An Integrated Approach to Defence Against Degrading Application-Layer DDoS Attacks

Dusan Stevanovic and Natalija Vlajic
Department of Computer Science and Engineering
York University
Toronto, Canada
dusan@cse.yorku.ca, vlajic@cse.yorku.ca

Abstract—Application layer Distributed Denial of Service (DDoS) attacks are recognized as one of the most damaging attacks on the Internet security today. In our recent work [1], we have shown that *unsupervised machine learning* can be effectively utilized in the process of distinguishing between regular (human) and automated (web/botnet crawler) visitors to a web site. We have also shown that with a slightly higher level of sophistication in the design of some web/botnet crawlers, their detection could become particularly challenging, requiring additional vigilance and investigation on the part of the site's defense team. In this paper, we demonstrate an application of time series analysis in order to perform a further fine-tuned detection of suspicious visitors to a web site. Additionally, we propose a novel application-layer DDoS detection system that integrates the use of our combined unsupervised learning and time-domain web-visitor classifier with the use of standardized challenge-response tests. The system is aimed to ensure reliable detection of malicious (web/botnet crawler) visitors to a web site while being minimally intrusive towards regular (human) visitors.

Keywords—system security; distributed denial of service, DDoS detection and prevention, browsing behavior model

I. Introduction

Many of the traditional essential services, such as banking, transportation, medicine, government, education and defence, are increasingly offered by means of Web-based applications. Unfortunately, the inherent vulnerabilities of the Internet architecture provide opportunities for various attacks on the availability of these applications. Distributed Denial-of-Service (DDoS) is an especially potent type of attack on Web availability, capable of severely degrading the response-rate and quality at which Web-based services are offered. Given the scale of their potential implications on both the US industry and government, the FBI has recently identified cyber attacks - including DDoS attacks - as the fastest growing national security threat [2].

The most common way of conducting a DDoS 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. An emerging and increasingly more prevalent set of DDoS attacks are the so-called application-layer or Layer-7 attacks that mimic a Flash Crowd event. A legitimate Flash Crowd event is a situation where some popular information emerges on a web site (such as a news story or a sports event), and many browsers (i.e., human visitors) attempt to access that information, thus creating a large demand/load on the server. An attacker can easily achieve a "Flash Crowd"-looking effect by performing an excessive number of seemingly legitimate actions on the target web application – such as database queries and transactions. From the logistics point of view, this kind of attack is typically executed by means of cleverly programmed crawlers instructed to perform a semi-random walk through the victim web site links, giving an illusion of a web site traversal conducted by a regular human visitor. Additionally, in order to hide their true identity, these smart DDoS-executing crawlers can resort to using spoofed user agent strings[1]. Since the signatures of such DDoS attacks look very much like a legitimate Flash Crowd event on a website, it is difficult to construct an effective metric for their mitigation, as well as to defend against them. Real-world examples of application-layer DDoS attacks that mimic Flash Crowd are reported in [3] and [4].

Now, the key tasks behind building a successful DDoS detection system that would defend against application-layer DDoS attacks that mimic a Flash Crowd event are: 1) to effectively distinguish between human and machine-generated web/HTTP sessions and, moreover, 2) in the group of machine-generated sessions to effectively distinguish between the sessions corresponding to benign vs. sessions corresponding to malicious crawlers.

Unfortunately, most real-world systems and techniques that are currently used to provide a defense against DDoS attacks are too generic and unsuitable for dealing with application layer DDoS attacks. Namely, on one side of anti-DDoS solution spectrum, there are rule-based and/or anomaly-detection firewalls and intrusion-prevention systems (IPSs). These devices/systems are generally effective in combating simple flood-type and 'off-the-shelf' forms of DDoS attacks. However, when dealing with more subtle and/or advanced

[1] A user agent string, part of the HTTP request packet, specifies the hardware/software (i.e., browser, crawler, Smartphone, tablet or others) used by the client to communicate with the server in the client-server communication.

forms of attacks, such as degrading application-layer attacks[2], their main drawbacks are:

a) Most firewalls and IPSs rely on the well-known and publicized attacks signatures in order recognize and defend against DDoS attacks. However, degrading application-layer DDoS attacks tend to be uniquely crafted for one/each particular web-site and thus do not conform to the 'generic' attack signatures. (For a good overview of "How Traditional Firewalls Fail Today's Networks" see a recent report by Dell SonicWall [5].)
b) Another drawback of traditional firewalls and IPSs is that they react with a 'delay' in blocking a malicious user (i.e., stopping a DDoS attack), as they need to be able to observe a user's behavior for a period of time before pronounce the user 'malicious'. In the case of degrading application layer DDoS attacks this delay may be significant.

On the other side of anti-DDoS solution spectrum are techniques that aim to distinguish between (malicious) automated visitors and regular human visitors to a web-site by relying on the so-called challenge-response tests (i.e., numerical and graphical puzzles), such as the well-known CAPTCHA. Although generally effective in accomplishing their task, the main drawbacks of this group of solutions are:

a) They are, often, annoying to human visitors, and therefore are rarely used for the protection of commercial web sites. (In a recent Scientific American article "Time to Kill-Off CAPTCHAs" [6], the author eloquently summarizes the commonly felt negative sentiment about CAPTCHA technology.)
b) They treat all automated crawlers equally – both the benign and malicious ones - by completely blocking their access to a web site.

In this paper, we propose an integrated machine-learning based anti-DDoS solution that aims to combine the best of both above mentioned approaches, while at the same time being effective in defending against application layer DDoS attacks with unique attack signature. Specifically, in Section II of this paper, we provide a general overview of our newly proposed anti-DDoS solution (see Fig. 1). In Section III we outline the main characteristics of the solution's first component – the SOM classifier – which is responsible for performing preliminary classification of visitors to a web site. We also present some of the most relevant experimental results derived using this classifier. (These results have been previously reported in [1].) In Section IV, we discuss the motivation behind employing the second-stage Time-Domain Analyzer, and present some key experimental findings pertaining to this analyzer. We close the paper in Section V by outlining the main directions for our future work.

II. Our Integrated Anti-DDoS System

An outline of our newly proposed multi-stage anti-DDoS system is provided in Fig. 1. The solution comprises an adaptable two-stage anomaly detection system – the first stage consisting of an SOM Classifier (described in more detail in Section III) and the second stage of a Time Domain Analyzer (described in Section IV). The main task of the SOM classifier is to categorize each visitor to a web site into one of the following four groups: human (benign) visitor, well-behaved automated visitor, malicious automated visitor, and unknown visitor. Visitors that exhibit clearly benign behavior (most human and well-behaved automated visitors) are granted uninterrupted access to the site. Visitors that are categorized as malicious crawlers or suspicious unknown visitors are immediately blocked from accessing the site. Finally, for visitors that are categorized as human, but in some aspects of their behavior resemble malicious crawlers, the system performs more detailed time-domain behavior analysis before resorting to the use of a challenge-response test (i.e., CAPTCHA). Clearly, any of these (human but suspiciously behaving) visitors that end up failing the challenge-response test will be denied further access/service.

Note that, in general, our system can be adapted to optimally operate for each particular web-site and its respective visitor population. In the case of some web-sites this may imply that the group of 'unknown visitors' be granted access to the site. For example, if the website being protected is an University website, web admin staff would likely allow benign known or brand new unknown search engine web crawlers to index their website. Alternatively, if the website being protected is an online content management application, with visitors that are exclusively users of the application, a web admin staff will likely chose to block everyone but human visitors since there is no need for search engines crawlers (and therefore public visitors) to index this type of a domain.

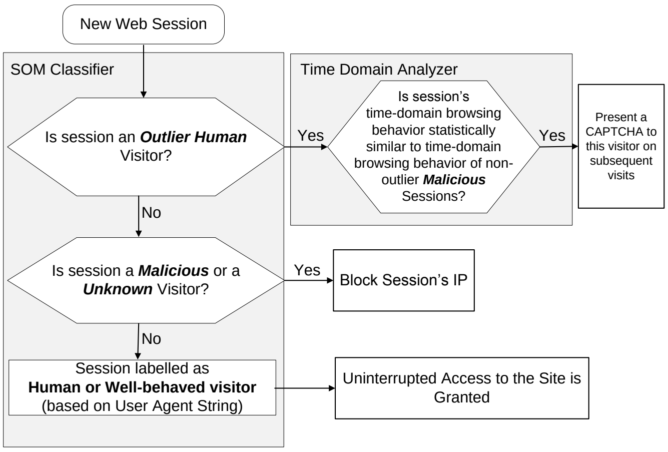

New Web Session
SOM Classifier
Time Domain Analyzer
Is session an ***Outlier Human*** Visitor?
Yes
Is session's time-domain browsing behavior statistically similar to time-domain browsing behavior of non-outlier ***Malicious*** Sessions?
Yes
Present a CAPTCHA to this visitor on subsequent visits
No
Is session a ***Malicious*** or a ***Unknown*** Visitor?
Yes
Block Session's IP
No
Session labelled as **Human or Well-behaved visitor** (based on User Agent String)
Uninterrupted Access to the Site is Granted

Fig. 1 The workflow of the real-time degrading application-layer anti-DDoS system

[2] In degrading Layer-7 DDoS attacks [23], attacker aims to partially degrade the victim's network response rate from the viewpoint of legitimate victim's clients, while in flooding or disruptive DDoS attacks, attacker aims to completely shot down the victim's network and prevent all legitimate clients from accessing it. Note also that in slow-rate DDoS attacks, attackers employ legitimate-looking network sessions that prevent the victim from detecting the attack. The flood DDoS attacks are much easier to detect since the network becomes completely unresponsive.

The main advantages of our solution presented in Fig. 1 over the existing anti-DDoS solutions are:

1) Unlike the systems that rely solely on the use of challenge-response tests, our solution makes a distinction between malicious and benign automated web visitors, and prevents only the malicious automated users from accessing a web site.
2) Through a customized machine-learning-based approach, the anomaly-detection component of our anti-DDoS system is able to identify (i.e., derive) attack signatures that are finely tuned to each particular web site. (I.e., the SOM network can be optimally trained for each particular web-site and its visitor population.) As a results, our system is far superior in dealing with degrading application-level DDoS attacks relative to the existing off-the-shelf firewalls and intrusion-prevention systems.
3) Our system resorts to the use of challenge-response tests only when there is a high certainty that a particular visitor to the site is malicious. Accordingly, the likelihood that a (benign) human user be exposed to (i.e., annoyed by) a challenge-response test is minimized.

III. SOM Web-Visitor Classifier

A. SOM Overview

The first stage of our anti-DDoS system deploys an unsupervised machine learning (i.e., neural network) classifier – the Self-Organizing Maps (SOM) [7]. The SOM algorithm was chosen mainly for the following reasons:

a) Its *topology preservation* ability, which implies that input samples that are close to each other in an n-dimensional space will also be close to each other in a 2D SOM map.
b) Its ability to produce natural clustering, i.e. clustering that is robust to statistical anomalies. This type of clustering is more effective in providing unbiased look and understanding of the underlying data set and, also, it is less sensitive to the presence of sporadic data outliers (i.e. presence of sporadically alterable features found in our dataset).
c) Superior visualization of high-dimensional input data in 2D-representation space. This was also important in our case since we were able to plot our 10-dimensional input data in 2D space for simple visual observation of cluster distributions.

B. SOM Classifier Experimentation

We performed our analysis of the SOM web-user classifier on two datasets: 1) a smaller web server access log from *www.cse.yorku.ca (CSE)* web domain and 2) a larger web server access log from www.*yorku.ca (YORKU)* web domain (see Table I). The purpose of performing our analysis on differently-sized datasets was to evaluate whether our analysis can be generalized to a significantly larger web domains with varied/different web visitors.

Table I Class distributions in the CSE and YORKU datasets

	CSE	**YORKU**
# of Human Sessions	53640	707854
# of Well-behaved Crawler Sessions	7607	9014
# of Malicious Visitor Sessions	287	860
# of Unknown Visitor Sessions	4042	3445
Total	65576	721193

As described in [1], for each web visitor session, we extracted the following features from the datasets: 1) Click number, 2) HTML-to-Image Ratio, 3) Percentage of PDF/PS file requests, 4) Percentage of 4xx error responses, 5) Percentage of HTTP requests of type HEAD, 6) Percentage of requests with unassigned referrers, 7) Number of bytes requested from the server, 8) Page Popularity index, 9) Standard deviation of requested page’s depth and 10) Percentage of consecutive sequential HTTP requests. As shown in past research studies, namely [8], [9], [10], [11] and [12], these features are shown to be useful in distinguishing between browsing patterns of web robots and humans.

As described in [1], the session labels were generated by matching the user agent string of each visitor to a list of known user agent strings of browsers, well-behaved crawlers and malicious crawlers. The log analyzer maintains a table of user agent fields of all known (malicious or well-behaved) web crawlers and browsers. This table was built by compiling the data found on web sites [13], [14] and [15]. The details of the dataset labeling process are shown in Fig. 2. Note that we label all sessions that carry a user agent string of a known browser but access the robots.txt (operation performed only by crawlers) as malicious as well.

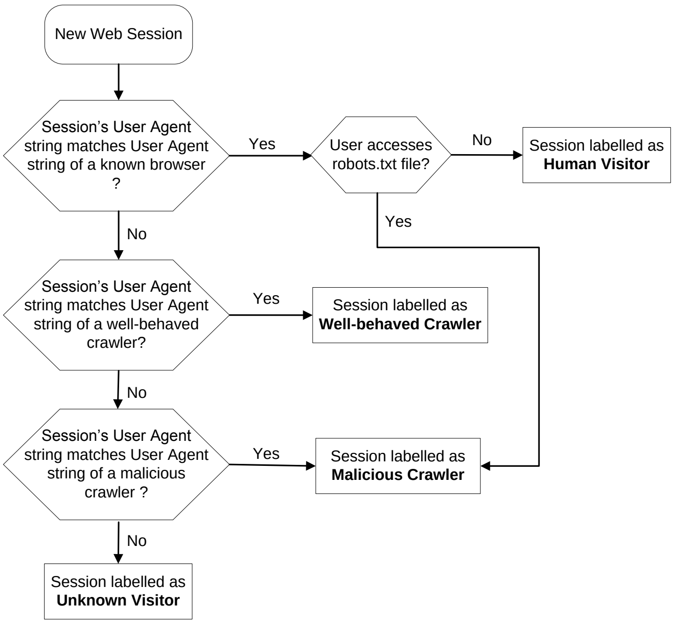

New Web Session
Session’s User Agent string matches User Agent string of a known browser ?
Yes
User accesses robots.txt file?
No
Session labelled as **Human Visitor**
Yes
No
Session’s User Agent string matches User Agent string of a well-behaved crawler?
Yes
Session labelled as **Well-behaved Crawler**
No
Session’s User Agent string matches User Agent string of a malicious crawler ?
Yes
Session labelled as **Malicious Crawler**
No
Session labelled as **Unknown Visitor**

Fig. 2 The flow chart of our data labelling algorithm

From the clustering results, we were able to identify three distinct groups of sessions that could present a particular challenge for any Layer-7 anti-DDoS systems (from [1]):

1) Sessions that are labelled as malicious crawlers but 'behave' like humans – we refer to these sessions as *outlier malicious sessions* The position of these sessions in the SOM graph is shown in Fig. 3. Security staff administering a web site would be very much interested in taking a closer look at this group of malicious sessions. Namely, currently the only way of identifying these sessions as malicious is by looking at their user agent string. However, with an incrementally higher level of sophistication – e.g., just by employing a fake but legitimate-looking user agent string – these sessions would blend in with actual/regular human sessions and become virtually undetectable.
2) Sessions that are labelled as unknown visitors but 'behave' like humans – we refer to these sessions as *outlier unknown sessions*. The position of these sessions in the SOM graph is shown in Fig. 4. Security staff administering a web site would be very much interested in taking a closer look at this group of sessions since they carry unknown, suspiciously incorrectly crafted or even missing user agent string labels.
3) Sessions that are labelled as humans but 'behave' like malicious crawlers – we refer to these as *outlier human sessions.* The position of these sessions in the SOM graph is shown in Fig. 5. We speculate that security staff administering a web site would be particularly interested in detecting and analyzing this group of sessions. Namely, these are likely sessions corresponding to sophisticated human-like-behaving malicious crawlers that are attempting to disguise their identity by spoofing their respective user agent strings.

Note that we arrived at the similar results with both the CSE and the YORKU datasets.

IV. Time-Domain Analyzer

The results of our study presented in [1] indicate that, even at the current level of web-crawlers sophistication, effective web-crawlers categorization is becoming an increasingly complex task. Accordingly, in order to effectively distinguish suspicious from truly malicious web sessions, we propose that another stage/component be added to our integrated anti-DDoS system – in particular, a stage that focuses on the time-domain behavior analysis of potentially suspicious visitors.

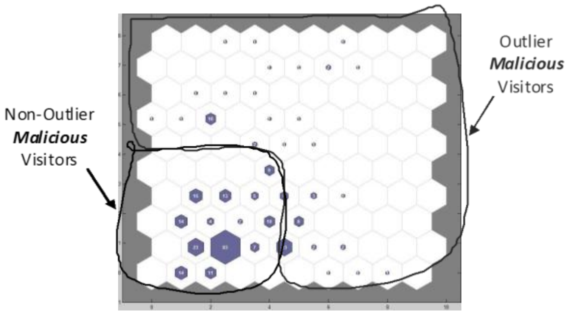


Fig. 3 Distribution of outlier and non-outlier malicious sessions in the SOM map

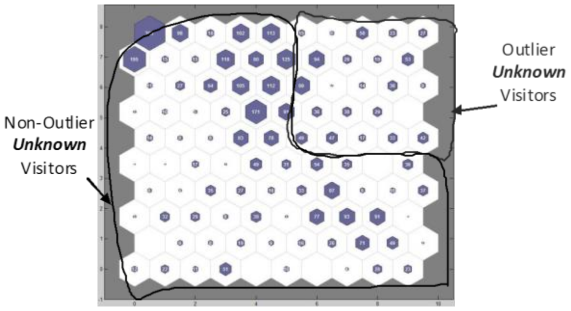


Fig. 4 Distribution of outlier and non-outlier unknown sessions in the SOM map

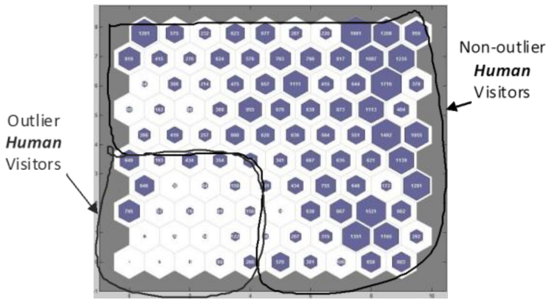
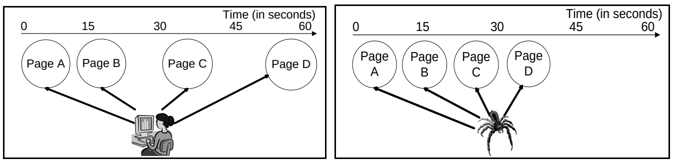


Fig. 5 Distribution of outlier and non-outlier human sessions in the SOM map

Here are the two simple illustrations as for how the inclusion of time-wise analysis may benefit the task of suspicious user classification:

- The time duration of a session is an important feature which has not been previously considered in our analysis or in the previous research works dealing with the issue of web-user classification. Namely, two sessions might comprise exactly the same number of accessed pages – looking at it as a simple number. However, it really matters whether all these pages are accessed as/in a rapid sequence, or over a longer period of time. Clearly, a rapid sequence access is likely to belong to a crawler, while longer sessions are likely to belong to humans (refer to Fig. 6).
- The time spent viewing each individual page is another important parameter to consider. Namely, crawlers are likely to spend the same amount of time ‘viewing’ each page, while the amount of time that humans spend viewing a page will likely be highly correlated to the contextual importance of that page relative to others – something an automated crawler is not able to comprehend/detect.



Human visitor Web crawler

Fig. 6 Comparison of typical timing differences in web page access behaviour between a human visitor and a web crawler

A. Time-Domain Browsing Behaviour Model

In our time domain analyzer, we characterize the time-wise browsing behavior of a session based on the model presented in the recent study in [16]. In this study, authors state that the human browsing characteristics, such as page popularity, page viewing time and browsing session length (i.e., number of web pages visited in a session), can be modeled by a Markov model based on the three statistical distributions.

For instance, the web page popularity can be modeled by the following Zipf-Mandelbrot distribution function:

$$\Pr(W = i) = \frac{\Omega}{(i+q)^{\alpha}} \quad (1)$$

where $\Pr(W = i)$ is the access probability of page w_i, i is the rank/popularity of the web page, α ($\alpha > 0$) is the skewness factor, which characterizes the length of the tail of the distribution, and q ($q \geq 0$) is the plateau factor.

The page viewing time interval can be modeled by the Pareto's probability density function defined as shown in (2),

$$\Pr(V = v) = \frac{\alpha \cdot v_m^{\alpha}}{v^{(\alpha+1)}} \quad (2)$$

where v is web page viewing time, v_m is the minimum viewing time for all web pages and α is called the Pareto index.

Finally, the browsing session length can be modeled by the following Inverse Gaussian's distribution function:

$$\Pr(L = l) = \sqrt{\frac{\lambda}{2\pi l^3}} \exp\left[\frac{\lambda(l-\mu)^2}{2\mu^2 l}\right], l = 1, 2, \ldots \quad (3)$$

where L is the number of links that a visitor visits (i.e., follows) on a web site in a single session, average value of L, i.e. $E[L] = \mu$, variance $Var[L] = \mu^3 / \lambda$ and $\lambda > 0$ is the shape parameter describing the length of the distribution's tail.

Note that this human browsing behavior model employs a time-wise feature, i.e. page viewing time, but as well two additional features: 1) page popularity (which models users' page selection behavior) and 2) browsing session length (i.e. click number) which are not based in time-domain. Also, note that the latter two features were employed in our unsupervised study, and as discussed, these features are known to be very effective at distinguishing between human and machine-generated sessions.

In [17], the same authors show that under very specific conditions, such as small Botnet size and specific traffic characteristics, these type of malicious bots (that model their browsing behavior to mimic human-like behavior) could be detected. The work in [16] and [17] are one of the most recent works on the browsing behavior modeling, however it builds upon a number of other works and results (namely [18], [19] and [20]) produced during the last decade and a half.

B. Correlation Testing in our Time-Domain Analyzer

In order to evaluate the time-wise browsing behavior differences between visitor types, with significant level of confidence, our time-domain analyzer employs two nonparametric correlation tests:

1) Kolmogorov–Smirnov Test of 2 Independent Samples – a nonparametric statistical test that measures if there is a significant difference at any point along the two cumulative distribution functions (CDFs) between two samples which also implies that the two samples are derived from different populations.
2) Mann-Whitney U Test – a nonparametric statistical test that detects the significant difference between the medians of the two samples.

Specifically, both of these tests are employed to identify outlier human sessions that exhibit the browsing behavior (characterized in terms of the web page popularity rankings, page viewing times and number of pages visited during a session) that is not significantly different from the browsing behavior of non-outlier malicious sessions.

Note that our system could identify suspicious outlier human visitors even in the absence of any maliciously-labeled web session. In this scenario, the human visitors that are significantly different from non-outlier human visitors in terms of the three browsing behavior metrics would be asked to solve CAPTCHA puzzles by the system. Also note that these two tests apply different strategies to evaluate the differences between the given samples. By applying both techniques, we tend to provide a more holistic evaluation of the statistical differences between session types.

C. Experimental Results of the Correlation Tests

We experimentally evaluated the application of the correlation tests on the CSE and YORKU datasets from [1]. The web page popularity rankings, the web page visiting/viewing times and browsing lengths for outlier and non-outlier sessions in CSE and YORKU datasets were fitted to distributions in (1), (2) and (3), respectfully. The α and v_m parameters were derived by utilizing the method described in [21]. The μ and λ parameters were derived by utilizing the maximum likelihood estimation function provided with Matlab software package.

The results of applying the two correlation tests on the three metrics are displayed in Tables II-IV. In the case of Mann-Whitney U Test, the medians from the two samples are significantly different with 95% confidence if the obtained absolute value of the z-score is equal to or greater than 1.96. In the case of Kolmogorov-Smirnov Test 2, the empirical CDFs for the two samples are significantly different with 95% confidence if the Kolmogorov-Smirnov K-S statistic is greater than or equal to the so-called critical value of Kolmogorov-Smirnov Test of 2 Independent Samples – i.e., K statistic. Note that all of the results displayed in Tables II-IV are

TABLE II CORRELATION METRIC SCORES FOR WEB PAGE POPULARITY

Session Type Comparisons	**Mann-Whitney z-score**		**K-S statistic / K statistic**	
	CSE	***YORKU***	***CSE***	***YORKU***
Actual Human Visitors vs. Outlier Malicious Visitors	-5.38	45.55	0.36 / 0.015	0.534 / 0.03
Actual Human Visitors vs. Outlier Unknown Visitors	-14.53	19.1	0.22 / 0.044	0.25 / 0.038
Outlier Human Visitors vs. Actual Malicious Visitors	-36.2	69.14	0.36 / 0.029	0.34 / 0.014

TABLE III CORRELATION METRIC SCORES FOR WEB PAGE VIEWING TIME

Session Type Comparisons	**Mann-Whitney z-score**		**K-S statistic / K statistic**	
	CSE	***YORKU***	***CSE***	***YORKU***
Actual Human Visitors vs. Outlier Malicious Visitors	119.3-	27.77	0.62 / 0.016	0.36 / 0.03
Actual Human Visitors vs. Outlier Unknown Visitors	-14.7	2.86	0.22 / 0.046	0.1 / 0.04
Outlier Human Visitors vs. Actual Malicious Visitors	-23.48-	22.58	0.33 / 0.03	0.18 / 0.015

TABLE IV CORRELATION METRIC SCORES FOR BROWSING SESSION LENGTH

Session Type Comparisons	**Mann-Whitney z-score**		**K-S statistic / K statistic**	
	CSE	***YORKU***	***CSE***	***YORKU***
Actual Human Visitors vs. Outlier Malicious Visitors	-1.97	8.24	0.14 / 0.09	0.35 / 0.11
Actual Human Visitors vs. Outlier Unknown Visitors	-2.92	19.38	0.066 / 0.065	0.28 / 0.034
Outlier Human Visitors vs. Actual Malicious Visitors	8.43	16.98	0.28 / 0.1	0.3 / 0.058

significantly different in terms of both correlation metrics with 95% confidence.

We have made the following main conclusions from our results:

- **Non-outlier human vs. outlier malicious/unknown sessions.** The application of Mann-Whitney U and Kolmogorov-Smirnov 2 Independent Sample Tests show that there is a significant difference between the medians and distributions of the three statistical metrics between non-outlier human visitors and outlier malicious/unknown sessions. As such, these results have a great practical significance. Namely, they suggest that in the case that the outlier malicious or unknown sessions were marked by a spoofed browser-based user agent string – which would make them less 'obvious' and not as easily identifiable by the SOM algorithm – the use of time-domain analysis would provide for an effective way of distinguishing them from non-outlier (true) human sessions.
- **Non-outlier malicious vs. outlier human sessions.** The application of Mann-Whitney U and Kolmogorov-Smirnov 2 Independent Sample Tests show that there is also a significant difference between the medians and distributions of the three statistical metrics between non-outlier malicious visitors and outlier human sessions. These results may be an indication that the outlier human session, as identified by the SOM algorithm (see Section III), are not actually malicious but instead may be generated by legitimate human visitors that happen to exhibit non-typical human browsing behavior. Note also that these human visitors would not be CAPTCHA-ed by our anti-DDoS system.

V. FUTURE WORK

In our future work, we plan to include additional web logs from one or more non-academic public (or private) organizations. By analyzing a set of web logs from different organizations we aim to generalize the conclusions we have derived at this point in our research.

Also, we plan to evaluate the real-world DDoS bot, such as a Dirt Jumper [22], in a sandbox environment. The purpose of this task would be to evaluate how closely the actual DDoS bot's browsing behavior compares with the browsing behavior of actual human visitors and malicious crawlers.

REFERENCES

[1] D. Stevanovic, N. Vlajic, and A. An, "Detection of Malicious and Non-malicious Website Visitors Using Unsupervised Neural Network Learning," *Applied Soft Computing*, vol. 13, no. 1, pp. 698-708, Jan. 2013.

[2] B. Gerneglia. (2012, Mar.) Infosecisland. [Online]. http://www.infosecisland.com/blogview/20727-Cyber-Attacks-are-Fastest-Growing-National-Security-Threat.html

[3] K. Poulsen. (2004) FBI Busts Alleged DDoS Mafia. [Online]. http://www.securityfocus.com/news/9411

[4] H. N. Security. (2011, Oct.) Top DDoS attacks of 2011. [Online]. http://www.corero.com/en/company/news_and_events?item_id=4

[5] D. SonicWall. (2012) IDG Connect. [Online]. http://www.idgconnect.com/view_abstract/13111/how-traditional-firewalls-fail-today-networks-and-why-next-generation-firewalls-will-prevail?source=connect

[6] D. Pogue. (2012, Feb.) Scientific American. [Online]. http://www.scientificamerican.com/article.cfm?id=time-to-kill-off-captchas

[7] T. Kohonen, *Self-Organizing Maps*, 3rd ed. New York: Springer-Verlag, Berlin Heidelberg, 2001.

[8] D. Doran and S. S. Gokhale, "Web robot detection techniques: overview and limitations," *Data Mining and Knowledge Discovery*, pp. 1-28, Jun. 2010.

[9] 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.

[10] P. N. Tan and V. Kumar, "Discovery of Web Robot Sessions Based on their Navigation Patterns," *Data Mining and Knowledge Discovery*, vol. 6, no. 1, pp. 9-35, Jan. 2002.

[11] J. X. Yu, O. Yuming, C. Zhang, and S. Zhang, "Identifying interesting visitors through Web log classification," *Intelligent Systems*, vol. 20, no. 3, pp. 55-59, Jun. 2005.

[12] C. Bomhardt, W. Gaul, and L. Schmidt-Thieme, "Web Robot Detection - Preprocessing Web Logfiles for Robot Detection," in *In Proc. SISCLADAG*, Bologna, Italy, 2005.

[13] (2011, Aug.) User-Agents.org. [Online]. http://www.user-agents.org

[14] (2011, Aug.) Bots vs. Browsers. [Online]. http://www.botsvsbrowsers.com/

[15] (2012, May) User-agent-string.inf. [Online]. http://user-agent-string.info/

[16] S. Yu, Z. Guofeng, S. Guo, X. Yang, and A. V. Vasilakos, "Browsing Behavior Mimicking Attacks on Popular Web Sites for Large Botnets," in *proceedings of 2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, Shanghai, China, April, 2011, pp. 947-951.

[17] S. Yu, S. Guo, and I. Stojmenovic, "Can we beat legitimate cyber behavior mimicking attacks from Botnets?," in *IEEE INFOCOMM*, Orlando, Florida, 2012, pp. 2851-2855.

[18] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and zipf-like distributions: Evidence and implications," in *Proceedings of the INFOCOM*, New York, 1999, pp. 126-134.

[19] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: evidence and possible causes," *IEEE/ACM Transactions on Networking*, vol. 5, no. 6, pp. 835-846, Dec. 1997.

[20] B. A. Huberman, P. Pirolli, J. E. Pitkow, and R. M. Lukose, "Strong regularities in world wide web surfing," *Science*, vol. 280, no. 5360, Apr. 1998.

[21] A. Clauset, C. R. Shalizi, and M. E. J. Newman, "Power-Law Distributions in Empirical Data," *SIAM*, vol. 51, no. 4, pp. 661-703, Nov. 2009.

[22] Prolexic. (2012, Mar.) Threat: Dirt Jumper v3. [Online]. http://unknown.prolexic.com/pdf/ProlexicThreatAdvisoryDirtJumper.pdf

[23] A. Asosheh and N. Ramezani, "Comprehensive Taxonomy of DDoS Attacks and Defense Mechanisms Applying in a Smart Classification," *WSEAS Transaction on Computers*, vol. 7, no. 4, pp. 281-290, Apr. 2008.