TÜBİTAK

Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2016) 24: 510 – 523
© TÜBİTAK
doi:10.3906/elk-1308-188

Behavior-based detection of application layer distributed denial of service attacks during flash events

Renukadevi SARAVANAN[*], **Saraswathi SHANMUGANATHAN**, **Yogesh PALANICHAMY**
Department of Information Science and Technology, College of Engineering, Anna University, Chennai,
Tamil Nadu, India

Received: 30.08.2013 • **Accepted/Published Online:** 17.12.2013 • **Final Version:** 05.02.2016

Abstract: Distributed denial of service (DDoS) attacks are ever threatening to the developers and users of the Internet. DDoS attacks targeted at the application layer are especially difficult to be detected since they mimic the legitimate users' requests. The situation becomes more serious when they occur during flash events. A more sophisticated algorithm is required to detect such attacks during a flash crowd. A few existing works make use of flow similarity for differentiating flash crowds and DDoS, but flow characteristics alone cannot be used for effective detection. In this paper, we propose a novel mechanism for discriminating DDoS and flash crowds based on the combination of the parameters reflecting their behavioral differences. Flow similarity, client legitimacy, and web page requested are identified as the principal parameters and are used together for effective discrimination. The proposed mechanism is implemented on resilient proxies in order to protect the server from direct flooding and to improve the overall performance. The real datasets are used for simulation, and the results are presented to evaluate the performance of the proposed system. The results show that the proposed mechanism does effective detection with fewer false positives and false negatives.

Key words: DDoS, flash crowd, flow similarity, security

1. Introduction

Detecting a distributed denial of service (DDoS) attack is a great challenge for the defenders since the attackers are targeting the victim with more sophisticated techniques. In particular, the application-level DDoS attacks imitate the request patterns of the legitimate users, thereby making detection tougher. The detection of such attacks is very critical when they occur during a flash crowd.

Flash crowds are legitimate users who flood a website over a relatively short period of time. This happens due to the sudden increase in the popularity of a website with the occurrence of any special events like breaking news, release of a popular products, announcement of results, and so on. Figure 1a shows the traffic profile caused by a flash crowd on the 90th day of the FIFA World Cup in 1998 (http://ita.ee.lbl.gov/html/contrib/WorldCup.html). Flash crowds try to access the server simultaneously, resulting in unexpected flooding. The effect of flash crowds is real and acute for servers hosting popular websites.

On the other hand, DDoS attacks arise from illegitimate users with a motive to deny the service provided by the server by sending an overwhelming number of requests. Examples of attacks include TCP SYN flooding, HTTP flooding, and so on. For example, a distributed denial of service against the Santa Cruz Operation (SCO) Group was executed by the Mydoom worm (http://www.cert.org/incident_notes/IN-

[*]Correspondence: renusaravanan@yahoo.co.in

2004-01.html). Figure 1b shows the attack traffic caused by the DDoS against SCO in December 2003 (http://www.caida.org/research/security/sco-dos/). Recently, attacks are launched with the help of botnets and target the application level. Botnets are a group of compromised hosts (bots), which are controlled by the bot master. To launch an attack, the bot master issues a command to all its bots and the live bots (the bots that are currently active) in turn flood the victim with an enormous number of requests to take the system down. The attackers try to bypass the defense mechanisms by mimicking the traffic patterns of legitimate clients. When such a type of attack occurs during flash events, it is referred to as a flash crowd attack.

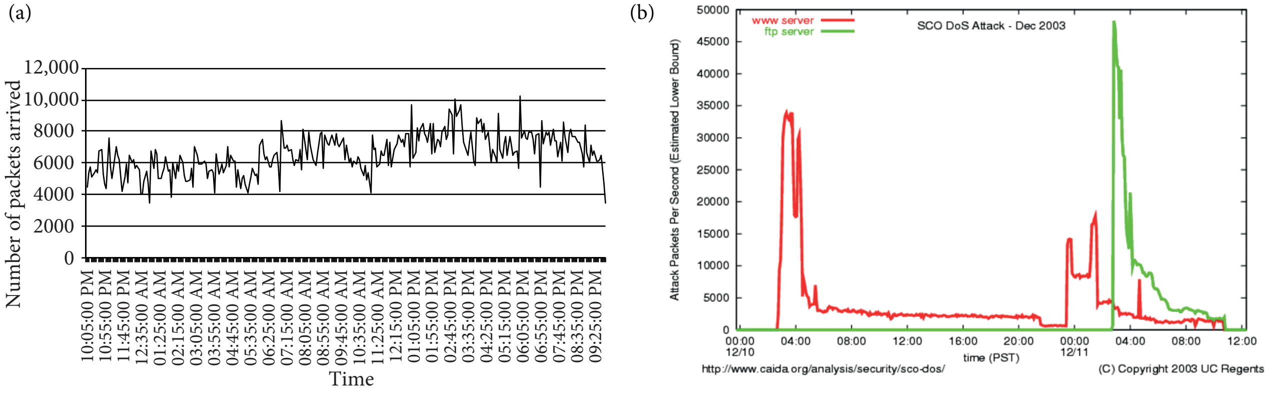

Figure 1. a) Flash crowd on day 90 of 1998 FIFA World Cup. b) DDoS against SCO on 11 December 2003.

At the outset, both the flash crowd and application-level DDoS attacks are unstable, bursty, and voluminous in terms of traffic phenomena. Even then, there exist some key differences between them, such as their access intents, the distributions of their source IP addresses, and the increased and decreased speeds of traffic between them [1]. Both DDoS attacks and flash crowds have similar impacts on the server, resulting in partial or complete failure. It is up to the web server to identify and serve as many legitimate requests as possible during flash events. In order to differentiate DDoS attacks from flash crowds, the behavioral difference between them must be analyzed by considering each of their individual properties.

This paper proposes a mechanism to discriminate a high-rate application layer DDoS attack from a flash crowd. To make the discrimination effective, the individual properties of flash crowds and DDoS attacks are analyzed, and the key parameters are identified and are used as the key components in differentiating DDoS during a flash crowd. The first and foremost parameter is the flow similarity among the suspicious sessions. Since the attack tools installed in the sources remain the same for all the bots belonging to a botnet, the request flows originating from the attack sources are found to be more similar. In contrast, the request flows from the legitimate human users are mostly random and do not follow any specific pattern among them. Second, the web page that the request is referring to plays an important role. It is assumed that the human users access the hot pages most of the time, and in particular during flash events, but the bots do not. Thirdly, user legitimacy is also considered for making decision. If the user is a well-known (frequent) user then a certain level of consideration can be given during a flash crowd attack. The detection is made based on the combination of the above mentioned 3 key parameters and the results are presented at the end to evaluate the efficiency of the proposed work.

The rest of the paper is organized as follows: Section 2 reviews the related work of our research. Section 3 analyzes the behavioral difference between a flash crowd and DDoS and identifies the key parameters for discrimination. Section 4 describes in detail the working principle of offline and online phases involved in the detection process. Section 5 analyzes the performance of the proposed mechanism with the existing techniques based on the simulation results. Section 6 concludes the proposed work. Finally, Section 7 presents the possible ideas of our research in the future.

2. Related work

Much research with respect to DDoS mitigation has been carried out and a variety of defense mechanisms have been developed with different perspectives. Initially the attack focus was at the network layer, aiming at exhausting the network bandwidth by pumping a huge number of packets. Extensive numbers of defense mechanism against network layer DDoS have been proposed. For example, Zhang and Dasgupta [2] proposed a hardened network system where intelligent routers are used for detection of attacks and trace-back of attack packets to its source. El-Moussa et al. [3] introduced a collaborative active routers-based mechanism to provide a distributed defense. Wang et al. [4] proposed a solution using hop count value. A received packet is dropped if there exists a wide difference between its hop count and the value stored in the previously built table. Keromytis et al. [5] proposed an overlay network through which the legitimate traffic is sent. Li et al. [6] proposed an information distance-based detection mechanism in which Renyi divergence is used to measure the difference between the legitimate and illegitimate flows.

To mitigate the transport layer attack, Limwiwatkul and Rungsawangr [7] proposed a defense mechanism in which the TCP header fields are used for identifying the normal and abnormal flows. Cabrera et al. [8] use management information base (MIB) data for early detection of an attack. The parameters indicating the unusual packet and routing statistics are extracted from routers and are used for detection.

Recently, attackers are targeting the application layer in order to make the detection tougher. To defend against such attacks, Wang et al. [9] proposed a method based on relative entropy in which the click ratio of the web object is used as a key component to detect the suspicious sessions. Liu and Chang [10] proposed a defense against tilt DDoS attack scheme. The user's features throughout a connection session are monitored and differentiated services are provided based on their behavior. Yu et al. [11] proposed a lightweight mechanism that uses trust and license for differentiating legitimate users from attackers. Srivatsa et al. [12] proposed a mechanism involving admission control to limit the number of concurrent clients getting service from the online service and congestion control, to allocate more resources to the admitted clients.

However, the available techniques fail to detect DDoS attacks effectively during flash crowds because the detection criteria change from one situation to another. General DDoS detection techniques employ either statistically based defense mechanisms, which rely on IP header information to discriminate legitimate traffic from abnormal traffic, or heuristically based defense mechanisms, which rely on a threshold. However, none of them are effective during flash events since the legitimate and illegitimate flows share many similar characteristics. The legitimate and illegitimate requests differ only in the intent, not in the content. Pattern-based detection will be more suitable during flash events due to the programmed attack tools driven by the attack sources (bots). A more sophisticated technique capable of identifying the key patterns and discriminating between DDoS attacks and flash crowds is required for the recent flash crowd attacks. In particular, a few works [1,13–18] are available on discriminating DDoS from flash crowds. The most popular among them is the use of graphical puzzles like CAPTCHAs to differentiate human users from bots [15]. However, they involve delay and require user response. Xie and Yu [19] proposed an anomaly detector for detecting and differentiating DDoS attacks from flash crowds based on the dynamic changes in the users' browsing behavior. Jung et al. [13] proposed a way to distinguish flash crowds and DDoS attacks using network-aware clustering.

Flow similarity is considered as a key parameter for discriminating between legitimate and illegitimate flows and a few works [1,17,18] were proposed based on it. Yu et al. [17] used flow correlation coefficients as a metric for discriminating between flash crowds and DDoS attacks. Yu et al. [18] proposed a mechanism in which 3 different distance metrics, namely the Jeffrey distance, the Sibson distance, and the Hellinger distance, are used to measure the flow similarity, and they concluded that out of the 3 metrics, the Sibson distance was the most suitable for their case. Li et al. [1] used 2 probability metrics to measure the distance between any 2 flows. The total variation metric mainly measures the difference of 2 discrete probability distributions, and the Bhattacharyya metric is then mainly used to measure the similarity of 2 discrete probability distributions. The combined results are used to distinguish the DDoS attacks from flash crowds. However, flow similarity alone cannot be used as a tool for the effective differentiation of DDoS attacks and flash crowds.

From all the above works, it is observed that no single key parameter/dimension is suitable for efficient detection of application level attacks, but the combination of multiple key parameters may end up with good results. The proposed work aims at providing such a solution at the victim end to deal with application level DDoS attacks during flash events.

2.1. Behavioral analysis

In this paper, a novel mechanism to detect DDoS attacks from flash crowds is proposed. The behavioral differences between DDoS attacks and flash crowds are analyzed and the following key parameters are identified and used for differentiating them. The importance of each identified parameter is discussed in this section.

2.2. Flow characteristics

Most of the recent DDoS attacks are launched with the help of botnets. The studies on botnets [20–22] imply that the attack tools are prebuilt programs, which are usually the same for all the compromised hosts that belong to a single botnet. The master attacker issues a command to all the bots to start an attack session. Since all the attack flows are driven by a single master, it is observed that the attack flows are much more similar than those of legitimate users (including flash crowds). A flow is defined as a group of web page requests to the victim server. Once flooding is detected at the victim end, the suspicious sessions are identified, and the similarity among every 2 suspicious session flows is computed and action is taken accordingly. In this paper, we propose the Hellinger distance as the metric to measure the distance between the suspicious flows. The Hellinger distance is used in this paper in order to overcome the asymmetric properties of other popular distance metrics like Kullback–Leibler information divergence [23] and Bregman distance (http://en.wikipedia.org/wiki/Bregman_divergence). The symmetric property is most important in our work since the distance between 2 suspicious flows computed at either end must be identical (i.e. $D_H(p, q) = D_H(q, p)$) for the same pair of flows for making a decision. Any symmetric distance metric can be applied to measure the flow similarity. The results of the similarity measure obtained from the existing techniques and from the Hellinger distance of the proposed work are presented in Section 5.1. The results of similarity among the flows obtained from different symmetric distance metrics are much closer. To improve the detection accuracy, the final discrimination is further fine-tuned by the other 2 parameters, pages referred and client legitimacy, in our technique. In general, the distance among the flows is sufficiently small in the case of DDoS attack generated from a single botnet and more in case of a flash crowd.

2.3. Page referred

In the case of an application level DDoS attack, the attack packets are in the form of web page requests. The web mining results show that about 90% of the web accesses are made on 10% of the web page [24], and mostly the legitimate users often visit the hot web pages that they are interested in. The pages that are accessed frequently by human users become “hot pages” and the rest of the pages are considered as “cold pages”, which are accessed by only a few people. During a flash crowd, the human users always try to access the hot pages, but the attackers, the bots, access pages randomly, irrespective of their importance. With this observation, we find that this parameter will also be helpful for differentiating between the flash crowd and the attack to some extent. To classify the web pages of a victim server as hot or cold, the number of references made to each and every web page of the victim server in the past is calculated during a nonattack case for a specific period of time. Then an appropriate threshold is fixed for classification. This can be done offline at the victim end, as discussed in Section 4.1.3. The web page with a reference count higher than the defined threshold is considered as the hot page and the rest of the pages are considered cold pages. In addition, the pages that include information relating to the flash events are by default considered as hot pages. This is done whenever an event is uploaded to the server.

2.4. Client legitimacy

Client characteristics play a major role in distinguishing flash crowd from DDoS attacks. Jung et. al. [13] distinguished DDoS and normal flash crowds using 2 main properties: 1) In the case of a DDoS attack, the request rate increases from a small group of clients, but in contrast, the number of clients increases drastically during a flash crowd. 2) The attackers originate mostly from new client clusters, whereas a flash crowd (genuine clients) originates mostly from known client clusters.

Research studies [1,13] also show that the distributions of the client’s IP addresses are quite different between flash crowd and DDoS attacks. The client’s IP addresses in flash crowds are highly distributed since users from all parts of the network may be interested in getting information on the occurrence of any special event. However, in DDoS attacks, the attackers are the compromised hosts and their distribution of IP addresses is concentrated due to the limited attackers or zombies.

Considering the above mentioned properties, users are classified based on their previous access behavior. The users who have frequently accessed the website in the past are classified as known users and users with no or very low reference count on a website are considered unknown or new. During the detection process, the incoming clients can be easily identified as known or unknown by looking into the known client list. Longest prefix match [25] is used for easy lookup. In this way, the known regular (legitimate) users are given importance during the flash event.

3. Detection mechanism

The incoming sessions are monitored continuously and if the overall incoming rate exceeds the predefined threshold, then there is a possibility of flooding, which may be due to a flash crowd or DDoS attack or both together. In such a situation, each session is marked as suspicious or normal based on the number of incoming requests. The identified suspicious sessions are then examined from different perspectives to conclude whether they are an attack or a genuine flash crowd.

The detection mechanism is implemented on the proxy system (placed just before the server) in order to protect the server from direct flooding. As previously mentioned, the detection is done based on 3 aspects: flow similarity between the suspicious sessions, legitimacy of the user, and web pages referred by the request. Out of the 3 parameters, flow similarity plays a vital role in discriminating the attacks from legitimate flows because it is the core and standalone parameter to identify the existence of an attack at the earliest time. It is given a high priority with 0.6 weightage on a scale of 0–1. The other 2 parameters, client legitimacy and paged referred, are given equal weightage at the next level for further decision-making. They are assigned a weightage of 0.2 each, since neither can individually or together detect the attack perfectly during flash events. This is due to the possibility that well-known users may be compromised to act as bots or the attackers may intelligently target the hot pages to hide themselves. At the same time, they are considered as value-added parameters that can be effectively used for fine-tuning the discrimination decision upon detection of the flow similarity. Based on the 3 parameters, the differentiation is made and the corresponding action is taken. Since the number of users visiting the website is higher during a flash crowd, resilient proxies are deployed for detection. The proxies forward the genuine requests to the server, which in turn responds to the clients.

The following assumptions are made in our proposed work to make our analysis clear:

- Only 1 botnet targets the victim at any given time.
- Bots target the victim with high-rate (flooding) attacks.
- The system log that is used in the offline phase is secure and is not intercepted by the attackers.
- The number of active (live) bots is much lower than the number of legitimate users.

3.1. Offline processing

This section describes the preprocessing steps (threshold computation, client classification, and page classification) that are done offline and are used for the detection process. The system log captured under normal conditions is taken as input, and various thresholds are computed and certain classifications are made based on it.

3.2. Threshold computation

In the proposed work, the thresholds for the overall request rate, request rate per session, page classification, and client classification are used. As an example, the computation of the threshold for the overall request rate is described below. All the other thresholds are calculated in a similar way.

The threshold is fixed by analyzing the web log of the server during a nonattack case. Let T and Δt be the sampling period and the sampling interval (unit time in this case), respectively. The total number of incoming requests during the sampling intervals within the sampling period is recorded as r$_1$, r$_2$, ..., r$_n$, where n is the number of samples (i.e. n = T / Δt).

Let μ be the mean of the incoming requests, given by:

$$\mu = \left(\sum_i r_i\right)/n. \tag{1}$$

The threshold value (th) is fixed based on the mean, standard deviation (σ), and maximum absolute deviation from the median (m) as given in Eqs. (2)–(4).

The maximum absolute deviation (D$_{\max}$) is computed as

$$D_{\max} = max(D_1, D_2, ..., D_n), \tag{2}$$

where D$_i$ = $|$ r$_i - m|$; 1 $\leq i \leq n$ represents the absolute deviation of each sample.

The maximum allowable deviation is given by

$$\delta = D_{\max} - \sigma. \tag{3}$$

Therefore, the threshold is fixed to "th" number of requests per unit time:

$$th = \mu + \delta. \tag{4}$$

3.3. Client classification

The clients of the target server are retrieved from the system log during a nonattack case and they are classified as known clients and unknown/new clients based on their visiting history. The reference count (number of references/accesses made on the server's website) for each user who accessed the server during the nonattack case is recorded from the system log for a specific period of time T. The threshold is computed as discussed in Section 4.1.1. The clients with reference counts greater than the threshold value are termed as known clients and the rest are the unknown/new clients. The result of the classification is used in the detection process. When the server suspects flooding, the legitimacy of a user sending an enormous number of requests is checked against the known client clusters and the decision is made accordingly.

3.4. Page classification

The web pages of the server's website are grouped into 2 categories: hot pages and cold pages. The hot pages are the most frequently referred pages and the cold pages are the rarely used pages. The grouping can be done by thoroughly analyzing the web log of the server under the normal case for a specific period of time and thereby fixing a threshold for classification. The procedure involved during the offline processing is summarized below.

Let N be the number of web pages of the victim server, T be the sampling time, and R_i be the number of references made on page i during the sampling interval Δt within T. Then we have $\{R_1, R_2, ..., R_N\}$ representing the reference count of each page during a nonattack situation. From the above set, the threshold is computed in the same way as already discussed in this section. The pages with a reference count higher than the threshold are classified as hot pages and the rest of the pages are classified as cold pages.

3.5. Online processing

The incoming sessions are continuously monitored, and the sessions having a higher number of requests are marked as suspicious and examined within 3 prescribed dimensions for further decision. The processes involved in detection are described in this subsection.

3.6. Distance calculation

The principle parameter for the discrimination of DDoS attacks and flash crowds is the flow similarity among the suspicious flows. The flow similarity between any 2 suspicious sessions is computed using the Hellinger distance metric. The Hellinger distance is a metric for measuring the distance (deviation) between probability distributions. The distance calculation using the Hellinger distance metric is as follows: the number of incoming requests from each and every suspicious session is sampled for every Δt time within the sampling period of time, say T. Let $x_i[1], x_i[2], ..., x_i[N]$ be the sequence of samples from a suspicious session flow where i $(i \geq 1)$ is the index of suspicious session, and N = T / Δt (denotes the number of samples per session).

Let $X_i$and X_j be the data sequence of any 2 suspicious sessions such that i $\neq$ j with the same length N. Let the probability distribution of the 2 flows, $X_i$and X_j, be p(x) and q(x), respectively. Then we compute the Hellinger distance as given in Eq. (5) (http://www.encyclopediaofmath.org/index.php/Hellinger_distance):

$$D_H(p,q) = [\sum_x (\sqrt{p(x)} - \sqrt{q(x)})^2]^{1/2}/\sqrt{2}. \tag{5}$$

In general, the number of suspicious sessions is more than 2 during the flooding. In such a case, a number of different pairwise comparisons are made among the suspicious sessions, and the final decision is made on the overall results. Suppose that if there exists M suspicious sessions, X_1, X_2, ..., X_M, then there are ${}_MC_2$ possible combinations of the suspicious flows, $X_i$and X_j, where 1 $\leq i,j \leq M$, and $i \neq j$. Finding the distance between all possible combinations as discussed in [17] is not a good solution, especially during the flash crowd scenario where M is too large. In general, it is cumbersome and may increase the computational overhead at the victim end.

Instead, identified suspicious sessions are further classified into 3 groups based on the other 2 factors: page requested and client legitimacy. There exist 4 possible combinations: 1) the suspicious flow from a known client cluster with a request to a hot page, 2) the suspicious flow from a known client cluster with a request to a cold page, 3) the suspicious flow from a new/unknown client cluster with a request to a hot page, and 4) the suspicious flow from a new/unknown client cluster with a request to a cold page. Out of the 4, the first combination is named as the less suspicious group, the second and third combinations are grouped into the more suspicious category, and the last combination is the most suspicious group.

All possible pairwise combinations are taken into account in the case of the most suspicious flow. In the case of the less and more suspicious groups, limited combinations are sufficient for making a decision. In this paper, 75% and 50% of the possible combinations are taken at random for the more suspicious and less suspicious groups, respectively, and the distance metric is applied to them. If required, the combinations are increased in steps of 5% until a valuable conclusion is yielded. The distance value D_H(p, q) can range from 0 to 1. A value close to 0 indicates that the flows are similar and it is an indication of the possibility of DDoS attacks, and a value close to 1 indicates the possibility of a flash crowd.

Let t_d be the threshold for discrimination; then the flows i and j are marked as either 1 (no similarity between them) or 0 (remarkable similarity), as given in Eq. (6):

$$\begin{gathered} D_{ij} = 1, if D_H(p,q) \geq t_d \\ 0, otherwise, \end{gathered} \tag{6}$$

where 1 $\leq i,j \leq M$, and $i \neq j$.

The threshold t_d is set to the value of 0.4 based on the statistics related to the flash crowd and DDoS attack as discussed in Section 5. Likewise, other possible pairwise distances within each of the 3 groups are computed and the final decision is made by aggregating all the distance values within the group. For example, considering the most suspicious group consisting of S suspected sessions, each session flow is compared with every other flow, i.e. with the rest (S − 1) of the flows, and the individual results are recorded. If more than 60% of the comparison results in 0 (remarkable flow similarity), then it is identified as an attack flow and dropped immediately. Likewise, each and every flow is examined and discriminated.

3.7. Resilient proxies

In our work, the detection mechanism is implemented on the proxy system that is deployed just before the victim server in order to prevent direct flooding of the server. In such cases, all the traffic converges at the proxy and the proxy becomes a critical bottleneck point. It is not practically possible to handle flash crowds together with the attack by a single proxy. To tackle such a problem, we propose resilient proxies (more than one proxy with the same detection setup) running together at the victim end to share the load of the incoming traffic. The presence of resilient proxies helps in better load balancing, faster detection, and less request drops due to the bottleneck of the queue. The resilient schedulers forward the requests to the server in a round-robin fashion.

The number of proxies to be used depends upon the expected network traffic. Too small a number may lead to request drops due to the overwhelming number of incoming requests, and higher numbers of proxies may lead to processing overhead and poor detection rate. The processing overhead is due to the updating of threshold-related information among proxies and scheduling of requests among them. The poor detection is due to the distribution of attack traffic to a number of proxies, leading to low flow similarity.

4. Performance evaluation

To examine the performance of the proposed discrimination algorithm and to analyze the effectiveness of the proposed metric in detecting DDoS attacks, we use real datasets in our experiment. For representing a flash crowd, the 1998 FIFA World Cup dataset (http://ita.ee.lbl.gov/html/contrib/WorldCup.html) is used. Datasets reflecting application layer DDoS attacks originating from a specific botnet are very rare and it is even harder to find suitable datasets for the proposed algorithm. Thus, in order to evaluate our scheme, we analyzed and identified the essential DDoS attack parameters, with attack durations of 5 to 10 min and attack flows of a minimum of 10,000 packets per second, from the CAIDA "DDoS Attack 2007" dataset (http://www.caida.org/data/passive/ddos-20070804_dataset.xml) and [26], and we used the same for our DDoS attack simulation.

The proposed work is simulated using the NS2 simulator. Each of the legitimate client nodes replays one user's trace collected from day 90 of the 1998 FIFA World Cup (http://ita.ee.lbl.gov/html/contrib/WorldCup.html). The attack nodes are randomly selected and the ratio of attack nodes to total nodes is 20%. To launch an attack, each attack node starts sending requests based on the identified DDoS attack parameters.

Our first task is to show that the flows among the flash crowds are random and do not follow any pattern or have any similarity among them. The threshold is fixed for differentiating the genuine flash crowds from the DDoS attacks. The flash crowd during the 90th day of the FIFA World Cup dataset is examined. In a 2-h interval (2200–2359 hours of day 90), there exist 2289 users with 133,670 requests. The user request details are shown in the Table. The users who sent huge numbers of requests (>150 requests) are taken for computing the flow similarity. The reason is that there is a wide chance that they may be treated as suspicious flows during a flash crowd attack. Extensive analysis is made of the dataset and the result related with the flow similarity is presented below. Almost all the flows related to the flash crowd have no remarkable similarity. As an example, a total of 7 such users are randomly chosen and their flow lasting 1 h is taken for computation. The number of requests in each suspicious flow for every minute is recorded and a total of 60 samples are taken. The Hellinger distances between all possible pairwise flows are computed. In the same way, 2 additional sets at different times are evaluated and the results are shown in Figure 2. The set in the figure refers to all the possible pairwise combinations of the flows that are considered for evaluation. For example, in Figure 2, 7 different user flows are taken for evaluation and the distance measures for all ${}_7C_2$ combinations (21 combinations) are projected as set 1. Likewise, other sets are evaluated and projected in the respective figures. In Figure 2, values 1 to 6 on the x-axis represents the combination of flow 1 with the rest of the 6 flows (i.e. flow 2 through flow 7), respectively. Values 7 to 11 represents the combination of flow 2 with the rest of the 5 flows (flow 3 through flow 7) respectively, and so on. The result shows that the flash crowds are dissimilar in their flow patterns and their distance value is above 0.4 for most of the cases.

Table. Number of requests and users involved during 2 h of FIFA World Cup in 1998.

Number of requests	Number of users
< 10	840
10–50	581
50–100	493
100–150	183
> 150	192

The second task is to analyze the flow similarity of the DDoS attack, so that the appropriate threshold is identified and fixed for discrimination. We have simulated the DDoS attack flow based on the statistics and identified the 3 sets of suspicious flow among them. Each flow is compared with every other flow in each set and the results associated with the 3 sets are shown in Figure 3. It is clear that there exists flow similarity among the attack flows and almost all values fall below 0.35 on the Hellinger scale of 0 to 1. As a consequence, the threshold for discriminating the flash crowd from DDoS attacks is fixed at 0.4.

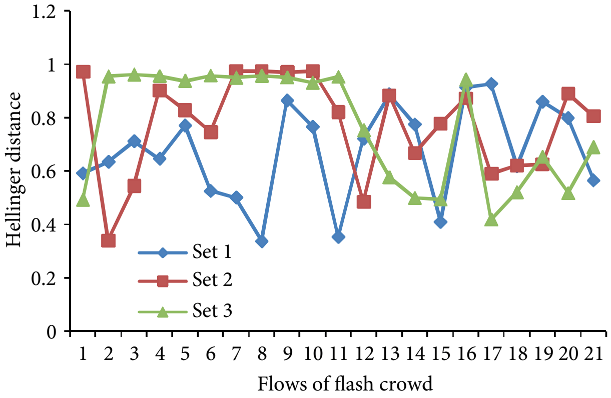

Figure 2. Flow similarity among the flash crowd.

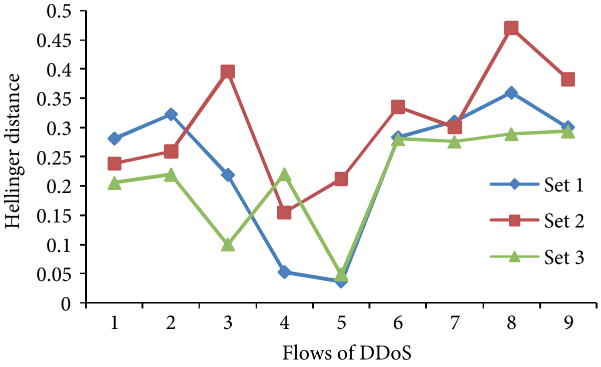

Figure 3. Flow similarity among the simulated DDoS flows.

Based on the above results, the threshold for discrimination is fixed at 0.4. Now, considering our mechanism, a few samples of the most suspicious group are taken and the flow similarity among the suspicious flows is shown in Figure 4. A total of 7 sessions having the most suspicious flows are compared with one another and the distances between each of the 7 sessions with all possible combinations are shown below. From Figure 4, it is clear that flows 1, 4, and 5 are completely above the discrimination threshold and they are identified as flash crowds. The rest of the flows, 2, 3, 6, and 7, have dispersed values. However, flows 2 and 6 have more than 60% of flow similarity and are hence identified as attack flows. The rest of the flows (i.e. 3 and 7) are identified as flash crowds.

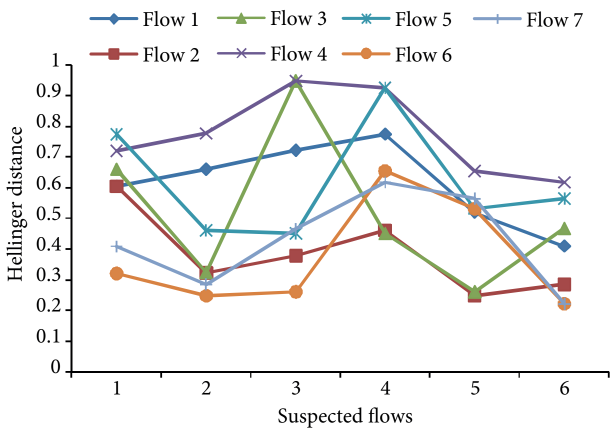

Figure 4. Flow similarity among the most suspicious flows.

5. Comparison with the existing techniques

In this paper, our technique is compared with the existing 5 techniques and the results are presented to exhibit the efficiency of the proposed work. In our comparison, the first technique makes use of the flow correlation coefficient-based [17] similarity measure, while the second and the third techniques make use of the Sibson distance measure and Jeffrey distance measure [18] to discriminate the DDoS attack from the flash crowd. Flow similarity is the only parameter considered for discrimination in the above mentioned existing techniques. The results of the similarity measure obtained from both the existing techniques and from the Hellinger distance of the proposed work are almost the same, but the discrimination is further fine-tuned by the other 2 parameters (pages referred and client legitimacy) in our technique. In the fourth technique, Li et al. [1] proposed a probabilistic metric to differentiate the DDoS attack and flash crowd. The detection is made based on the results of two metrics: the total variation metric and the Bhattacharyya metric, which are to measure the difference and the similarity of two discrete probability distributions, respectively. Finally, Xu et al. [27] proposed a detection mechanism in which the Pearson correlation coefficient is applied to the user activity degree index to detect the application layer DDoS attacks in flash crowds.

The user traces from the CAIDA 2007 dataset and 1998 FIFA World Cup are taken for comparison to reflect on the DDoS attack and the flash crowd, respectively. The attack intensity is varied from 10% to 90%. The detection accuracy of both the existing and proposed techniques is exhibited in the Figure 5. The results show that the proposed technique outperforms with a detection rate of around 91%.

The results of false positives and false negatives from both of the techniques are presented in Figures 6 and 7, respectively. The false positives are the legitimate sessions that are mistakenly identified as malicious sessions, whereas the false negatives are the malicious sessions that get access to the system without being detected. This also shows that the proposed technique detects considerably fewer false positives and false negatives when compared to the existing works.

6. Conclusion

The DDoS attack is one of the most destructive modes of attack in the last decade. Flash crowds share certain significant characteristics with DDoS attacks, which in turn aggravates the problem of detecting DDoS attacks. The behavioral differences between them are thoroughly analyzed and 3 key parameters, flow similarity, pages referred, and client legitimacy, are identified and used as the principal components for detecting application

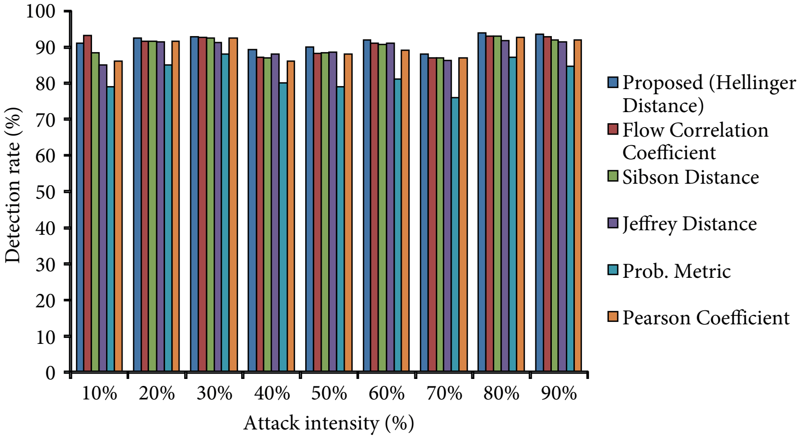

Figure 5. Detection rate.

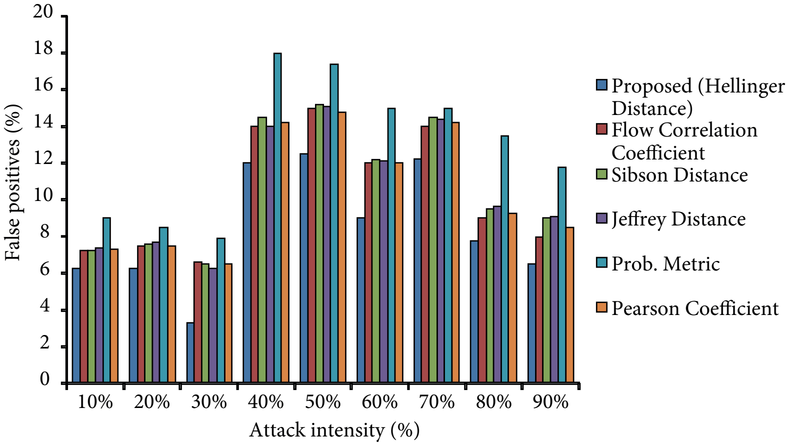

Figure 6. False positives.

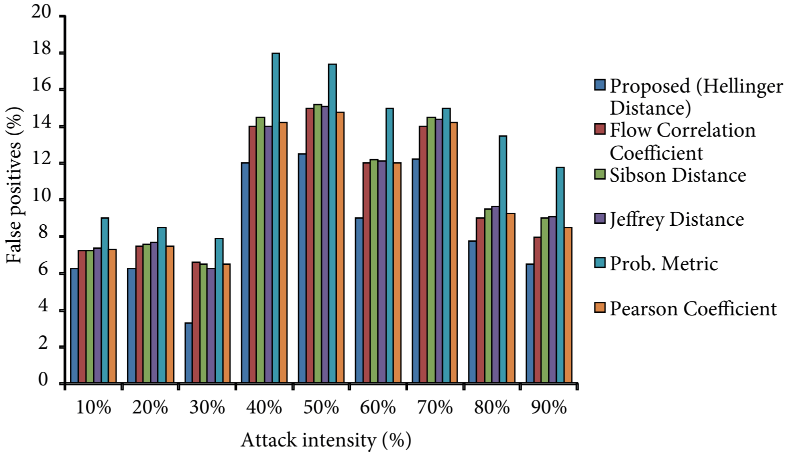

Figure 7. False negatives.

layer DDoS attacks during a flash event. We evaluate the performance of our technique through simulations. Two real publicly available datasets, the CAIDA "DDoS Attack 2007" and "1998 FIFA World Cup", are used as representatives of DDoS attacks and flash crowds to validate our approach. Our proposed technique is compared with other existing techniques and the results show that the proposed technique outperforms with a detection rate of around 91% for varying attack intensities with considerably low false positives and false negatives.

7. Future work

The proposed work can further be extended in the following ways. First of all, we are keen to evaluate the proposed technique in real network situations. This could help us to confirm the performance of the predictability test and to find more recognizable characteristics of the attack flow to achieve better detection. Second, we would like to analyze and identify the additional possible parameters and use them for further discrimination. Third, we are interested in tackling super botnets where multiple botnets together target a server at the same time. The attack patterns associated with each botnet may be different and developing a mitigating mechanism for such kinds of attacks is quite a challenging task.

References

[1] Li K, Zhou W, Li P, Hai J, Liu J. Distinguishing DDoS attacks from flash crowds using probability metrics. In: International Conference on Network and System Security; 19–21 October 2009; Gold Coast, Australia. New York, NY, USA: IEEE. pp. 9–17.

[2] Zhang S, Dasgupta P. Denying denial-of-service attacks: a router based solution. In: International Conference on Internet Computing; June 2003. pp. 301–307.

[3] El-Moussa FA, Linge N, Hope M. Active router approach to defeating denial-of-service attacks in networks. IET Commun 2007; 1: 55–63.

[4] Wang H, Jin C, Shin KG. Defense against spoofed IP traffic using hop-count filtering. IEEE T Networking 2007; 15: 40–53.

[5] Keromytis AD, Misra V, Rubenstein D. SOS: An architecture for mitigating DDoS attacks. IEEE J Sel Area Comm 2004; 22: 176–188.

[6] Li K, Zhou W, Yu S. Effective metric for detecting distributed denial-of-service attacks based on information divergence. IET Commun 2009; 3: 1851–1860.

[7] Limwiwatkul L, Rungsawangr A. Distributed denial of service detection using TCP/IP header and traffic measurement analysis. In: International Symposium on Communications and Information Technology; 26–29 October 2004; Japan. New York, NY, USA: IEEE. pp. 605–610.

[8] Cabrera JBD, Lewis L, Qin X, Lee W, Prasanth RK, Ravichandran B, Mehra RK. Proactive detection of distributed denial of service attacks using MIB traffic variables a feasibility study. In: IEEE/IFIP International Symposium on Integrated Network Management; 14–18 May 2001; Seattle, WA, USA. New York, NY, USA: IEEE. pp. 609–622.

[9] Wang J, Yang X, Long K. A new relative entropy based app-DDoS detection method. In: IEEE Symposium On Computers and Communications; 22–25 June 2010; Riccione, Italy. New York, NY, USA: IEEE. pp. 966–968.

[10] Liu H, Chang K. Defending systems against tilt DDoS attacks. In: International Conference on Telecommunication Systems, Services, and Applications; 20–21 October 2011; Bali, Indonesia. New York, NY, USA: IEEE. pp. 22–27.

[11] Yu J, Fang C, Lu L, Li Z. Mitigating application layer DDoS attacks via effective trust management. IET Commun 2010; 4: 1952–1962.

[12] Srivatsa M, Iyengar A, Yin J, Liu L. Mitigating application-level denial of service attacks on web servers: a client-transparent approach. ACM T Web 2008; 2: 15.

[13] Jung J, Krishnamurthy B, Rabinovich M. Flash crowds and denial-of-service attacks: characterization and implications for CDNs and web sites. In: International World Wide Web Conferences; 2002. pp. 293–304.

[14] Kandula S, Katabi D, Jacob M, Berger A. Botz-4-sale: Surviving organized DDoS attacks that mimic flash crowds. In: 2nd Conference on Symposium on Network and Distributed System Security; May 2005; Boston, MA, USA. pp. 287–300.

[15] Oikonomou G, Mirkovic J. Modeling human behavior for defense against flash-crowd attacks. In: IEEE Conference on Computer Communication; 14–18 June 2009; Dresden, Germany. New York, NY, USA: IEEE. pp. 1–6.

[16] Thapngam T, Yu S, Zhou W, Beliakov G. Discriminating DDoS attack traffic from flash crowd through packet arrival patterns. In: International Workshop on Security in Computers, Networking and Communications; 10–15 April 2011; Shanghai, China. New York, NY, USA: IEEE. pp. 952–957.

[17] Yu S, Zhou W, Jia W, Guo S, Xiang Y, Tang T. Discriminating DDoS attacks from flash crowds using flow correlation coefficient. IEEE T Parall Distr 2012; 23: 1073–1080.

[18] Yu S, Thapngam T, Liu J, Wei S, Zhou W. Discriminating DDoS flows from flash crowds using information distance. In: International Conference on Network and System Security; 2009. New York, NY, USA: IEEE. pp. 351–356.

[19] Xie Y, Yu S. Monitoring the application-layer DDoS attacks for popular websites. IEEE T Networking 2009; 17: 15–25.

[20] Stone-Gross B, Cova M, Cavallaro L, Gilbert B, Szydlowski M, Kemmerer R, Kruegel C, Vigna G. Your botnet is my botnet: analysis of a botnet takeover. In: Proceedings of the 16th ACM Conference on Computer and Communications Security; 2009. New York, NY, USA: ACM. pp. 635–647.

[21] Thing VLL, Sloman M, Dulay N. A survey of bots used for distributed denial of service attacks. In: International Information Security Conference; 2007. pp. 229–240.

[22] Yu S, Guo S, Stojmenovic I. Can we beat legitimate cyber behavior mimicking attacks from botnets? In: IEEE International Conference on Computer Communications; 2012. New York, NY, USA. pp. 2851–2855.

[23] Garrido A. About some properties of the Kullback-Leibler divergence. Advanced Modeling and Optimization 2009; 11: 4.

[24] Kantardzic M. Data Mining: Concepts, Models, Methods, and Algorithm. New York, NY, USA: IEEE Press, 2002.

[25] Comer DE. Computer Networks and Internets. Upper Saddle River, NJ, USA: Prentice Hall Press, 2008.

[26] Moore D, Shannon C, Brown DJ. Inferring internet denial-of-service activity. ACM T Comput Syst 2006; 24: 115–139.

[27] Xu C, Du C, Kong X. An application layer DDoS real-time detection method in flash crowd. In: IACSIT Hong Kong Conferences; 2012. pp. 68–73.