Webology, Volume 15, Number 2, December, 2018

Distributed Denial of Service Attack Detection in Application Layer Based on User Behavior

Silvia Bravo

Corresponding Author, Professor, Faculty of Engineering and Applied Sciences, Technical University of Cotopaxi, Ecuador. PhD Candidate, Department of Computer Science, National University of San Marcos, Peru. ORCID. E-mail: silvia.bravom@utc.edu.ec

David Mauricio

Professor, Department of Computer Science, National University of San Marcos, Peru. ORCID. E-mail: dmauricios@unmsm.edu.pe

Received June 10, 2018; Accepted December 20, 2018

Abstract

Distributed Denial of Service (DDoS) attacks are a threat to the security of red. In recent years, these attacks have been directed especially towards the application layer. This phenomenon is mainly due to the large number of existing tools for the generation of this type of attack. The highest detection rate achieved by a method in the application capacity is 98.5 percent. Therefore, the problem of detecting DDoS attacks persists. In this work an alternative of detection based on the dynamism of the web user is proposed. To do this, evaluate the user's characteristics, mouse functions and right click. For the evaluation, a data set of 11055 requests was used, from which the characteristics were extracted and entered into a classification algorithm. To that end, it can be applied once in Java for the classification of real users and DDoS attacks. The results showed that the evaluated characteristics achieved an efficiency of 100 percent. Therefore, it is concluded that these characteristics show the dynamism of the user and can be used in a detection method of DDoS attacks.

Keywords

User behavior; Distributed denial of service; Application layer; Attack detection

Introduction

The detection of Distributed Denial of Service (DDoS) attacks is one of the biggest problems facing the security architecture of the network. Therefore, it has become an important factor of study in the field of computer security. A DDoS attack occurs when an attacker coordinates their attacks using several machines, called zombies, towards a specific target or server. The aim of the attacker is to make massive requests to the victim machine to saturate it and that it stops serving the requests of real users.

To counteract this type of attack, several detection mechanisms have been proposed, both at the network level and at the application level (Table 1). The highest detection rate obtained to date is 99.4 percent, and has been achieved by implementing a network-level method (Kumar et al., 2011). The dataset used in that work is KDD cup dataset, from which 300,000 connection records were extracted between DDoS attacks and real users. On the other hand, in the methods implemented at the application layer level, the best detection rate obtained is 98.5 percent (Zolotukhin et al., 2016), of which the dataset used is not available, however for the tests, service requests were simulated and used Sslsqueeze and Slowloris for the generation of attacks.

The detection mechanisms, for the most part, focus their efforts on the network layer. However, currently the largest number of attacks have been directed to the application layer, because they are easy to execute because of the large amount of existing software (Zolotukhin et al., 2016; Johnson Singh et al., 2016), and more difficult to detect because they are illegitimate requests that they camouflage themselves as requests from real users. So the present work focuses on the detection of attacks in the application layer.

All methods of detection of attacks in the application layer are based on characteristics, their efficiency depends on them. However, no detection method contemplates the user's interaction with the system, which is a feature that can differentiate between a human and a robot (Zhou et al., 2014). In this work we identify new features based on the interaction of the user with the system, specifically its interaction with the mouse (mouse movement and right click), and verify its influence on the detection of DDoS attacks.

This work is organized as follows. In section 2, a literature review of the characteristics for the detection of DDoS attacks at the application layer level is made. Section 3 presents the characteristics of user behavior for the detection of attacks, presents the methods used to capture the characteristics and proposes a classification algorithm to identify a real user and a robot, in section 4 the numerical experiments, in section 5 the results and discussions are shown and, finally, the conclusions are presented.

Table 1. Mecanism at the network level and application level

Level	Reference
Network	Al-Duwairi et al. (2006); Al-Wang et al. (2014); Anurekha et al. (2012); Beak et al. (2007); Chen et al. (2005); Chen et al. (2006); Chen et al. (2008); Chen et al. (2007a); Chen et al. (2007b); Chen et al. (2013a); Chen et al. (2013b); Chonka et al. (2009); Doron et al. (2011); Duwairi et al. (2013); François et al. (2012); Kang et al. (2013); Kang et al. (2014); Kim et al. (2006); Kulkarni et al. (2006); Kumar et al. (2011); Kumar et al. (2013), Lee et al. (2005), Lee et al. (2008), Lee et al. (2012), Li et al. (2005); Liu et al. (2011); Lu et al. (2009); Luo et al. (2013); Luo et al. (2014); Ma et al. (2014); Meenakshi et al. (2007); Mirkovic et al. (2005); Rahmani et al. (2012); Seo et al. (2013); Spyridopoulos et al. (2013); Yan et al. (2009); Sachdeva et al. (2014); Udhayan et al. (2013); Varalakshmi et al. (2013); Wang et al. (2007); Wang et al. (2012); Wang et al. (2014); Wu et al. (2013); Xiang et al. (2011); Xiao et al. (2006); Xiao et al. (2015); Yaar et al. (2005); Yau et al. (2005); Zhang et al. (2012); Zhenwei et al. (2011)
Application	Dick et al. (2016); Giralte et al. (2013); Huang et al. (2014); Johnson Singh et al. (2016); Ranjan et al. (2009); Saravanan et al. (2016); Xie et al. (2009); Zhou et al. (2014); Zolotukhin et al. (2016)

Literature review of features

The DDoS attacks in the application layer are characterized by the massive sending of requests, causing limitations in the access to the web services of legitimate users. Figure 1 shows, the transactionality of the system, we observe the requests made by the user or attacker to the web server. In the process of detecting this type of attacks, it is necessary to extract the characteristics of the requests sent to the server. For this, algorithms or procedures are used that filter information on characteristics such as distance measurements (Gavrilis et al., 2005) and Nguyen et al. (2008) provided by the request flows (Wang et al., 2008). Once the characteristics are obtained, algorithms or classification criteria are used to detect attacks. Machine learning algorithms are commonly used in the classification of real users and DDoS attacks (Xiang et al., 2005). There are also classification criteria based on Soft computing techniques and its hydrological approach (Kumar et al., 2011). Finally, when a DDoS attack is detected, these will be discarded from the set of requests, while the requests of the real users enter the web server to obtain a response.

The detection of DDoS attacks depends to a large extent on the characteristics that are used. The adequate selection of characteristics will allow to improve the detection process in efficiency and processing time (Kumar et al., 2011). Therefore, in recent years, the efforts in the detection of DDoS attacks have focused on the search for features that contribute to the detection of attacks in the application layer. Table 2 shows 30 characteristics that are used in the detection of attacks.

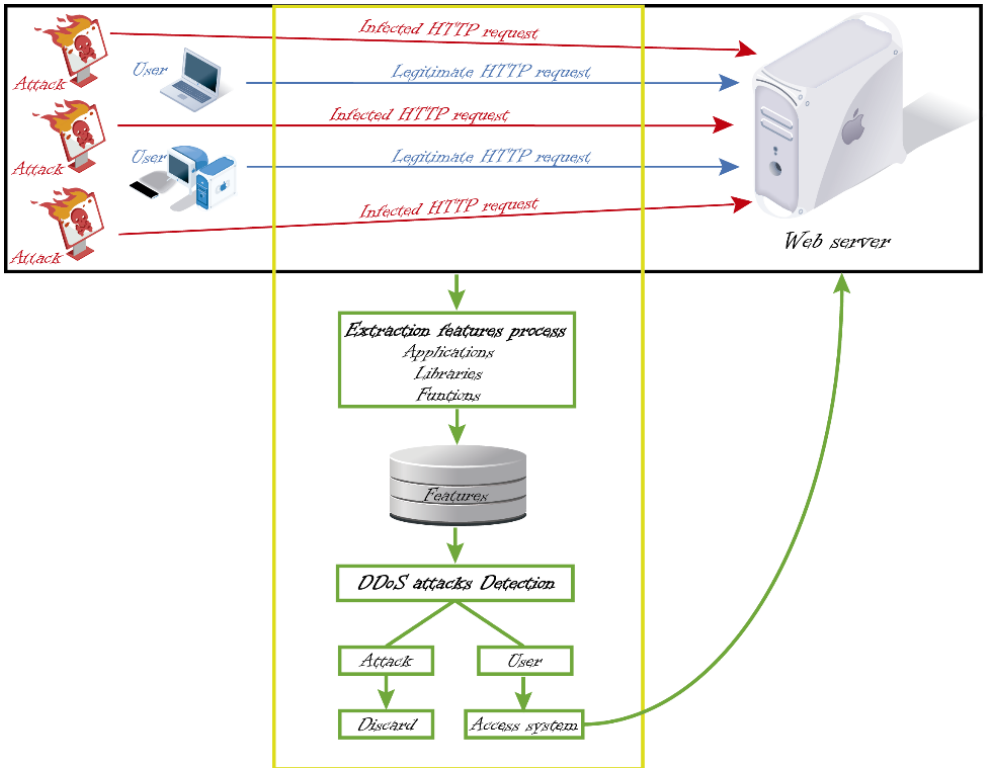

Figure 1. Execution and detection of DDoS attack in the application layer

Table 2 shows the characteristics of the data flow of each client, the characteristics of IP packets in a time interval and the behavior patterns of each user. They are extracted at intervals of time when a client connects to a domain (Dick et al., 2016). These characteristics are of the statistical type and record the client's access to system resources and the frequency with which each client requests a resource in the domain.

The highest detection rate obtained to date is 98.5 percent and has been achieved using software generated in Pytohn using the Intrusion Detection System (IDS) technique (Zolotukhin et al., 2016). However, the resources available to attackers are evolving day by day. Therefore, despite the fact that attack detection mechanisms reach high rates, the problem persists.

Table 2. Description of features

Feature	Description	Reference
Access pattern	Access pattern is constantly repeated, develop a frequent path detector which involves checking the requests of the complete flow.	Ranjan et al. (2009)
Average length of query strings of client	Average of consultations made by clients.	Huang et al. (2014)
Click number of web objects	The deviation from the entropy of the training data set fitting to the hidden semi-Markov model can be considered as the abnormality of the observed data set.	Zhou et al. (2014)
Client legitimacy	The legitimacy of a user sending an enormous number of requests is checked against the known client clusters.	Saravanan et al. (2016)
Duration of the conversation	Conversations initiated by one client to the destination socket during some short time interval.	Zolotukhin et al. (2016)
Entropy of request type (GET/POST/OTHER)	The fractions of request types per connection (GET, POST,or OTHER).	Huang et al. (2014)
Entropy of the requests	Entropy to measure the amount of disorder in the flow of the packets or request in the form of an HTTP GET request at multiple time slots.	Johnson Singh et al. (2016)
Flow similarity	Flow similarity is considered as a key parameter for discriminating between legitimate and illegitimate flows and a few works	Saravanan et al. (2016)
Fraction of connections for domain that accepts any version of English	Connection (e.g., en-us) in Accept-Language.	Huang et al. (2014)
Fraction of connections of client that request the most frequent resource path	A client accesses and also count how often each client requests the currently most common path on the domain.	Huang et al. (2014)
HTTP GET request count	The operation of HTTP starts with a client by sending a request to the server in the form of a request method.	Johnson Singh et al. (2016)
IP address	Source IP addresses, we are able to classify them into different traffic.	Giralte et al. (2013)
Maximal, minimal and average packet size	Average of these packet numbers and the mutual information of the fast Fourier Transform.	Zolotukhin et al. (2016)
Maximal, minimal and average size of TCP window	Number of packets received at the current time horizon and at the previous one.	Zolotukhin et al. (2016)
Maximal, minimal and average time to live (TTL)	Account time intervals between subsequent packets of the same flow.	Zolotukhin et al. (2016)
Number of bytes sent in 1 second	Packets in bytes sent from the client to the server and from the server to the client.	Zolotukhin et al. (2016)
Number of different resource paths of client	It includes the number of different resource paths that client has accessed.	Huang et al. (2014)
Number of packets sent in 1 second	Packets sent from the client to the server and from the server to the client.	Zolotukhin et al. (2016)
Number of request	Requests for the currently open windows and whether the number of requests for an open window.	Ranjan et al. (2009)
Number of users	Set of real users accessing a server.	Ranjan et al. (2009)

Percentage of encrypted packets with different properties	Since the traffic may be encrypted it is not always possible to define what web page these clients request.	Zolotukhin et al. (2016)
Percentage of packets with different TCP flags	As it was mentioned in the previous section, in this study, we concentrate on the traffic transferred over TCP.	Zolotukhin et al. (2016)
Session's requests	Requests for the currently open windows and whether the number of requests for an open window.	Xie et al. (2009)
Sum of incoming payload of all clients of domain	If requests from attacking IP addresses were to be processed, inspected, and filtered based on the individual payload.	Huang et al. (2014)
Sum of outgoing payload of all clients	If requests from attacking IP addresses were to be processed, inspected, and filtered based on the individual payload.	Huang et al. (2014)
Sum of response times of all clients of domain	Properties of all clients that interact with the domain in the time interval	Huang et al. (2014)
Sum of response times of client	Average durations until the first FIN packet is received and until the connection is closed, as well as the response time.	Huang et al. (2014)
Users browsing process	We see average and total length of such browsing sequences.	Dick et al. (2016)
Variance of the entropy	Variance of the entropy value, since the value of the variance provides the variations in the entropy value.	Johnson Singh et al. (2016)
Web page requested	In the case of an application level DDoS attack, the attack packets are in the form of web page requests.	Saravanan et al. (2016)

Feature of user behavior

Proposed features

The dynamism of the user is the user's interaction with the system and through it it is possible to know the behavior of a user and its difference with others (Ghezzi et al., 2014). The authentication of a user by means of his behavior has been a task studied from the point of view of information security (Stevanovic et al., 2014). Therefore, in order to avoid access by unauthorized users, several investigations (Ghezzi et al., 2014; Stevanovic et al., 2014; Urban, 2015; Abramson et al., 2013; Kim et al., 2014; Shen et al., 2013) have focused their efforts on a process called biometric behavior. Within this process are: the use of keystrokes, mouse dynamics and the interaction with the graphical user interface (GUI) (Stevanovic et al., 2014) for the identification of users.

Table 3 shows 24 characteristics that allow detecting the dynamism of the user and differentiating it from another. These characteristics are divided into two groups, these groups arise from the interaction of the mouse or keyboard and the user. In this paper, two characteristics are evaluated (mouse movement and right click), because in the data set used for the evaluation, these characteristics are present.

Mouse movement and right click allow to unequivocally identify a real user of a robot. In the case of mouse movement, a real user moves this peripheral to navigate through the web environment (Salmeron-Majadas et al., 2014). While right click is a special event that allows access to drop-down sub-menus, although it is not an event used regularly, it also identifies the dynamics of the user and the environment (Shen et al., 2013). On the other hand, the robots are generated by specialized software to make the largest number of requests to a system (Kumar et al., 2011), without the use of any peripheral. It is worth mentioning that the characteristics presented in Table 3, despite being used in the biometric process to identify a user of another, these have not been proven in the differentiation of real users and robots.

Table 3. Features of the mouse and keyboard

ID	Mouse Features	Reference
M1	Single-click	Shen et al. (2013)
M2	Double-click	
M3	Movement offset	
M4	Speed curve against time	
M5	Acceleration curve against time	
M6	Time	Salmeron-Majadas et al. (2014)
M7	Movement	
M8	Left or right button pressed or released	
M9	Coordinates of an event	
M10	Mouse position coordinates	Graepel et al. (2010)
M11	Mouse trajectory	
M12	Angle of the path in various directions	
M13	Curvature and its derivative	
M14	Mouse movement	
M15	Angular velocities	
M16	Tangential acceleration and jerk	
M17	Mouse movement coordinate	Gamboa et al. (2003)
M18	Movement angle	
M19	Time to move	
M20	Time of mouse clicks	
ID	Keyboard Features	
K1	Number of key press events	Shen et al. (2013)
K2	Average time between key press events	
K3	Average time per stroke	
K4	Number of times a given key has been pressed	

Features capture

Table 4 describes the characteristics of the mouse that can be captured and the techniques used for such purposes. These features can be captured using software developed in programming

languages that incorporate libraries or special functions (Shen et al., 2013; Salmeron-Majadas et al., 2014; Graepel et al., 2010; Gamboa et al., 2013).

Table 4. Features of the mouse and keyboard

Id	Library / Programming language	Reference
M1	Windows application (written in C#)	Shen et al. (2013)
M2		
M3		
M4		
M5		
M6	Java (kSquared.de library)	Salmeron-Majadas et al. (2014)
M7		
M8		
M9		
M10	NA	Graepel et al. (2010)
M11		
M12		
M13		
M14		
M15		
M16		
M17	Java applet and javascript	Gamboa et al. (2003)
M18		
M19		
M20		

Classification algorithm

Figure 2 shows the classification algorithm that allows the identification of DDoS attacks by means of mouse characteristics. The proposed characteristics allow to know if there is an attack or not, the process consists in verifying if the service request includes at least one of the proposed characteristics, which is considered a human user otherwise it is considered a robot. The algorithm calculates the accuracy rate of DDoS attacks by verifying the number of attacks found by the algorithm between the numbers of actual attacks in the dataset.

```
Input: Dataset
begin
Query right click, mouse movement, request URL
Loop Dataset
if request URL is active
   if right click is active or mouse movement is active
      add user;
   else
      add attack;
Query abnormal URL
   accuracy is equal request - abnormal_URL;
end
output accuracy;
```

Figure 2. Classification algorithm of real users and robots

Numerical experiments

Detection criteria

Figure 3 shows the architecture of the validation environment used for the construction of the classification algorithm of real users and DDoS attacks. In it, we consider the set of input data given by Linchman (2013). The use of the MySQL database manager was also observed for the extraction of the characteristics that were used in the validation, in order to create a new set of data with the selected characteristics. It enters the application created in Java for the classification process. It should be noted that the classification algorithm, allows the evaluation of the two interaction characteristics for the detection of computer attacks, these being: mouse movement and right click. Finally, results reports are generated, in which the total number of DDoS attacks and actual users found is shown, as well as the total time spent executing the entire process.

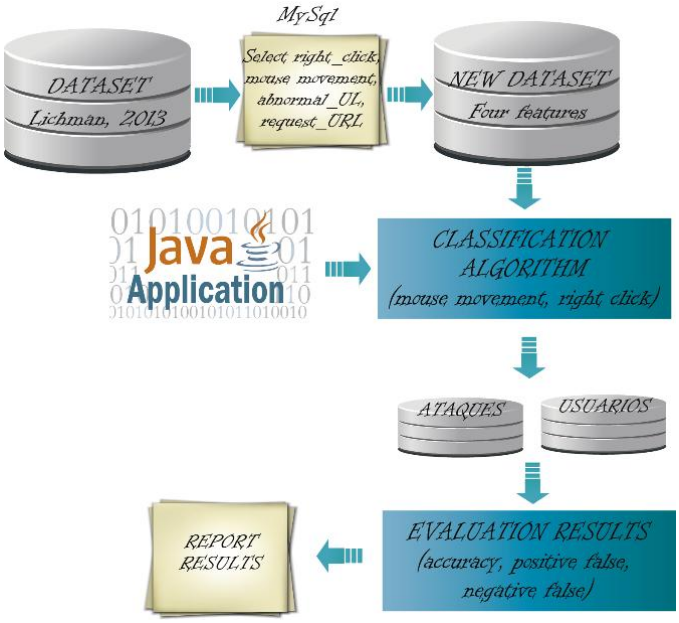

Figure 3. Validation environment architecture

Dataset

The dataset used in this work for the validation process of the classification algorithm was created by Lichman (2013). It contains 11055, of which 9096 are real users and the rest are DDoS attacks. This data set was selected because it reports the characteristics of the mouse to be evaluated. In addition, this data set contains 31 attributes from which four were extracted to perform the validation (right click, mouse movement, abnormal URL and request URL). It should be noted that, through the URL request feature, it is known whether a request was made to the system or not. On the other hand, the abnormal URL allows identifying the requests that are computer attacks.

Feature extraction

Figure 4 shows the general algorithm that extracts the features proposed in this work. To do this, an active request is identified in the set with the data to then identify the proposed variables. The features are extracted by SQL queries to the database. After executing the consultations, all records are obtained where a service or resource has been requested for subsequent analysis and reporting of results.

```
Input: Request = active
begin
Open database
Query = Select mouse_movement,
right_click from dataset
Execute Query
end
```

Figure 4. Algorithm used for the extraction of features

Results

The algorithm used to implement the classification criteria was created in Java version 1.8.0 using NetBeans IDE 8.2. The tests were developed on a machine whose processor is Intel (R) Core (TM) i7 CPU 2.60 GHz, 8 GB RAM, with Windows 10 operating system. Table 5 shows the attack detection rate obtained using the two characteristics of the mouse, this being 100 percent, both for the number of real users and for the number of DDoS attacks.

This result shows that with the use of software designed for the detection of attacks and the use of the two characteristics of the user's dynamism, the highest precision rate is reached. It is worth mentioning that the time used by the application to perform the classification was 50 milliseconds. It should be mentioned that in this work it is difficult to identify false positives and negatives, because a dataset with exact data is used, where the interaction of the real user in the requests made is observed. Therefore, when a request is made, this is done through interaction with the mouse, otherwise it is a DDoS attack. However, it can be said that with the use of more features and means of data entry, there could be cases of false positives and negatives. These percentages show the importance of these characteristics for the detection of this type of computer attack.

Table 5. Detection efficiency of DDoS attacks

Users	Real data	Detection criteria	Compliance Rate (%)	Execution time (mls)
Real users	9096	9096	100	90
DDoS attacks	1959	1959	100	

Discussion

The results obtained in the tests carried out show that all DDoS attacks do not have the mouse and right click characteristics, so their detection is 100 percent. The evaluated characteristics (mouse movement and right click) show the dynamism of the user. Therefore, these characteristics allow to differentiate a real request from a computer attack. They use a low cost for the detection of an attack against other characteristics proposed in the literature, because the

algorithm used consumes few resources because of the simplicity of the programmed code. These features also allow you to detect user behaviors that other features do not. For example, mouse operations that had not been proposed in other works aimed at detecting DDOS attacks. It is worth mentioning that there are other characteristics of the dynamism of the user that can be considered for the identification of real users and robots. However, with the use of more features and means of data entry, cases of false positives and negatives would appear. It should also be noted that with the advance in attack detection mechanisms, attackers find new alternatives to circumvent the mechanisms that are being proposed. Therefore, in the future attackers could falsify the variables that measure the characteristics of user behavior, simulating the input data and identifying a robot as a real user.

Conclusion

The review of the state of the art on the variables used in the detection of DDoS attacks at the application layer level shows that 30 variables have been used in the mechanisms published in the last 10 years. In this work we have introduced 24 new features based on the behavior of the web user. They are extracted from the transactionality of the user with the system in real time, therefore, they are computationally economic characteristics due to their easy obtaining. The numerical tests were performed using a dataset of 11055 requests between real users and attacks. The dataset used in the tests contains two of the 24 variables proposed in this paper for the detection of attacks in the application layer. The evaluation of the two variables (mouse movement and right click), using software designed in Java, managed to achieve 100 percent efficiency in the differentiation of real user and robot. Therefore, the right click and mouse movement variables are identified as characteristics of the user's dynamism. Therefore, these variables can be considered for their implementation in DDoS attack detection mechanisms.

Acknowledgment

The first author acknowledges the contributions, made by Professor Angel H. Moreno and the Technical University of Cotopaxi for the assigned doctoral scholarship.

References

Abramson, M., & Aha, D. W. (2013). User authentication from web browsing behavior. *In FLAIRS conference*, 268-273.

Al-Duwairi, B., & Manimaran, G. (2006). Distributed packet pairing for reflector based DDoS attack mitigation. *Computer communications*, 29(12), 2269-2280.

Al-Duwairi, B., Al-Qudah, Z., & Govindarasu, M. (2013). A novel scheme for mitigating botnet-based DDoS attacks. *Journal of Networks*, 8(2), 297.

Beak, C., Chaudhry, J. A., Lee, K., Park, S., & Kim, M. (2007). A novel packet marketing method in DDoS attack detection. *American Journal of Applied Sciences*, 4(10), 741-745.

Chen, R., Park, J. M., & Marchany, R. (2007). A divide-and-conquer strategy for thwarting distributed denial-of-service attacks. *IEEE Transactions on Parallel and Distributed Systems*, 18(5), 577-588.

Chen, S. W., Wu, J. X., Ye, X. L., & Guo, T. (2013). Distributed denial of service attacks detection method based on conditional random fields. *Journal of Networks*, 8(4), 858.

Chen, S., & Song, Q. (2005). Perimeter-based defense against high bandwidth DDoS attacks. *IEEE Transactions on Parallel & Distributed Systems*, (6), 526-537.

Chen, Y., & Hwang, K. (2006). Collaborative detection and filtering of shrew DDoS attacks using spectral analysis. *Journal of Parallel and Distributed Computing*, 66(9), 1137-1151.

Chen, Y., Das, S., Dhar, P., El-Saddik, A., & Nayak, A. (2008). Detecting and Preventing IP-spoofed Distributed DoS Attacks. *IJ Network Security*, 7(1), 69-80.

Chen, Y., Hwang, K., & Ku, W. S. (2007). Collaborative detection of DDoS attacks over multiple network domains. *IEEE Transactions on Parallel & Distributed Systems*, (12), 1649-1662.

Chen, Y., Ma, X., & Wu, X. (2013). DDoS detection algorithm based on preprocessing network traffic predicted method and chaos theory. *IEEE Communications Letters*, 17(5), 1052-1054.

Chonka, A., Singh, J., & Zhou, W. (2009). Chaos theory based detection against network mimicking DDoS attacks. *IEEE Communication Letters*, 13(9), 717-719.

Dick, U., & Scheffer, T. (2016). Learning to control a structured-prediction decoder for detection of HTTP-layer DDoS attackers. *Machine Learning*, 104(2-3), 385-410.

Doron, E., & Wool, A. (2011). Wda: A web farm distributed denial of service attack attenuator. *Computer Networks*, 55(5), 1037-1051.

François, J., Aib, I., & Boutaba, R. (2012). FireCol: a collaborative protection network for the detection of flooding DDoS attacks. *IEEE/ACM Transactions on Networking (TON)*, 20(6), 1828-1841.

Gamboa, H., & Fred, A. L. (2003). An identity authentication system based on human computer interaction behaviour. *In PRIS*. 46-55.

Gavrilis, D., & Dermatas, E. (2005). Real-time detection of distributed denial-of-service attacks using RBF networks and statistical features. *Computer Networks*, 48(2), 235-245.

Ghezzi, C., Pezzè, M., Sama, M., & Tamburrelli, G. (2014). Mining behavior models from user-intensive web applications. *In Proceedings of the 36th International Conference on Software Engineering*, 277-287.

Giralte, L. C., Conde, C., De Diego, I. M., & Cabello, E. (2013). Detecting denial of service by modelling web-server behaviour. *Computers & Electrical Engineering*, 39(7), 2252-2262.

Graepel, T., Candela, J. Q., Borchert, T., & Herbrich, R. (2010). Web-scale bayesian click-through rate prediction for sponsored search advertising in Microsoft's Bing search engine. *Omnipress*.

Huang, C., Wang, J., Wu, G., & Chen, J. (2014). Mining Web User Behaviors to Detect Application Layer DDoS Attacks. *JSW*, 9(4), 985-990.

Johnson Singh, K., Thongam, K., & De, T. (2016). Entropy-based application layer DDoS attack detection using artificial neural networks. *Entropy*, 18(10), 350.

Kang, H. S., & Kim, S. R. (2014). sShield: small DDoS defense system using RIP-based traffic deflection in autonomous system. *The Journal of Supercomputing*, 67(3), 820-836.

Kang, S. H., Park, K. Y., Yoo, S. G., & Kim, J. (2013). DDoS avoidance strategy for service availability. *Cluster computing*, 16(2), 241-248.

Kim, Y., & Kim, I. (2014). Involvers' behavior-based modeling in cyber targeted attack. *Proceedings of SECURWARE*.

Kim, Y., Lau, W. C., Chuah, M. C., & Chao, H. J. (2006). PacketScore: A statistics-based packet filtering scheme against distributed denial-of-service attacks. *IEEE transactions on dependable and secure computing*, 3(2), 141-155.

Kulkarni, A., & Bush, S. (2006). Detecting distributed denial-of-service attacks using Kolmogorov complexity metrics. *Journal of Network and Systems Management*, 14(1), 69-80.

Kumar, P. A. R., & Selvakumar, S. (2011). Distributed denial of service attack detection using an ensemble of neural classifier. *Computer Communications*, 34(11), 1328-1341.

Kumar, P. A. R., & Selvakumar, S. (2013). Detection of distributed denial of service attacks using an ensemble of adaptive and hybrid neuro-fuzzy systems. *Computer Communications*, 36(3), 303-319.

Lee, F. Y., & Shieh, S. (2005). Defending against spoofed DDoS attacks with path fingerprint. *Computers & Security*, 24(7), 571-586.

Lee, K., Kim, J., Kwon, K. H., Han, Y., & Kim, S. (2008). DDoS attack detection method using cluster analysis. *Expert systems with applications*, 34(3), 1659-1665.

Lee, S. M., Kim, D. S., Lee, J. H., & Park, J. S. (2012). Detection of DDoS attacks using optimized traffic matrix. *Computers & Mathematics with Applications*, 63(2), 501-510.

Li, L., & Lee, G. (2005). DDoS attack detection and wavelets. *Telecommunication Systems*, 28(3-4), 435-451.

Lichman M.. UCI Machine Learning Repository *[http://archive.icc.uci.edu/ml]*. Irvine, CA: University of California, School of Information and Computer Science, 2013.

Liu, H., Sun, Y., & Kim, M. S. (2011). A Scalable DDoS Detection Framework with Victim Pinpoint Capability. *JCM*, 6(9), 660-670.

Lu, W. Z., Gu, W. X., & Yu, S. Z. (2009). One-way queuing delay measurement and its application on detecting DDoS attack. *Journal of Network and Computer Applications*, 32(2), 367-376.

Luo, H., Lin, Y., Zhang, H., & Zukerman, M. (2013). Preventing DDoS attacks by identifier/locator separation. *IEEE network*, 27(6), 60-65.

Luo, J., Yang, X., Wang, J., Xu, J., Sun, J., & Long, K. (2014). On a mathematical model for low-rate shrew DDoS. *IEEE Trans. Information Forensics and Security*, 9(7), 1069-1083.

Ma, X., & Chen, Y. (2014). DDoS detection method based on chaos analysis of network traffic entropy. *IEEE Communications Letters*, 18(1), 114-117.

Meenakshi, S., & Srivatsa, S. K. (2007). A distributed framework with less false positive ratio against distributed denial of service attack. *Information Technology Journal*, 6(8), 1139-1145.

Mirkovic, J., & Reiher, P. (2005). D-WARD: A source-end defense against flooding denial-of-service attacks. *IEEE transactions on Dependable and Secure Computing*, 2(3), 216-232.

Nguyen, H. V., & Choi, Y. (2010). Proactive detection of DDoS attacks utilizing k-NN classifier in an anti-DDoS framework. *International Journal of Electrical, Computer, and Systems Engineering*, 4(4), 247-252.

Rahmani, H., Sahli, N., & Kamoun, F. (2012). DDoS flooding attack detection scheme based on F-divergence. *Computer Communications*, 35(11), 1380-1391.

Ranjan, S., Swaminathan, R., Uysal, M., Nucci, A., & Knightly, E. (2009). DDoS-shield: DDoS-resilient scheduling to counter application layer attacks. *IEEE/ACM Transactions on networking*, 17(1), 26-39.

Sachdeva, M., & Kumar, K. (2014). A traffic cluster entropy based approach to distinguish DDoS attacks from flash event using DETER testbed. *ISRN Communications and Networking*, 2014.

Salmeron-Majadas, S., Santos, O. C., & Boticario, J. G. (2014). An evaluation of mouse and keyboard interaction indicators towards non-intrusive and low cost affective modeling in an educational context. *Procedia Computer Science*, 35, 691-700.

Saravanan, R., Shanmuganathan, S., & Palanichamy, Y. (2016). Behavior-based detection of application layer distributed denial of service attacks during flash events. *Turkish Journal of Electrical Engineering & Computer Sciences*, 24(2), 510-523.

Seo, D., Lee, H., & Perrig, A. (2013). APFS: Adaptive probabilistic filter scheduling against distributed denial-of-service attacks. *Computers & Security*, 39, 366-385.

Shen, C., Cai, Z., Guan, X., Du, Y., & Maxion, R. A. (2013). User authentication through mouse dynamics. *IEEE Transactions on Information Forensics and Security*, 8(1), 16-30.

Spyridopoulos, T., Karanikas, G., Tryfonas, T., & Oikonomou, G. (2013). A game theoretic defence framework against DoS/DDoS cyber attacks. *Computers and Security*, 38, 39-50.

Stevanovic, D., & Vlajic, N. (2014). Application-layer DDoS in dynamic Web-domains: Building defenses against next-generation attack behavior. *In Communications and Network Security (CNS)*, 2014 IEEE Conference on (pp. 490-491). IEEE.

Udhayan, J., & Babu, M. R. (2013). Deteriorating distributed denial of service attack by recovering zombies using penalty scheme. *Journal of Computer Science*, 9(11), 1618.

Urban, R. J. (2015). U.S. Patent No. 9,141,976. *Washington, DC: U.S. Patent and Trademark Office.*

Varalakshmi, P., & Selvi, S. T. (2013). Thwarting DDoS attacks in grid using information divergence. *Future Generation Computer Systems*, 29(1), 429-441.

Wang, D., Chang, G., Feng, X., & Guo, R. (2008). Research on the detection of distributed denial of service attacks based on the characteristics of IP flow. *In IFIP International Conference on Network and Parallel Computing*, 86-93. Springer, Berlin, Heidelberg.

Wang, F., Wang, H., Wang, X., & Su, J. (2012). A new multistage approach to detect subtle DDoS attacks. *Mathematical and Computer Modelling*, 55(1-2), 198-213.

Wang, H., Jin, C., & Shin, K. G. (2007). Defense against spoofed IP traffic using hop-count filtering. *IEEE/ACM Transactions on Networking (ToN)*, 15(1), 40-53.

Wang, Y., & Sun, R. (2014). An IP-traceback-based packet filtering scheme for eliminating DDoS attacks. *Journal of Networks*, 9(4), 874.

Wu, X., & Chen, Y. (2013). Validation of chaos hypothesis in NADA and improved DDoS detection algorithm. *IEEE Communications Letters*, 17(12), 2396-2399.

Xiang, Y., & Zhou, W. (2005). Mark-aided distributed filtering by using neural network for DDoS defense. *In GLOBECOM'05: IEEE Global Telecommunications Conference*, 28 November-2 December 2005 St. Louis, Missouri, USA, discovery past and future, 1701-1705. IEEE Globecom.

Xiang, Y., Li, K., & Zhou, W. (2011). Low-rate DDoS attacks detection and traceback by using new information metrics. *IEEE transactions on information forensics and security*, 6(2), 426-437.

Xiao, B., Chen, W., & He, Y. (2006). A novel approach to detecting DDoS attacks at an early stage. *The Journal of Supercomputing*, 36(3), 235-248.

Xiao, P., Qu, W., Qi, H., & Li, Z. (2015). Detecting DDoS attacks against data center with correlation analysis. *Computer Communications*, 67, 66-74.

Xie, Y., & Yu, S. Z. (2009). Monitoring the application-layer DDoS attacks for popular websites. *IEEE/ACM Transactions on Networking (TON)*, 17(1), 15-25.

Yaar, A., Perrig, A., & Song, D. (2005). FIT: Fast internet traceback. In INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. *Proceedings IEEE*, Vol. 2, pp. 1395-1406. IEEE.

Yan, R., & Zheng, Q. (2009). Using Renyi cross entropy to analyze traffic matrix and detect DDoS attacks. *Information Technology Journal*, 8(8), 1180-1188.

Yau, D. K., Lui, J., Liang, F., & Yam, Y. (2005). Defending against distributed denial-of-service attacks with max-min fair server-centric router throttles. *IEEE/ACM Transactions on Networking (TON)*, 13(1), 29-42.

Zhang, C., Cai, Z., Chen, W., Luo, X., & Yin, J. (2012). Flow level detection and filtering of low-rate DDoS. *Computer Networks*, 56(15), 3417-3431.

Zhenwei, Y. (2011). Intrusion detection: A machine learning approach. *World Scientific*, Vol. 3.

Zhou, W., Jia, W., Wen, S., Xiang, Y., & Zhou, W. (2014). Detection and defense of application-layer DDoS attacks in backbone web traffic. *Future Generation Computer Systems*, 38, 36-46.

Zolotukhin M., Kokkonen T., Hämäläinen T., Siltanen J., (2016). On application layer DDoS attack detection in high-speed encrypted networks. *International Journal of Digital Content Technology and its Applications (JDCTA)*, 10(5), p. 14 - 33, Advanced Institute of Convergence IT.

Bibliographic information of this paper for citing:

Bravo, Silvia, & Mauricio, David (2018). "Distributed denial of service attack detection in application layer based on user behavior." *Webology*, 15(2), Article 171. Available at: http://www.webology.org/2018/v15n2/a171.pdf