Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 5 (2011) 123–131

Procedia Computer Science

The 2nd International Conference on Ambient Systems, Networks and Technologies (ANT-2011)

Unsupervised Clustering of Web Sessions to Detect Malicious and Non-malicious Website Users

Dusan Stevanovic*, Natalija Vlajic, Aijun An

Department of Computer Science and Engineering, York University, 4700 Keele St.,Toronto, Ontario, M3J 1P3, Canada

Abstract

Denial of Service (DoS) attacks are recognized as one of the most damaging attacks on the Internet security today. Recently, malicious web crawlers have been used to execute automated DoS attacks on web sites across the WWW. In this study, we examine the use of two unsupervised neural network (NN) learning algorithms for the purpose web-log analysis: the Self-Organizing Map (SOM) and Modified Adaptive Resonance Theory 2 (Modified ART2). In particular, through the use of SOM and Modified ART2, our work aims to obtain a better insight into the types and distribution of visitors to a public web-site based on their link-traversal behaviour, as well as to investigate the relative differences and/or similarities between malicious web crawlers and other non-malicious visitor groups. The results of our study show that, even though there is a pretty clear separation between malicious web-crawlers and other visitor groups, around 8% of malicious crawlers exhibit very 'human-like' browsing behaviour and as such pose a particular challenge for future web-site security systems.

Keywords: Web Crawler Detection; Neural Networks; Web Server Access Logs; Machine Learning; Clustering; Denial of Service;

1. Introduction

The today's business world is critically dependent on the availability of Internet. For instance, the phenomenal growth and success of Internet has transformed the way traditional essential services, such as banking, transportation, medicine, education and defence, are operated. In ever increasing numbers, these services are being offered by means of Web-based applications. However, the inherent vulnerabilities of the Internet architecture provide opportunities for various attacks on the security of Web-based applications. Distributed denial-of-service (DDoS) is an especially potent type of security attack, capable of severely degrading the response-rate and quality at which Web-based services are offered. According to the United States' Department of Defence report from 2008, presented in [1], the number of cyber attacks (including the DDoS attacks) from individuals and countries, targeting economic, political and military organizations, are expected to increase in the future and cost billions of dollars.

The most common way of conducting a denial of service (DoS) attack is by sending a flood of messages to the target (e.g., a machine hosting a web site) with the aim to interfere with the target's operation, and make it hang, crash, reboot, or do useless work. In the past, most DoS attacks were single-sourced, which means they were

* Corresponding author. Tel.: 1 416 736 2100 x70143
E-mail address: dusan@cse.yorku.ca.

Selection and/or peer-review under responsibility of Prof. Elhadi Shakshuki and Prof. Muhammad Younas.
doi:10.1016/j.procs.2011.07.018

reasonably easy to prevent by locating and disabling the source of the malicious traffic. Nowadays, however, almost all DoS attacks involve a complex, distributed network of attacking machines - comprising hundreds to tens of thousands of hijacked zombies. These, the so-called Distributed DoS (or DDoS) attacks, are extremely difficult to detect due to the sheer number of hosts participating in the attack. At the same time, they can generate enormous amount of traffic towards the victim and result in substantial loss of service and revenue for businesses under the attack.

An emerging (and increasingly more prevalent) type of DDoS attacks, known as *application-layer* or *layer-7* attacks [2], are shown to be particularly challenging to detect. The reasons for this are: 1) in an application-layer attack, the attacker utilizes a legitimate-looking layer-7 network session, and 2) HTML requests sent to a web server are often constructed in a way that mimics a semi-random walk through the web site links, and thus appears as a web site traversal conducted by a legitimate human user. Given the fact that application-layer DDoS attacks resemble the legitimate traffic, it is quite challenging not only to defend against these attacks but also to construct an effective metric for their detection.

So far, a number of studies on the topic of application-layer DDoS attacks have been reported. Thematically, these studies can be grouped into two main categories: 1) detection of application-layer DDoS attacks during a *flash crowd* event based on aggregate-traffic analysis ([3] and [4]) and 2) differentiation between well-behaved and malicious web crawlers[†] based on web-log analysis ([5], [6] and [7]).

The study presented in this paper falls in the latter of the above mentioned categories, as we examine the use of two unsupervised neural network (NN) learning algorithms for the purpose web-log analysis: the Self-Organizing Map (SOM) [8] and Modified Adaptive Resonance Theory 2 (Modified ART2)[‡] [9]. In particular, through the use of SOM and ART2, our work aims to obtain a better insight into the types and distribution of visitors to a public web-site based on their link-traversal behaviour, as well as to investigate the relative differences and/or similarities between malicious web crawlers and other non-malicious visitor groups.

We have chosen to use the SOM and ART algorithms in our study for the following reasons. The SOM algorithm is very well known for its ability to produce natural clustering, i.e. clustering that is robust to statistical anomalies. Furthermore, unlike other clustering algorithms, the SOM algorithm achieves superior visualisation of high-dimensional input data in 2D-representation space. The ART2 algorithm, on the other hand, is based on the learning paradigm known as 'stability-plasticity dilemma', where the exposure to new training data does not destroy previously learned information – regardless of the statistical representation of different dataset groups. Consequently, ART2 has a unique ability to identify statistically underrepresented but significant clusters, and is greatly suited for imbalanced datasets. Also, by employing unsupervised learning, the process that labels sessions without previous *a priori* knowledge, we can discover unbiased sessions distributions in the underlying data.

The content of this paper is organized as follows: In Section 2, we discuss previous works on web crawler detection. In Section 3, we give an overview of our web-log analyzer that is used to generate a meaningful training dataset out of any given access log file. In Section 4, we briefly outline our experimentation setup. In Section 5, we present and discuss the obtained web-session clustering results. In Section 6, we conclude the paper with our final remarks.

2. Related work

So far, several research studies have looked at the use of supervised learning for the purposes of data-mining and/or clustering of web sessions. Note that supervised learning process clusters sessions based on previous a priori knowledge. In one of the first such studies [10], the authors attempt to discover distinct groups of web robot sessions by applying C.4.5 algorithm (i.e. a decision tree classifier) to 25-dimensional feature vector space. The 25 features,

---

[†] *Web-crawlers* are programs that traverse the Internet autonomously, starting from a *seed* list of web pages and then recursively visiting documents accessible from that list. Crawlers are also referred to as *robots (bots)*, *wanderers*, *spiders*, or *harvesters*. Their primary purpose is to discover and retrieve content and knowledge from the Web on behalf of various Web-based systems and services. For instance, search-engine crawlers seek to harvest as much Web content as possible on a regular basis, in order to build and maintain large search indexes. On the other hand, shopping bots crawl the Web to compare prices and products sold by different e-commerce sites. Malicious crawlers are type of web robots that, for instance, generate DDoS traffic that can overwhelm web server's resources and thus limit or unable legitimate users' access to the website. Another example of malicious activity attributed to malicious crawlers is collecting email addresses for spam mail.

[‡] Modified ART2 is a variation of the original ART algorithm [18]. Its advantages over the original algorithm are: 1) stable learning that results in gradually increasing/merging clusters, and 2) learning/clustering that can be terminated either when the radius of the formed clusters reaches some predetermined size, or when the number of formed clusters reaches some predetermined number.

i.e. their respective values, are derived from the navigational properties of each identified robot session. In advance of clustering, and depending on the value of *user-agent fields*, each session is pre-labelled as known *robots*, *known browsers*, *possible robots*, and *possible browsers*. The results of the study show that, by applying the proposed feature set in combination with C.4.5 algorithm, robots can be detected with more than 90% accuracy after only four web-page requests. In [11], the authors utilize supervised Bayesian classifier to detect the presence of web crawlers from web server logs and, subsequently, they compare their results to the results obtained with the decision tree technique. The proposed methodology achieves very high recall and precision values in web robot detection. Another study utilizing logistic regression and decision trees has been reported in [6]. In this study, authors propose a robot detection tool that speeds up the tasks for pre-processing web server access logs and achieves very accurate web robot detection.

Several studies have looked at the use of unsupervised learning for the purpose of more general web log analysis. In [12], the authors employ the SOM algorithm to achieve automatic demographic-based classification of web-site visitors based on the number and sequence of their web-page visits. In [13], the authors also examine the application of the SOM algorithm on web-server access logs, with the aim to group web-visitors thematically and, as a result of that, help them find relevant information in a shorter period of time. In a similar study [14], the authors propose employing the ART algorithms to cluster web users according to their thematic interests.

In the view of the previous works, the novelty of our research is twofold. Firstly, to the best of our knowledge, this is the first study that applies unsupervised learning to the problem of web-visitor categorization, ultimately aiming to promote effective differentiation between malicious web-crawlers and other (non-malicious) visitor groups to a web site. (Note, in [12], [13], and [14], only human web-visitors have been considered, and little to no attention has been given to automated web-crawlers.) Secondly, this is the first study that attempts to examine the actual, qualitative differences between malicious web-crawlers and other non-malicious crawler types, such as Googlebot and MSNbot, by applying the SOM-based data visualisation methods.

3. Pre-processing of server logs

In our study, a Java-based log analyzer has been utilized to pre-process the web server access-log files. A typical web server access log file includes the information such as the IP address/host name of the site visitor, the URL of requested page, the date and time of the request, the size of the data requested and the HTTP method of request. Additionally, the log contains the user agent string describing the hardware and/or software the visitor was using to access the site, and the referrer field which specifies the web page by which the client reached the current page.

On each provided access log file, our log analyzer performs the following: 1) scans the entries in the log to identify unique visitor sessions, and 2) for each identifies session, the analyzer examines its key features to generate the sessions' 9-dimensional feature-vector representation.

In the reminder of this section, we provide a detailed description of the above mentioned processes – session identification and generation of sessions' feature-vector representations – as performed by our log analyzer. (Note, the aggregate of the obtained feature-vectors represents the actual training data-set that is to be fed into the SOM and Modified ART2 algorithm.) We close this section with the description of data-set labelling process – a step that needs to be performed in order be able to comprehend and validate the results of the clustering process.

3.1. Session identification

Session identification is the process of dividing a server access log into sessions. Typically, session identification is performed by: 1) grouping of all HTTP requests that originate from the same IP address and are described by the same user-agent string, and 2) by applying a timeout approach to break this grouping into unique sessions. Therefore, a session is defined as a sequence of requests coming from the same IP address (and is described by the same user-agent string) and where the time-lapse between any two consecutive HTTP requests in the sequence is within a pre-defined threshold. The key challenge of session identification is to determine the proper value of the given threshold, as different Web users exhibit different navigational behaviour. In this study, we employ a 30-minute threshold, because it has generated fairly successful web crawler classification results in the past (see [11]).

3.2. Features

From previous studies on web session analysis, namely [6], [10] and [11], we have adopted seven different features that are shown to be useful in identifying and distinguishing between automated and human visitors to a web site. These features are enlisted below:

1. Click rate – a *numerical* attribute calculated as the number of HTTP requests sent by a user in a single session. The click rate metric appears to be useful in detecting the presence of the web crawlers because higher click rate can only be achieved by an automated script (such as a web robot) and is usually very low for a human visitor.
2. HTML-to-Image Ratio – a *numerical* attribute calculated as the number of HTML page requests over the number of image file (JPEG and PNG) requests sent in a single session. Web crawlers generally request mostly HTML pages and ignore images on the site which implies that HTML-to-Image ratio would be higher for web crawlers than for human users.
3. Percentage of PDF/PS file requests – a *numerical* attribute calculated as the percentage of PDF/PS file requests sent in a single session. In contrast to image requests, some crawlers, tend to have a higher percentage of PDF/PS requests than human visitors.
4. Percentage of 4xx error responses – a *numerical* attribute calculated as the percentage of erroneous HTTP requests sent in a single session. Crawlers typically would have higher rate of erroneous request since they have higher chance of requesting outdated or deleted pages.
5. Percentage of HTTP requests of type HEAD – a *numerical* attribute calculated as percentage of requests of HTTP type HEAD sent in a single session. Most web crawlers, in order to reduce the amount of data requested from a site, employ the HEAD method when requesting a web page. On the other hand, requests coming from a human user browsing a web site via browsers are, by default, of type GET.
6. Percentage of requests with unassigned referrers – a *numerical* attribute calculated as the percentage of blank or unassigned referrer fields set by a user in a single session. Typically, web crawlers would initiate HTTP requests with unassigned referrer field.
7. 'Robots.txt' file request – a *nominal* attribute with values of either 1 or 0, indicating whether 'robots.txt' file was requested or not requested by a user during a session, respectively. Web administrators, through the Robots Exclusion Protocol, use a special-format file called *robots.txt* to indicate to visiting robots which parts of their sites should not be visited by the robot. For example, when a robot visits a Web-site, say http://www.cse.yorku.ca, it should first check for http://www.cse.yorku.ca/robots.txt. It is unlikely, that any human would check for this file, since there is no link from the website to this file, nor are (most) users aware of its existence.

As mentioned earlier, features 1-7 have been used in the past for distinguishing between human- and robot-initiated sessions. However, based on the recommendations and discussion presented in [15], we introduce two novel features for characterization of web-browsing sessions:

8. Standard deviation of requested page's depth – a *numerical* attribute calculated as the standard deviation of page depth across all requests sent in a single session. For instance, we assign a depth of three to a web page '/cshome/courses/index.html' and a depth of two to a web page '/cshome/calendar.html'.
9. Percentage of consecutive sequential HTTP requests – a *numerical* attribute calculated as the number of sequential requests for pages belonging to the same web directory and generated during a single user session. For instance, a series of requests for web pages matching pattern '/cshome/course/*.* will be marked as consecutive sequential HTTP requests. However, a request to web page '/cshome/index.html' followed by a request to a web page 'cshome/courses/index.html' will not be marked as consecutive sequential requests.

The importance of features 8 and 9 can be explained as follows. In a typical web-browsing session, humans are set to find information of interest by following a series of thematically correlated and progressively more specific links. Loops may also be present if a human becomes disoriented during their visit. In contrast, robots are neither expected to have such complex navigational patterns, nor would they be restricted by the link structure of the web site. After an initial crawl of a site, robots are capable of learning precisely where the information that they are seeking resides, so that on repeated visits they may only send requests for specific files or restrict their crawling to specific areas of the site. For the above reasons, the standard deviation of requested pages' depths, i.e. attribute 8, should be low for web robot sessions since a web robot should scan over a narrower directory structure of a web site than a human user.

Also the number of resources requested in a single session is another distinction between robot and human traffic that is not expected to change over time. This distinction arises because human users retrieve information from the Web via some interface, such as a web browser. This interface forces the user's session to request additional resources automatically. Most Web browsers, for example, retrieve the HTML page, parse through it, and then send a barrage of requests to the server for embedded resources on the page such as images, videos, and client side scripts to execute. Thus, the temporal resource request patterns of human visitors are best represented as short bursts of a large volume of requests followed by a period of little activity. In contrast, web robots are able to make their own decisions about what resources linked on an HTML page to request, and may choose to execute the scripts available on a site only if they have the capacity to do so. For the above reasons, it is reasonable to expect that the number of consecutive sequential HTTP requests would be relatively high in human user sessions and relatively low in web robot sessions.

3.3. Dataset labelling

Once the training data-set (comprising feature-vector representations) is generated, the log analyzer labels each feature-vector as belonging to one of the following 4 categories: *human visitors*, *well-behaved web crawlers*, *malicious crawlers* and *unknown visitors*. The goal of data labelling is to facilitate our understanding and validation of the results that are to be obtained by the actual clustering process. Namely, through quick association of feature-vectors corresponding to a cluster with their pre-assigned labels, we hope to be able to obtain a better understanding of the cluster's nature and significance.

The labelling of feature vectors is performed as follows:

1. Any feature vector that corresponds to a web session whose user agent string matches a known browser and does not access the 'robots.txt' file is labelled as *human visitors*.
2. Any feature vector that corresponds to a web session whose user agent string matches a known well-behaved web crawler is labelled as *well-behaved web crawlers*.
3. Any feature vector that corresponds to a web session whose user agent string matches a known malicious web crawler is placed in a cluster of *malicious web crawlers*. Also any web session whose user agent string matches a known browser's user agent string and accesses the 'robots.txt' file is also placed in a cluster of *malicious web crawlers*. Additionally, any unknown session that neither belongs to a well-behaved web crawler or malicious crawler but accesses the 'robots.txt' file is also placed in a cluster of *malicious web crawlers*.
4. All other web sessions are labelled as *unknown visitors*.

(Note, the log analyzer maintains a table of user agent fields of all known, malicious or well-behaved, web crawlers. This table can be built from the data found on web sites [16] and [17]. The web sites also maintain the list of various browsers' user agent strings that can be used to identify human visitors to the site as well.)

4. Experimental design

4.1. Training data

In the experimental stage of our study, the training data sets were constructed by pre-processing web server access log files provided by York CSE department. The log file stores detailed information about user web-based access into the domain www.cse.yorku.ca during a 4-week interval - between mid January 2011 and mid February 2011. There are a total of about 3 million log entries in the file.

Since we are investigating the behaviour as evident from the click-stream of a user-agent, it is fair to assume that

Table 1. Class distribution in the dataset

	Number of Sessions
Total #	55920
Total # of Human Sessions	51252
Total # of Well-behaved Crawler Sessions	1391
Total # of Malicious Crawler Sessions	1020
Total # of Unknown Visitor Sessions	2257

any session with less than 5 requests in total, is too short to enable labelling, and therefore is ignored in our analysis. Table 1 lists the number of sessions and class label distributions generated by the log analyzer.

4.2. Clustering algorithms

The detection of web crawlers was evaluated with the following two unsupervised neural network algorithms: SOM and Modified ART2. The implementation of SOM algorithm is provided within MATLAB as a part of Neural Network Toolbox software package. We have chosen a SOM comprising 100 neurons in 10-by-10 hexagonal arrangement. The map was trained with 200 epochs. The Modified ART2 implementation was based on the pseudo-code outlined in [9]. The algorithm was executed with $\rho_{max} = 1.4$, $\Delta\rho = 0.1$ and $n_{max} = 6$. All input vectors were normalized prior to being fed to SOM and Modified ART2.

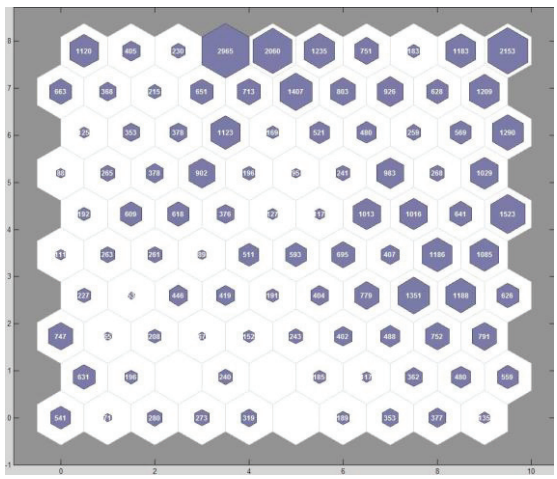
5. Clustering results

5.1. SOM results

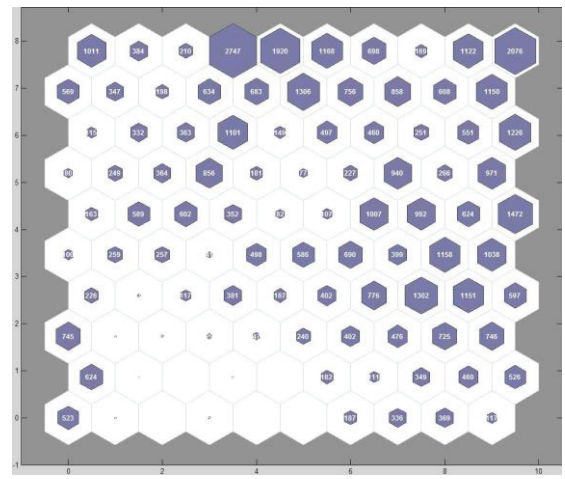
Figure 1 displays the results of dataset clustering obtained with a 10-by-10 neuron SOM. On each of the shown maps, the size of the blue region inside a neuron's hexagon depicts the number of session hits for that neuron, i.e. number of sessions whose feature vectors end up firing the (same) given neuron. The exact number of a neuron's session hits is also explicitly provided within the neuron's hexagon region.

The map in Figure 1.a) shows the neuron hits for all sessions, and thus helps us visualise the actual distribution of the training dataset (i.e. helps us get an idea about the number, size and spatial proximity of the dataset's most dominant clusters). Figures 1.b) to 1.e) show the neuron hits for sessions that were pre-labelled as belonging to human, well-behaved crawler, malicious crawler and unknown visitors, respectively. From the obtained maps, the following interesting conclusions can be drawn:

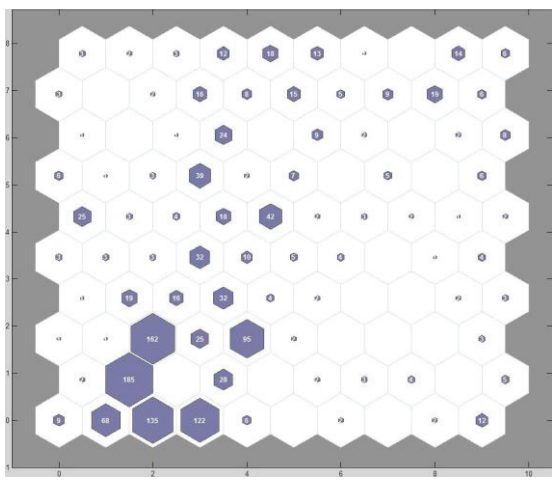
- *Human vs. crawlers sessions*: Based on the distribution of fired neurons in Figure 1.b), 1.c) and 1.d), there appears to be a pretty good separation between human visitor sessions and web-crawler sessions (both malicious and well-behaved). Namely, while crawler sessions are almost exclusively associated with neurons in the lower left corner of the map, human sessions are spread over a large area of the map, with most human sessions firing the neurons in the upper right corner of the map. It might be worth pointing out that the large spread of fired neurons in the map of Figure 1.b) is <u>not</u> an indicator of greater variability in humans sessions compared to other session groups. Instead, it is the result of the statistical dominance of training-data corresponding to human sessions - see Table 1. (As indicated in the introduction, the SOM algorithm produces results that are dependent on the input data density; hence, data clusters with higher density tend to 'win-over' a larger number of SOM neurons, regardless of their inter-cluster variance.)
- *Sessions that are labeled as human but 'behave' like malicious crawlers:* A detailed inspection of Figures 1.b) and 1.d) reveals that, in spite of the well-formed separation between human and malicious web-crawlers, a percentage of sessions/visitors that declare themselves as regular (human) visitors end up firing neurons in the region (or close to the region) dominated by malicious web-crawlers – lower left corner of the map. This observation raises the question whether those sessions, in fact, correspond to malicious crawlers whose aim is to bypass web-site security by purposely falsifying the value of user agent string field. Recall, user agent string appears as a parameter in HTML requests, and can be (relatively) easily altered.



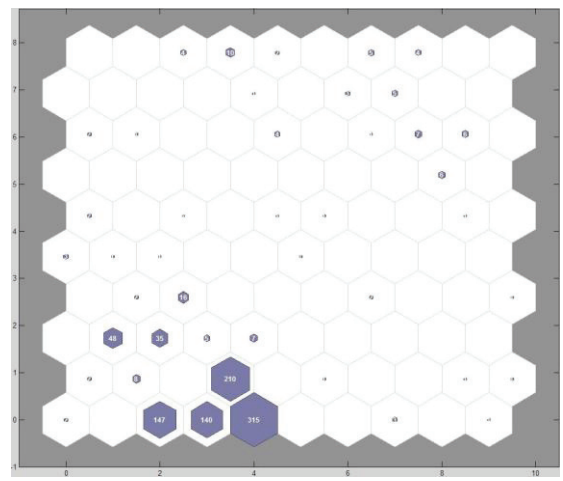
1.a) *All* sessions neuron hits



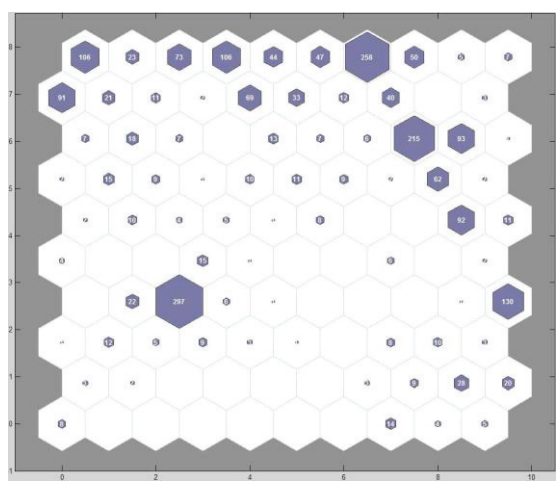
1.b) *Human Visitor* sessions hits



1.c) *Well-behaved Crawler* sessions hits



1.d) *Malicious Crawler* sessions hits



1.e) *Unknown Visitor* sessions associations with neurons

Fig. 1. Session Hits per Neuron visualized in 2 dimensional SOM Map

- *Sessions that are labeled as malicious crawlers but 'behave' like humans:* A detailed inspection of Figures 1.b) and 1.d) also reveals that a number of sessions/visitors that are identified as malicious crawlers end up firing neurons in the region dominated by human generated sessions – upper right corner of the map. (It is reasonable to assume that these visitors are indeed malicious crawlers, as it is unlikely that any regular human visitor would change the value of its agent string into 'malicious crawler', thus risking to be blocked by the web-site.) Accordingly, this observation implies that the behavior of some malicious crawlers - those that fire the nodes in the upper right corner - is very similar to the behavior of regular users. It should be obvious that such malicious crawlers are potentially very dangerous. Namely, had they attempted to falsify the value of agent string (i.e. declare themselves as regular visitors), they would have 'perfectly' blended into the population of regular human visitors, and would be very hard to detect by the web-site's security system.
- *Unknown visitor sessions*: As explained in Section 3.3, unknown visitor sessions are sessions whose user agent strings are not known and thus are not enlisted on [16] and [17]. By comparing Figures 1.b and 1.e), it is interesting to observe that there is a significant overlap between fired neurons in the respective maps. This leads us to conclude that most unknown sessions are likely generated by regular human users, i.e. are likely non-malicious by their behavior and intent.

5.2. ART2 results

Figure 2 displays the results of dataset clustering using Modified ART2. The plot displays the ratio of each session type (human, well-behaved crawler, malicious crawler and unknown visitor) per cluster placement. A session is placed in cluster *i,* if its 9-dimensional vector representation is the closest (measured in the Euclidean distance) to the center of cluster *i* among all other clusters. The plot displays the sample results when Modified ART2 algorithm generates 6 clusters of sessions.

While the results obtained with SOM are useful for obtaining information about the spatial distribution, i.e. proximity, of data clusters, Modified ART2 gives us an insight into the inter-cluster variance. (As indicated in the introduction, Modified ART2 creates equal-size clusters and is not influenced by statistical irregularities in the training dataset.) With this in mind, and by expecting Figure 2, we derive the following conclusions:

- *Human sessions*: Nearly 96% of human sessions fall into cluster 1, thus suggesting a very small variance of this cluster group. In practical terms, this implies that most human web users follow a very similar web browsing pattern.
- *Unknown sessions*: 95% of unknown sessions belong to the same cluster as human sessions. This confirms our hypothesis from section 5.1, that most unknown sessions are in fact human-generated.
- *Malicious web-crawler sessions*: Out of all session groups, malicious crawlers exhibit the greatest variability – they are spread over all 6 formed clusters, with most being assigned to cluster 2. It is interesting to observe that nearly 8% of malicious web-crawler sessions are assigned to cluster 1, together with human visitors. This again confirms our earlier hypothesis, that some malicious web crawlers behave very-much like regular users, and in the case of a falsified user agent string value their detection would have present a particular challenge.

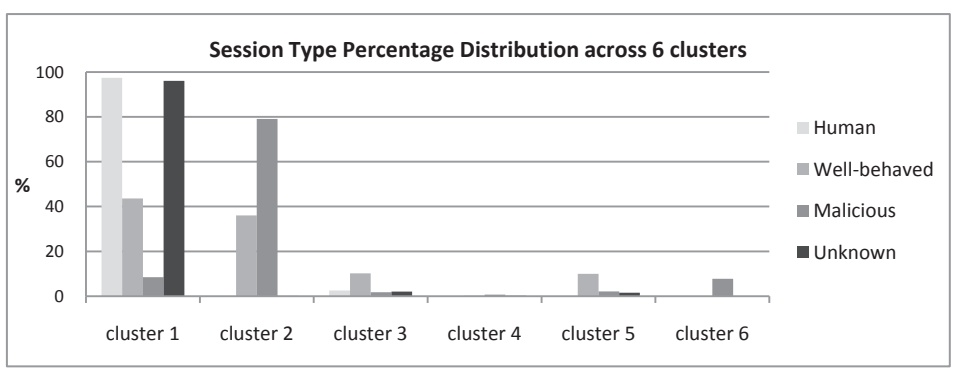


Fig. 2. Session Type Percentage Distribution across 6 Clusters

6. Conclusion and final remarks

The detection of malicious web crawlers is one of the most active research areas in network security. In this paper, we approach the problem of malicious web-crawler detection through the use of unsupervised neural learning.

The following important conclusions were derived from our study:

There exists a pretty good separation between malicious and non-malicious web users in terms of their browsing behaviour. And, while human visitors tend to follow rather similar browsing patterns (i.e. human visitors exhibit similar web browsing characteristics in terms of the average values of the 9 features), malicious web crawlers exhibit a range of browsing strategies (i.e. malicious sessions are spread over greater number of clusters in Figure 2). Moreover, nearly 8% of web crawlers exhibit very much 'human-like' browsing behaviour. With a higher level of sophistications, these crawlers could pose a serious challenge for future web-site security systems.

The results presented in our study do not provide insights on how to develop detection signatures for malicious bots. However, given the clear separation between malicious and non-malicious web sessions, network security personnel can employ the mean and variance values of the 9 features as guidance in building signatures that can detect malicious crawlers. We plan to analyze and present the exact values of these metrics in a future journal paper.

References

1. C. Wilson, Botnets, Cybercrime, and Cyberterrorism: Vulnerabilities and Policy Issues for Congress, Foreign Affairs, Defense, and Trade Division, United States Governemnt, CRS Report for Congress, 2008.
2. Prolexic Technologies, Evolving Botnet Capabilities - and What This Means for DDoS, White Paper, 2010.
3. Y. Xie and S.-Z. Yu, Monitoring the Application-Layer DDoS Attacks for Popular Websites, IEEE/ACM Transactions on Networking, vol. 17, no. 1, pp. 15-25, Feb. 2009.
4. G. Oikonomou and J. Mirkovic, Modeling Human Behavior for Defense against Flash-Crowd Attacks, in In Proceedings of IEEE International Conference on Communications, Dresden, Germany, 2009, pp. 1-6.
5. P. Hayati, V. Potdar, K. Chai, and A. Talevski, Web spambot detection based on web navigation behaviour, in International Conference on Advanced Information Networking and Applications, Perth, Australia, 2010, pp. 797-803.
6. C. Bomhardt, W. Gaul, and L. Schmidt-Thieme, Web Robot Detection - Preprocessing Web Logfiles for Robot Detection, in In Proc. SISCLADAG, Bologna, Italy, 2005.
7. K. Park, V. Pai, K. Lee, and S. Calo, Securing Web Service by Automatic Robot Detection, in Proceedings of the annual conference on USENIX '06 Annual Technical Conference, Berkeley, CA, 2006, pp. 23-29.
8. T. Kohonen, Self-Organizing Maps, 3rd ed. New York: Springer-Verlag, Berlin Heidelberg, 2001.
9. N. Vlajic and H. C. Card, Vector quantization of images using modified adaptive resonance algorithm for hierarchical clustering, IEEE Transactions on Neural Networks, vol. 12, no. 5, pp. 1147-1162, Sep. 2001.
10. P. N. Tan and V. Kumar, Patterns, Discovery of Web Robot Sessions Based on their Navigation, Data Mining and Knowledge Discovery, vol. 6, no. 1, pp. 9-35, Jan. 2002.
11. A. Stassopoulou and M. D. Dikaiakos, Web robot detection: A probabilistic reasoning approach, Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 53, no. 3, pp. 265-278, Feb. 2009.
12. Y. Hiltunen and M. Lappalainen, Automated Personalization of Internet Users Using Self-Organizing Maps, in IDEAL, Manchester, UK, 2002, pp. 31-34.
13. D. Petrilis and C. Halatsis, Two-level Clustering of Web Sites Using Self-Organizing Maps, Neural Process Letters, vol. 27, no. 1, pp. 85-95, Feb. 2008.
14. J. Martín-Guerrero, E. Soria-Olivas, P. J. G. Lisboa, A. Palomares, and E. Balaguer-Ballester, User Profiling from Citizen Web Portal Accesses using the Adaptive Resonance Theory Neural Network, in IADIS, San Sabastian, Spain, 2006, pp. 334-337.
15. D. Doran and S. S. Gokhale, Web robot detection techniques: overview and limitations, Data Mining and Knowledge Discovery, pp. 1-28, Jun. 2010.
16. User-Agents.org. [Online]. http://www.user-agents.org. (2011, Jan.)
17. Bots vs. Browsers. [Online]. http://www.botsvsbrowsers.com/. (2011, Jan.)
18. G. A. Carpenter and S. Grossberg, Adaptive Resonance Theory, in The handbook of brain theory and neural networks , M. A. Arbib, Ed. Cambridge, MA, USA: MIT Press, 1998, pp. 79-82.