New Features of User’s Behavior to Distributed Denial of Service Attacks Detection in Application Layer

https://doi.org/10.3991/ijoe.v14i12.9439

Silvia Bravo (✉)
Technical University of Cotopaxi, Latacunga, Ecuador
silvia. bravom@utc.edu.ec

David Mauricio
National University of San Marcos, Lima, Peru

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%. 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%. 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—DDoS, user’s behavior, application layer, attack detection

1 Introduction

The detection of 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 [1]-[49] and at the application level [50]-[58]. The highest detection rate obtained to date is 99.4%, and has been achieved by implementing a network-level method [1]. 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% [50], 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 [50], [58], 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 [55]. 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.

## 2 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 [59], [60] provided by the request flows [61]. 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 [62]. There are also classification criteria based on Soft computing techniques and its hydrological approach [1]. 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.

Table 1 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 [51]. 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 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 [1]. 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 1 shows 30 characteristics that
are used in the detection of attacks.

The highest detection rate obtained to date is 98.5% and has been achieved using
software generated in Python using the Intrusion Detection System (IDS) technique
[50]. 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.

![Fig. 1. Execution and detection of DDoS attack in the application layer](http://www.i-joe.org)

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access pattern</td>
<td>Access pattern is constantly repeated, develop a frequent path detector which involves checking the requests of the complete flow.</td>
<td>[53]</td>
</tr>
<tr>
<td>Average length of query strings of client</td>
<td>Average of consultations made by clients.</td>
<td>[56]</td>
</tr>
<tr>
<td>Click number of web objects</td>
<td>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.</td>
<td>[55]</td>
</tr>
<tr>
<td>Client legitimacy</td>
<td>The legitimacy of a user sending an enormous number of requests is checked against the known client clusters.</td>
<td>[57]</td>
</tr>
<tr>
<td>Duration of the conversation</td>
<td>Conversations initiated by one client to the destination socket during some short time interval.</td>
<td>[50]</td>
</tr>
</tbody>
</table>

Table 1. Features of application layer
<table>
<thead>
<tr>
<th>Feature Description</th>
<th>Explanation</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Entropy of request type (GET/POST/OTHER)</td>
<td>The fractions of request types per connection (GET, POST, or OTHER).</td>
<td>[56]</td>
</tr>
<tr>
<td>Entropy of the requests</td>
<td>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.</td>
<td>[58]</td>
</tr>
<tr>
<td>Flow similarity</td>
<td>Flow similarity is considered as a key parameter for discriminating between legitimate and illegitimate flows and a few works.</td>
<td>[57]</td>
</tr>
<tr>
<td>Fraction of connections for domain that accepts any version of English</td>
<td>Connection (e.g., en-us) in Accept-Language.</td>
<td>[56]</td>
</tr>
<tr>
<td>Fraction of connections of client that request the most frequent resource path</td>
<td>A client accesses and also count how often each client requests the currently most common path on the domain.</td>
<td>[56]</td>
</tr>
<tr>
<td>Access pattern</td>
<td>Access pattern is constantly repeated, develop a frequent path detector which involves checking the requests of the complete flow.</td>
<td>[53]</td>
</tr>
<tr>
<td>Average length of query strings of client</td>
<td>Average of consultations made by clients.</td>
<td>[56]</td>
</tr>
<tr>
<td>HTTP GET request count</td>
<td>The operation of HTTP starts with a client by sending a request to the server in the form of a request method.</td>
<td>[58]</td>
</tr>
<tr>
<td>IP address</td>
<td>Source IP addresses, we are able to classify them into different traffic.</td>
<td>[54]</td>
</tr>
<tr>
<td>Maximal, minimal and average packet size</td>
<td>Average of these packet numbers and the mutual information of the fast Fourier Transform.</td>
<td>[50]</td>
</tr>
<tr>
<td>Maximal, minimal and average size of TCP window</td>
<td>Number of packets received at the current time horizon and at the previous one.</td>
<td>[50]</td>
</tr>
<tr>
<td>Maximal, minimal and average time to live (TTL)</td>
<td>Account time intervals between subsequent packets of the same flow.</td>
<td>[50]</td>
</tr>
<tr>
<td>Number of bytes sent in 1 second</td>
<td>Packets in bytes sent from the client to the server and from the server to the client.</td>
<td>[50]</td>
</tr>
<tr>
<td>Number of different resource paths of client</td>
<td>It includes the number of different resource paths that client has accessed.</td>
<td>[56]</td>
</tr>
<tr>
<td>Number of packets sent in 1 second</td>
<td>Packets sent from the client to the server and from the server to the client.</td>
<td>[50]</td>
</tr>
<tr>
<td>Number of request</td>
<td>Requests for the currently open windows and whether the number of requests for an open window.</td>
<td>[53]</td>
</tr>
<tr>
<td>Number of users</td>
<td>Set of real users accessing a server.</td>
<td>[53]</td>
</tr>
<tr>
<td>Percentage of encrypted packets with different properties</td>
<td>Since the traffic may be encrypted it is not always possible to define what web page these clients request.</td>
<td>[50]</td>
</tr>
<tr>
<td>Percentage of packets with different TCP flags</td>
<td>As it was mentioned in the previous section, in this study, we concentrate on the traffic transferred over TCP.</td>
<td>[50]</td>
</tr>
<tr>
<td>Session's requests</td>
<td>Requests for the currently open windows and whether the number of requests for an open window.</td>
<td>[52]</td>
</tr>
<tr>
<td>Sum of incoming payload of all clients of domain</td>
<td>If requests from attacking IP addresses were to be processed, inspected, and filtered based on the individual payload.</td>
<td>[56]</td>
</tr>
<tr>
<td>Sum of outgoing payload of all clients of domain</td>
<td>If requests from attacking IP addresses were to be processed, inspected, and filtered based on the individual payload.</td>
<td>[56]</td>
</tr>
</tbody>
</table>
3 Feature of user behavior

3.1 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 [63]. The authentication of a user by means of his behavior has been a task studied from the point of view of information security [64]. Therefore, in order to avoid access by unauthorized users, several investigations [63]-[68] 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) [64] for the identification of users.

Table 2 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 [69]. 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 [68]. On the other hand, the robots are generated by specialized software to make the largest number of requests to a system [1], without the use of any peripheral.

It is worth mentioning that the characteristics presented in Table 2, 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>
<thead>
<tr>
<th>ID</th>
<th>Mouse Features</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>Single-click</td>
<td>[68]</td>
</tr>
<tr>
<td>M2</td>
<td>Double-click</td>
<td></td>
</tr>
<tr>
<td>M3</td>
<td>Movement offset</td>
<td></td>
</tr>
<tr>
<td>M4</td>
<td>Speed curve against time</td>
<td></td>
</tr>
<tr>
<td>M5</td>
<td>Acceleration curve against time</td>
<td></td>
</tr>
</tbody>
</table>
3.2 Features capture

Table 3 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 for this [68]-[71].

Table 3. Extraction Features of the mouse

<table>
<thead>
<tr>
<th>ID</th>
<th>Extraction Method</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>M1</td>
<td>Windows application (written in C#)</td>
<td>[68]</td>
</tr>
<tr>
<td>M2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M7</td>
<td>Java (kSquared.de library)</td>
<td>[69]</td>
</tr>
<tr>
<td>M8</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M10</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M11</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M12</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M13</td>
<td></td>
<td></td>
</tr>
<tr>
<td>M14</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
3.3 Classification algorithm

Figure 2 shows the classification algorithm that allows the identification of DDoS attacks by means of mouse features. 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 features, 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
```

Fig. 2. Classification algorithm of real users and robots

4 Numerical experiments

4.1 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 Lichman [12], and which is discussed in section 4.2. 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, the same one mentioned in section 3.3, 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.

Fig. 3. Validation environment architecture

4.2 Dataset

The dataset used in this work for the validation process of the classification algorithm was created by Lichman [72]. 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.

4.3 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.
4.4 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 4 shows the attack detection rate obtained using the two characteristics of the mouse, this being 100%, 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 4. Detection efficiency of DDoS attacks

<table>
<thead>
<tr>
<th>Users</th>
<th>Real data</th>
<th>Detection criteria</th>
<th>Compliance Rate (%)</th>
<th>Execution time (mls)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Real user</td>
<td>9096</td>
<td>9096</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>DDoS attacks</td>
<td>1959</td>
<td>1959</td>
<td>100</td>
<td>90</td>
</tr>
</tbody>
</table>

4.5 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%. 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 (keyboard). 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.

5 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% 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.

6 Acknowledgment

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

7 References


8 Authors

Silvia Bravo was born in Latacunga, Ecuador. She graduated from the Technical University of Cotopaxi in 2007, where she received the title of “Computer Science”. She is currently pursuing a Ph.D. from National University of San Marcos within the Doctoral Program of “Computer and System”. She is currently working as a professor and researcher at the Faculty of Engineering Science, in the Technical University of Cotopaxi. Her research activity is mainly focused on the software development and informatics security.

David Mauricio was born in Lima, Peru. He graduated from the National San Marcos University in 1987, where he received the title of “Computer Science”. He obtained the title of “Master in Mathematics Applied” from the Federal University of Rio de Janeiro, Brazil, in 1991. In 1994, he obtained the title of “Doctor in Systems Engineering” from the Federal University of Rio de Janeiro. He is currently working as a professor at the Faculty of Systems Engineering, in the National Mayor de San Marcos University and scientific consultant in National Council for Science and Technology (CONCYTEC). His research activity is mainly focused on the Combinatorial optimization, Designs and analysis of algorithms, Heuristics search, Metaheuristics, Mathematical programming, Expert systems, Data mining, Artificial intelligence.

Article submitted 29 August 2018. Resubmitted 14 and 16 September 2018. Final acceptance 14 October 2018. Final version published as submitted by the authors.