results are also available used as supplementary for other method such as using lexical or contents features in malicious domain or URL detection.

6. CONCLUSIONS AND FUTURE WORK

In this paper, a study about HTTP traffic features generated from HTTP-based auto-ware is presented. From that, web access behaviour model is also proposed, based on that a score helping to filter out auto-ware accessed domains is suggested and working well with positive results in experiment. Filtered out auto-ware accessed domains are also good resources for other multiple security purposes examination action such as malicious domain detection or investigation of HTTP malware from internal of a network. The continuous improvement of method is due to reduce the error rate. Besides that, other valuable features need to be considered in attempting to classify which are malicious domains from the current results.

Table 1. Clustering based on AWS using K-Means

Group No	Centroid Value	Domain Number	Group Label
1	171.52	41	Auto-ware
2	30183	10	User

Table 2. Experiment results

Group (No)	True Positive		False Detection	
	Number	%	Number	%
Auto-ware (1)	35	85.37	6	16.67
User (2)	10	100%	0	0

7. REFERENCES

[1] D. Ashley. An algorithm for http bot detection. In *Technical Report*, pages 1–24. University of Texas at Austin - Information Security Office, January 2011.

[2] W. Lu, M. Tavallaee, and A. A. Ghorbani. Automatic discovery of botnet communities on large-scale communication networks. In *Proceedings of the 4th International Symposium on Information, Computer, and Communications Security*, ASIACCS '09, pages 1–10, New York, NY, USA, 2009. ACM.

[3] M. Eslahi, H. Hashim, and N. Tahir. An efficient false alarm reduction approach in http-based botnet detection. In *Proceedings of IEEE Symposium on Computers & Informatics*, ISCI '13, pages 201–205. IEEE, April 2013.

[4] A. Le and M. Faloutsos. Phishdef: Url names say it all. In *Proceedings of Infocom*, INFOCOM '11, pages 191–195. IEEE, 10-11 April 2011.

[5] Y. Zhang, J. I. Hong, and L. F. Cranor. Cantina: A content-based approach to detecting phishing web sites. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, pages 639–648, New York, NY, USA, 2007. ACM.

[6] R. Perdisci, W. Lee, and N. Feamster. Behavioral clustering of http-based malware and signature generation using malicious network traces. In *Proceedings of the 7th USENIX Conference on Networked Systems Design and Implementation*, NSDI '10, pages 26–26, Berkeley, CA, USA, 2010. USENIX.

[7] W. T. Strayer, R. Walsh, C. Livadas, and D. Lapsley. Detecting botnets with tight command and control. In *Proceedings of the 31st IEEE Conference on Local Computer Networks*, pages 195–202. IEEE, Nov 2006.

[8] M.-P. Dubuisson and A. Jain. A modified hausdorff distance for object matching. In *Proceedings of the International Conference on Pattern Recognition*, IAPR '94, pages 566–568. IEEE, Oct 1994.