Dynamic Detection Strategy Based Flow Analysis Scheme for Flash Crowd Attacks

Narmadha.B[*1],AyyaMuthukumar.D[*2],Kannan.M[*3]
[1]PG Scholar of Computer Science, K.S.Rangasamy College of Technology, Tiruchengode, India.
Email: narmadha.kirthi@gmail.com, Mobile No: +91 8344613613.
[2] Research Scholar & Associate professor , K S Rangasamy College of Technology, Tiruchengode, India.
Email: ukdkumar@yahoo.co.in
[3] Professor, K.S.Rangasamy College of Technology,Tiruchengode, India

Abstract--**Denial-of-Service (DoS) attack is an attempt by attacker to prevent legitimate users from using resources. Denial-of-Service denies a victim (host, router, or entire network) from providing or receiving normal services. Distributed Denial of Service (DDoS) Attacks are generated in a "many to one" dimension. In DDoS attack model Large number of compromised host are gathered to send useless service requests, packets at the same time.The server manages the user requests and sends the video data to the users. The video data requests are referred as flashcrowds. Sophisticated botmasters attempt to disable detectors by mimicking the traffic patterns of flash crowds. Graphical puzzles are used to differentiate between humans and bots on DDoS attack defending process. Known features are used to detect and filter the DDoS attacks. The current attack flows are usually more similar to each other compared to the flows of flash crowds. A discrimination algorithm uses the flow correlation coefficient as a similarity metric among suspicious flows. Flow similarity is used to defeat flash crowd attacks under current botnet size and organization. The random delay is inserted into the request flows issued by the attackers.The proposed system is designed to handle user actions and reactions decisions. The flow correlation coefficient similarity based discriminative algorithm is enhanced to manage attacker actions. Dynamic flow similarity assessment is integrated with the system. Traffic flow matching is improved to minimize the detection latency. The system reduces the computational overhead in the attack analysis process.**

Keywords—**DDoS Attacks, FlashCrowds, Threshold, Discrimination, Correlation.**

I. INTRODUCTION

Distributed Denial of Service (DDoS) attacks pose a critical threat to the Internet. A recent survey [1] of the 70 largest Internet operators in the world demonstrated that DDoS attacks have increased dramatically in recent years. Moreover, individual attacks are becoming stronger and more sophisticated. Motivated by huge financial rewards, such as renting out their botnets for attacks or collecting sensitive information for malicious purposes, hackers are encouraged to organize botnets to commit these crimes [2]. Furthermore, in order to sustain their botnets, botmasters take advantage of various antiforensic techniques to disguise their traces, such as code obfuscation, memory encryption, fresh code pushing for resurrection [4], peer-to-peer implementation technology [6], [7], or flash crowd mimicking [3], [5]. Flash crowds are unexpected, but legitimate, dramatic surges of access to a server, such as breaking news. One powerful strategy for attackers is to simulate the traffic patterns of flash crowds to fly under the radar. This is referred to as a flash crowd attack.

The similarity among the current DDoS attack flows is higher than that of a flash crowd. Therefore, we propose a flash crowd attack detection method using the flow correlation coefficient. We aim to protect potential victims from flash crowd attacks within a community network. A community or ISP network often operates with the same Internet service provider domain or the virtual network of different entities which are all cooperating with one another. The community network benefits the defence of DDoS attacks in a wider range and in a cooperative way.

This is hard to achieve in the realm of the Internet, where anarchy is the underlying principle [11]. We first established a model for DDoS attack detection in a community network where the potential victim is situated. We then theoretically proved that attack flows can be discriminated from flash crowds under current botnet sizes and organization. Our experiments confirmed our theoretical conclusions. This paper makes the following contributions:

- We found a new feature of flow similarity to defeat flash crowd attacks under current botnet size and organization. It is the first work in this field to the best of our knowledge. Within the relevant literature, flash crowd attacks continue to be a challenge. Our work sheds light on a new perspective in addressing this problem at the network layer.
- The proposed algorithm works independently of specific DDoS flooding attack genres. Therefore, it is effective against unknown forthcoming flooding attacks.
- The proposed correlation coefficient-based method is delay proof. This property is very

effective against explicit random delay insertion among attack flows.

- We verified our observations with real data sets of flash crowds and real attack tool experiments in various scenarios. We conclude that it can effectively beat flash crowd attacks.

II.DEFINITIONS AND PROBLEM SETTING

In this section, we begin by presenting a number of preliminary definitions, and then discuss the setting of the discrimination problem. For simplicity, we use the terms flow and network flow interchangeably in this paper.

Definition 1 (Network Flow). For a given router in a local network, we cluster the network packets that share the same destination address as one network flow.

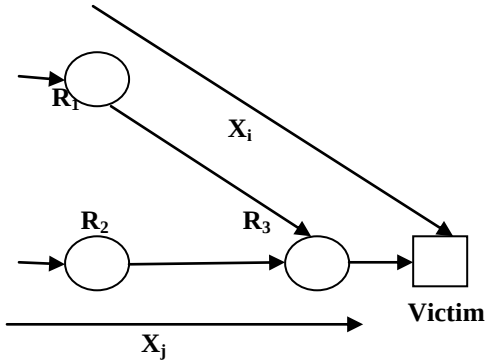

Fig. 1. A sample community network with network flows.

A sample community network with flows can be found in Fig. 1. In the sample community network, R_2 and R_3 are the edge routers, and the server is the potential victim that we try to protect. There are two incoming flows, X_i and X_j observed at R_3 and R_2, respectively. They merge at router R_1 and both are addressed to the potential victim, and enter the community network via different paths. We sample the number of packets for a given network flow with a given time interval. Therefore, a network flow can be represented by a data sequence Xi[n], where i($i \geq 1$) is the index of network flows, and n denotes the nth element in a data sequence. For example, if the length of a given network flow Xi is N, then the network flow can be expressed as follows:

$$\mathbf{X_i = \{x_i[1], x_i[2], \ldots, x_i[N]\},} \quad \mathbf{(1)}$$

Where $x_i[k](1 \leq k \leq N)$ represents the number of packets that we counted in the kth time interval for the network flow. According to the definition of flow, a router may have many network flows at any given point in time.

Definition 2 (Flow Strength). For a network flow X_i, let the length of the network flow be N ($N \geq 1$). We define the expectation of the flow as the flow strength of X_i.

$$\mathbf{E[x_i]} = \frac{1}{N}\sum_{n=1}^{N} \mathrm{xi}[n] \quad \mathbf{(2)}$$

Flow strength represents the average packet rate of a network flow. If X_i is a DDoS attack flow, then we also call $E[x_i]$ attack strength.

Definition 3 (Flow Fingerprint). For a given network flow X_i with length N, its fingerprint X'_i is the unified representation of X_i, namely,

$$\mathbf{X'_i} = \{\mathbf{x'i[1]}, \mathbf{x'i[2]}, \ldots, \mathbf{x'i[N]}\} = \left\{\frac{x_i[1]}{N.E[X_i]}, \frac{x_i[2]}{N.E[X_i]}, \ldots, \frac{x_i[N]}{N.E[X_i]}\right\} \quad \mathbf{(3)}$$

Following this definition, we know PN $\sum_{k=1}^{N} \mathbf{x'i[K]} = 1$. Based on Definitions 2 and 3, we obtain the following relationship between a network flow and its fingerprint

$$\mathbf{x_i = N - E[X_i] - X'_i} \quad \mathbf{(4)}$$

As previously discussed, the current botnets, such as SDbot, Rbot and Spybot, employ the same program to generate attack packets. Furthermore, in order to achieve the purpose of denial of service, each bot has to generate as many attack packets as they can, usually with a very short delay between two attack packets. This indicates that flow fingerprint does exist in attack flows for a given botnet:

Let X_i and X_j ($i \neq j$) be two network flows with the same length N, then the correlation between the two flows is defined as

$$\mathbf{r\ x_i, x_j} = \frac{1}{N}\sum_{n=1}^{N} xi[n]xj[n] \quad \mathbf{(5)}$$

The correlation is used to describe the similarity of different flows. However, in some cases, it may indicate zero correlation although the two flows are completely correlated but with a phase difference. However, there might still be a magnitude difference for the same similarity in different scenarios, therefore, unification is necessary.

Definition 4 (Flow Correlation Coefficient). Let X_i and X_j ($i \neq j$) be two network flows with the same length N.The flow correlation coefficient is used to indicate similarity between two flows. It is sometimes the case that two similar flows may have a phase difference which will decrease the correlation coefficient. Fortunately, this is easy to deal with because we can shift one flow to match the other, and take the maximum value of the correlation coefficients to represent the similarity of two flows.

III. SIMILARITY-BASED DETECTION METHOD

In this section, we present the similarity-based detection method against flash crowd attacks. For a given community network, we set up an overlay network on the routers that we have control over. We execute software on every router to count the number of packets for every flow and record this information for a short term at every router.

Under this framework, the requirement of storage space is very limited and an online decision can be achieved.

A real community network may be much more complex with more routers and servers. However, for a given server, we can always treat the related community network as a tree, which is rooted at the server. We must point out that the topology of the community network has no impact on our detection strategy, whether it is a graph or a tree, because our detection method is based on flows rather than network topology.

Once an access surge on the server occurs, our task is to identify whether it is a genuine flash crowd or a DDoS attack. According to our proposal, when a possible DDoS attack alarm goes off, the routers in the community network start to sample the suspected flows by counting the number of packets for a given time interval, for example, 100 milliseconds. When the length of a flow, N, is suitable, we start to calculate the flow correlation coefficient between suspected flows.

Suppose we have sampled M network flows, X_1, X_2,. . . ,X_M, therefore, we can obtain the flow correlation coefficient of any two network flows, X_i ($1 \le i \le M$) and X_j ($1 \le j \le M$, $i \ne j$). Let I X_i, X_j be an indicator for the similarity of flow X_i and X_j, and I X_i, X_j has only two possible values: 1 for DDoS attacks and 0 otherwise.

In general, we may have more than two suspected flows in a community network. This means we can conduct a number of different pairwise comparisons, and the final decision can be derived from them in order to improve the reliability of our decision. We can, therefore, have an integrated DDoS attack positive probability .

IV. DDOS ATTACK DETECTION FLOW CORRELATION COEFFICIENT

In this section, we first prove that flash crowds and DDoS attacks can be differentiated using the flow correlation coefficient in theory. Following this foundation, we analyze the effectiveness of the proposed discrimination method, and prove that the threshold does exist.

In order to make our analysis clear, we make the following assumptions:

1. There is only one server in a community network which is under attack or experiencing a flash crowd at any given time.
2. The attack packets enter the community network via a minimum of two different edge routers.
3. In one attack session, all the attack packets are generated by only one botnet, therefore the fingerprints of the attack flows are the same.
4. The network delays are discrete and countable.

Based on our knowledge of current botnets, the above assumptions are applicable in practice. However, attackers may disable our detection method by circumventing some conditions once our strategy is known to them.

Theorem 1. Let X_i and X_j ($i \ne j$) be two traffic flows that share the same distribution, and the standard deviation σ is a random variable, the correlation coefficient of the two flows is inversely proportional to σ, namely, $\rho\ X_i, X_j \alpha \frac{1}{\sigma}$

We now investigate the flow correlation coefficient of any two independent network flows, such as flash crowd flows. Previous research has demonstrated that web traffic follows the Pareto law, hence, the Pareto distribution represents the flow fingerprint of flash crowds.

Theorem 2. Given two same length instances, X_i and X_j ($i \ne j$), of a flash crowd that are generated by the same function and same parameters,

$$\lim_{N\to\infty} \rho x_i, x_j[k] = 0$$

Corollary 1. For two independent flash crowds X_i and X_j with the same length N, $\forall\ \delta\ (\delta < 1)\ \exists N'$, when $N > N'$, we have $\rho\ x_i, x_j[k] < \delta$.

We now move to explore the flow correlation coefficient among DDoS attack flows. Let us first find the expression of a DDoS attack flow, X_i, which we obtained at an edge router. Suppose the observed attack flow is a mixture of attack flows that came from K different bots, and let X'_0 be the fingerprint of the attack flows. Based on the aforementioned discussion, the fingerprint of different attack flows in one attack session is the same, except that there are delays in different attack flows.

Theorem 3. Let X'_0 be the fingerprint of attack flows for one attack session. Under the condition of no network delay and no background noise, for two mixed attack flows X_i and X_j ($i \ne j$) that we observed at two edge routers, the correlation coefficient of X_i and X_j is 1, namely, $\rho\ x_i, x_j[k] = 1$.

In reality, however, delay and noise do exist and bots in a centralized botnet are coordinated by their botmaster. This means the delays among the attack flows from different bots depend on normal Internet delays, and therefore are limited compared to fast Internet transportation facilities. As a result, the delay free condition can be satisfied to some degree. On the other hand, noise in attack flows is the legitimate packets that are also addressed to the victim at the same time when a DDoS attack is ongoing. However, the strength of noise is much smaller compared with that of DDoS attack flows. Following Theorem 3, we further have the following corollary.

Corollary 2. Let Y_i and Y_j be the noises for two DDoS attack flows X_i and X_j of one attack session, $\forall\ \delta\ (\delta < 1)$,$\exists\Delta$, $\rho\ x_i, x_j[k] \ge 1$, holds when $\frac{E[x_i]}{E[y_i]} > \Delta$ and $\frac{E[x_j]}{E[y_j]} > \Delta$

Corollary 2 indicates that the correlation coefficient of DDoS attack flows approaches 1 if the

Signal-Noise-Ratio (SNR), $\frac{E[x_i]}{E[y_i]}$, is sufficiently large. It is true that $E[X_i] >> E[Y_i]$ and $E[X_j] >> E[Y_j]$ for the DDoS attack cases, therefore, the correlation coefficient of attack flows is close to 1 in an ongoing DDoS attack scenario.

Theorem 4. DDoS attack flow can be discriminated from flash crowds by the flow correlation coefficient at edge routers under two conditions: the length of the sampled flow is sufficiently large, and the DDoS attack strength is sufficiently strong.

The proof of Theorem 4 can be found in the online supporting material. It is necessary that we obtain an upper bound,δ, of the flow correlation coefficient for flash crowds for a given flow length. In the case that the flow correlation coefficient is greater than δ, we assume them to be DDoS attack flows.

V. ANTI DETECTIONS ISSUES

As we know, detection and antidetection is an endless battle between defenders and attackers. Our discrimination method is effective under the current conditions of botnet size and organization. Hackers may make efforts to circumvent our similarity-based detection. We discuss them here for readers to carry on further research in this field.

First of all, if attackers are able to organize a super botnet, in which the number of live bots is the same or close to the number of concurrent users of a flash crowd, then, one bot can mimic the legitimate behavior of one user. We have to note that it is still an open problem for both attackers and defenders: Can botnet owners organize this kind of super botnet or not? There are many factors that limit the number of live bots of a botnet, such as time zone, antivirus software, operating system patching. Second, in order to disguise their flow fingerprints, bot writers may include many attack packet generation functions in their binary, and make each bot randomly choose one function to generate the attack packets. Flow similarity drops among different distribution flows compared with that of the same distribution flows. However, this impact is limited compared with that from the number of live bots. Moreover, we believe there must be some differences between a mimicking attack and a genuine flash crowd. What we need to do is to discover them and deploy them to defeat mimicking attacks.

VI. REACTIVE MODEL BASED FLOW CORRELATION ANALYSIS

The server is requested by a group of users. All the user requests are received by the server and their requests are processed. The response data values are redirected to the users. The user request flow is monitored by the server. Request count and request interval values are observed by the server. The system maintains a threshold request interval for the user requests. The request count level is verified and flow analysis is initiated with reference to the request levels. The request flow similarity is measured with correlation coefficient values. The suspected data is verified with other flow sequences. The attack requests are identified by the variations detected from the flow correlation coefficient analysis. The server takes an action against the users. The user requests are discarded by the server. The user identifies the action taken to their requests. In this case the user changes their request flow sequences.

The flow correlation coefficient similarity based discriminative algorithm is enhanced to manage attacker actions. The users change the attack activity model with different flow sequences. In this case the request flow similarity analysis is tuned with recent request flow data. The request flow changes for the same user must be monitored to identify the user attacks. Dynamic flow similarity assessment is integrated with the system. The user request flow verification can divided into a set of partitions. The request flow verification can be performed in two ways. They are verification request flows for the user level and the verification of request flow between the users. The user actions can be detected using the user level flow changes. The user level flow variation can be observed in two different threshold models. They are dynamic request count and period thresholds. The request flow is partitioned with reference to the server action against the users.

The flow correlation analysis is initiated in different request count threshold levels. The threshold dynamically estimated with reference to the network request types. In the case of attack count is increased the threshold is reduced. In normal flows the threshold is increased for correlation analysis. Traffic flow matching is improved to minimize the detection latency. The system maintains the recent flows for the analysis. Existing flow information are aggregated and used for the analysis. Computational cost is reduced by the system.

VII. CONCLUSION

Distributed Denial of Service (DDoS) attacks are initiated by the botnets. Traffic flow analysis scheme is used to detect and filter DDos attacks on Flash Crowds. Flow Correlation Coefficient based discriminative algorithm is enhanced with reactive mechanism. Cost control mechanism is added with the system. The system improves the response time for the video stream transmission. Dynamically changing attack behaviors are identified by the system. Historical data and live attack streams are used to test the system efficiency. False error rates are reduced.

.

REFERENCES

[1] Arbor, "IP Flow-Based Technology," http://www.arbornetworks.com, 2011.

[2] B. Stone-Gross, M. Cova, L. Cavallaro, B. Gilbert, M. Szydlowski, R. Kemmerer, C. Kruegel, and G. Vigna, "Your Botnet Is My Botnet: Analysis of a Botnet

Takeover," Proc. ACM Conf. Computer Comm. Security, 2009.
[3] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of Network-Based Defense Mechanisms Countering the DoS and DDoS Problems," ACM Computing Survey, vol. 39, no. 1, pp. 123-128, 2007.
[4] C.Y. Cho, J. Caballero, C. Grier, V. Paxson, and D. Song, "Insights from the Inside: A View of Botnet Management from Infiltration," Proc. Third USENIX Conf. Large-Scale Exploits and Emergent Threats: Botnets, Spyware, Worms, and More (USENIX LEET), 2010.
[5] A. Scherrer, N. Larrieu, P. Owezarski, P. Borgnat, and P. Abry, "Non-Gaussian and Long Memory Statistical Characterizations for Internet Traffic with Anomalies," IEEE Trans. Dependable Secure Computing, vol. 4, no. 1, pp. 56-70, Jan.-Mar. 2007.
[6] T. Holz, M. Steiner, F. Dahl, E. Biersack, and F.C. Freiling, "Measurements and Mitigation of Peer-to-Peer-Based Botnets: A Case Study on Storm Worm," Proc. First Usenix Workshop Large-Scale Exploits and Emergent Threats (LEET), 2008.
[7] M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and M. Karir, "A Survey of Botnet Technology and Defenses," Proc. Cybersecurity Applications and Technology Conf. for Homeland Security, 2009.
[8] Shui Yu, Wanlei Zhou, Weijia Jia, Song Guo, Yong Xiang and Feilong Tang, "Discriminating DDoS Attacks from Flash Crowds Using Flow Correlation Coefficient" IEEE Transactions on Parallel and Distributed Systems, Vol. 23, No. 6, June 2012.