Information Theory Based Detection Against Network Behavior Mimicking DDoS Attacks

Shui Yu, *Member, IEEE,* Wanlei Zhou, *Member, IEEE,* and Robin Doss, *Member, IEEE*

***Abstract*— DDoS is a spy-on-spy game between attackers and detectors. Attackers are mimicking network traffic patterns to disable the detection algorithms which are based on these features. It is an open problem of discriminating the mimicking DDoS attacks from massive legitimate network accessing. We observed that the zombies use controlled function(s) to pump attack packages to the victim, therefore, the attack flows to the victim are always share some properties, e.g. packages distribution behaviors, which are not possessed by legitimate flows in a short time period. Based on this observation, once there appear suspicious flows to a server, we start to calculate the distance of the package distribution behavior among the suspicious flows. If the distance is less than a given threshold, then it is a DDoS attack, otherwise, it is a legitimate accessing. Our analysis and the preliminary experiments indicate that the proposed method can discriminate mimicking flooding attacks from legitimate accessing efficiently and effectively.**

***Index Terms*— DDoS detection, distribution distance.**

I. Introduction

THE distributed denial of service (DDoS) is a critical threat to the Internet, flooding attack is the most popular method taken by hackers. To guard against flooding attacks, researchers have designed and implemented a number of countermeasures based on traffic features [1], [2], [3]. [3] tried to use three dimensions, traffic patterns, client characteristics and file reference characteristics, to discriminate the flash event (surge massive legitimate assessing) from DoS attacks. However, this counter attack method can not follow the ever changing attack methods, as the attack patterns are changing from time to time. Moreover, the attacker will mimic the network traffic pattern of flash event, which will disable the detector quickly. The entropy detector mentioned in the paper [4] used entropy very shallowly. It can raise the alarm for a sudden massive accessing, however, it can not discriminate the DDoS attacks from the surge of legitimate accessing. [2] suggested to separate flash crowd from DDoS flooding by "other" performance metrics (except the change-point detection method) to capture their difference. However, there is no clue about the possible metrics in the paper at all. Based on our knowledge, there are no effective methods to discriminate DDoS flooding attacks from the legitimate surging accessing so far.

In this letter, we are motivated by identifying the attacks which simulate normal network accessing patterns to fly under the radar. In order to achieve the goal, we take use of information theory to explore the distance (relative entropy in information theory) among network flows. We apply our methods on community networks, e.g. ISPs, in which a number of routers can cooperate to identify DDoS at the very early stage. When the current DDoS detection algorithms, e.g. the entropy detection algorithm, raise an attack alarm, it may be a false alarm. For example, it is a surging of legitimate accessing for a breaking news or event, unfortunately, these legitimate accessing will be blocked as DDoS attacks. On the other hand, real attacks maybe reach the victim via huge number of paths with a 'normal' package rate and distribution for each path, in this case, the current detection algorithms will be fooled and can not identify the attack. In this letter, we concentrate on solving this problem. With our method, once a DDoS attack alarm is raised, the routers of the community network will calculate the distance among the suspicious flows to the possible victim on different paths, respectively. If the distances among the suspicious flows are the same or very close, we can therefore identify it as an attack. The remainder of the paper is organized as follows.

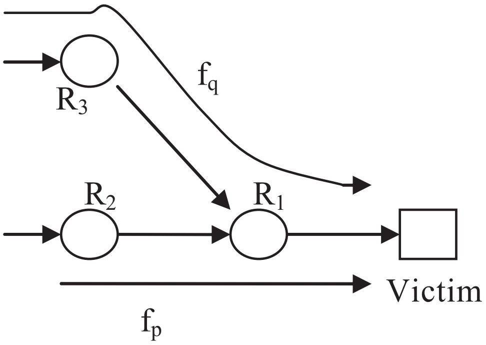

Fig. 1. A simple attack diagram in a community network.

The system analysis and modeling are done in Section 2. The distance detection algorithm is presented in Section 3. Our experiments are discussed in Section 4. Finally, in Section 5, we summarize the paper and discuss the future work.

II. System Analysis

In a community network, we can manage and configure our routers. Therefore, the routers can cooperate with each other to identify the DDoS attacks. Figure 1 displays a very simple attack diagram with two attack flows in a community network with three routers and one server as victim.

We have a few assumptions in order to make the discussion clear and easy to be understood.

Manuscript received December 5, 2007. The associate editor coordinating the review of this letter and approving it for publication was C.-K. Wu. This work is partially supported by the ARC Discovery grant (Project number DP0773264).

The authors are with the School of Engineering and Information Technology, Deakin University, Melbourne, Australia (e-mail: wanlei@deakin.edu.au).

Digital Object Identifier 10.1109/LCOMM.2008.072049.

1) The attackers use the same function to generate attack packages at the zombies, for example, the Poisson distribution or the Chi-square distribution.
2) There is only one server in the community network that is under attacking or massive accessing at a time.
3) The network system is linear and stable (this is quite reasonable for a short time period).

Similar to [5], we define the packages which share the same destination address as a flow on each of the routers in a community network. Once an alarm is raised, the routers start to sample the number of packages of the suspicious flows in a time unit, e.g. 0.05 second, respectively. Suppose in an attack scenario, the attacker uses a random variable X to control the generation speed of the attack packages. The possible methods could be:

1) Using a constant speed to generate the packages, namely,$P(X = C) = 1$, and C is a constant;
2) Increasing the number of attack packages according to attack time t, $X = at + b$, a and b are constants;
3) Simulating the network traffic as Poisson distribution, namely, $P(X = k) = \frac{\lambda^k e^{-\lambda}}{k!}, k = 0, 1...$, and λ is a constant;

For example, in the community network as Figure 1, once a DDoS attack alarm is raised, we find that there are two suspicious flows,f_p and f_q, the sampling job can be done either on router R_2 and R_3 or R_1 for a suffice time period T. Once the sampling job is done, we obtain the two distributions for f_p and f_q, respectively,

$$\left\{ \begin{array}{l} P(X) = p(x_1, x_2, x_3...x_n) \\ Q(X) = q(x_1, x_2, x_3...x_n) \end{array} \right. \tag{1}$$

We hence obtain the distance, relative entropy or Kullback-Leibler distance [2], of the two distributions as follows,

$$D(p||q) = \sum_{x \varepsilon \chi} p(x) \log \frac{p(x)}{q(x)} \tag{2}$$

where χ is the sample space of X

If f_p and f_q are attacking flows, then they are generated by the same function $f(x)$ at different zombies. Let $g_p(.)$ and $g_q(.)$ be the system functions for flow f_p and f_q from the zombie to the sampling routers, respectively. Then we have,

$$\left\{ \begin{array}{l} p(x) = g_p(f(x)) \\ q(x) = g_q(f(x)) \end{array} \right. \tag{3}$$

Ideally, if $g_p(.)$ and $g_q(.)$ are linear, then $D(p||q) = 0$. Although it is impossible that the Internet satisfies the assumption, however, the distance of two attack flows should be very small compared with the case that one is attack flow and one is normal flow. We us the following formula to make the judgment of the two flows, p and q, are the same flows (attack flows) or not.

$$A(p, q) = \left\{ \begin{array}{ll} 1 & D(p||q) \leq \delta \\ 0 & D(p||q) > \delta \end{array} \right. \tag{4}$$

Where δ is a given small number as a threshold. We now extend the number of attack flows to $n(n > 2)$. We make $m(2m < n)$ flow pairs randomly from the n suspicious flows. Notated as $p_i, q_i, i = 1, 2, ...m$, respectively. Hence, we can have $A_i(p_i, q_i), i = 1, 2...m$. Suppose the possibility of misjudgment for each pair is p, then the positive judgment for an attack could be expressed as follow.

$$A(p_1, p_2, ...p_m, q_1, q_2...q_m) = 1 - p^m \tag{5}$$

For example, if $p = 0.5$, and we expect 99% positive for the judgment, then the condition is $m \geq 7$.

TABLE I

THE DISTANCE ALGORITHM

1. Identify the suspicious flow on the local router
2. Count the number of packages of the suspicious flow in a time slot t.
3. Identify the sample space, $x_1, x_2...x_n$ of sampling period T, and the responding frequency
4. Calculate the possibility distribution of the flow as $p(x_i) = \frac{f_i}{\sum_{1=1}^{n} f_i}, i = 1, 2...n$
5. Submit the distribution to an aggregate router, e.g. R1 in Figure 1, to calculate the distance according to equation (2).
6. Compare the distances for the suspicious flows on different routers in the community network, and make the decision based on equation (4)
7. If it is an attack, then discard the packages of the suspicious flow on the routers.

A. The Distance Algorithm

Based on the analysis of the previous section, we have the corresponding algorithm. The algorithm is listed as in Table I.

Based on this algorithm, we can identify DDoS attacks as early as possible, therefore, actions will be taken to discard the attack packages.

III. THE SIMULATIONS

In order to confirm the proposed method, we conducted a number of simulations. We implemented the ARPA network in Opnet. The simulation environment includes 21 routers and 336 network nodes. We tested two different attack methods, Poisson attack and Chi-square attack, with background traffic as Poisson distribution. We compare the distance between an attack flow and a normal flow, the distance between two attack flows in different attack scenarios, respectively. The results are presented in Figure 2 and Figure 3.

From the above findings, we obtain that:

1) The system gets to stable after 50 time units after the sampling procedure starts. If we make 20 samples per second, then we only need 2.5 seconds to make the decision.
2) The distance between an attack flow and a normal flow is normally around 1.5 once the system is stable.
3) The distances between two attack flows are less than 0.2 ($\delta = 0.2$), which is pretty different from the distance between one attack flow and one normal flow. Therefore, we can conclude that our methods can discriminate the DDoS attacks from legitimate surge of accessing effectively and efficiently.

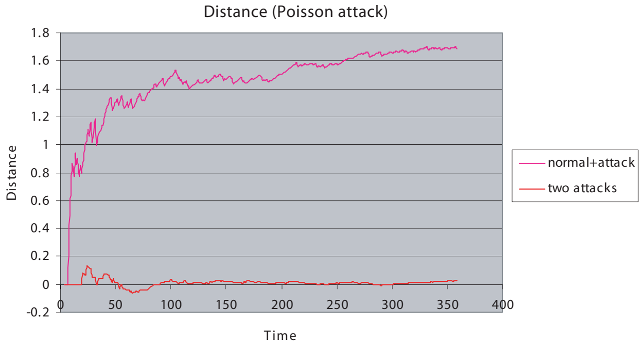

Fig. 2. Distance comparison for Poisson attacks.

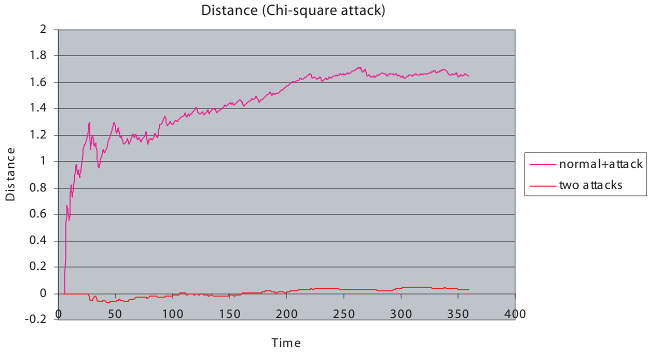

Fig. 3. Distance comparison for Chi-Square attacks.

IV. Summary and Future Work

In this letter, we are motivated by discriminating the DDoS attacks from surge legitimate accessing, and identifying attacks at the early stage, even before the attack packages reach the target server. We concentrate on the different attack flows of one attack essentially share the same attack pattern, which is not possessed by legitimate accessing flows in a short time period. Therefore, once the DDoS attack detection algorithms raise the alarm of a potential attack, we start to calculate the distance among the different suspicious flows in the community network. If the distance is less than a given threshold, for example 0.2 based on our experiments, then it is an attack, otherwise, we treat it as a surge of legitimate accessing.

The advantages of the proposed methods are

1) The detection and defense can be implemented in a community network easily. Further, it can be done independently in the community network.
2) The detection can identify the attack at the very early stage, e.g. a few seconds.
3) The proposed method does not have the pressure of storage of the past packages for analysis, it also does not cost the computing power of routers much.

We present the basic idea in this letter, and our theoretical analysis and simulations demonstrate that it is an excellent method for DDoS detection. However, there is more interesting work to do, which includes,

1) The compromise of detection accuracy and the time of confirming is a critical aspect in the battle with DDoS attacks, how to obtain an optimal solution for this is quite demanded in practice;
2) Identify DDoS attacks when there are quite a number of legitimate accessing to the same server.
3) Attackers may using multiple attack package generation functions in one attack, trying to fool our detection algorithm, e.g. they may use random functions with different seeds, however, we believe, there are definitely rules behind the attack distributions if they are 'man-made' traffic rules. This is also a direction that we are going to explore with high interest.
4) Extensive experiments on the Internet against real DDoS attacks

References

[1] R. B. G. Carl, G. Kesidis, and S. Rai, "Denial of service attack detection techniques," *IEEE Internet Computing*, Jan. 2006.
[2] Y. Chen and K. Hwang, "Collaborative change detection of DDoS attacks on community and ISP networks," in *Proc. IEEE CTS 2006.*
[3] J. J. B. Krishnamurthy and M. Rabinovich, "Flash crowds and denial of service attacks: characterization and implications for CDNs and Web sites," in *Proc. International WWW conferences 2002.*
[4] R. C. J. K. Kumar and K. Singh, "A distributed approach using entropy to detect DDoS attacks in ISP domain," in *Proc. International Conference on Signal Processing, Communications and Networking (ICSCN'07).*
[5] T. M. Cover and J. A. Thomas, *Elements of Information Theory, Second Ed.*, 2007.