A WEB BASED APPROACH TO DETECT MIMICKING ATTACKS IN HOMOGENEOUS ENVIRONMENT

R. Padmapriya and S. Igni Sabasti Prabu
Information technology, Sathyabama University, Chennai, India
E-Mail: padmapriyacse60@gmail.com

ABSTRACT

Botnets have become major engines for malicious activities in cyberspace nowadays. Botnets are the main drivers of cyber attacks, such as distributed denial of service (DDoS), flash crowds, information phishing and email spamming. Both flash crowds and DDoS attacks have very similar properties in terms of internet traffic. Flash crowds are legitimate flows whereas DDoS attacks are illegitimate flows. To sustain their botnets, botnet owners are mimicking legitimate cyber behavior. This poses a critical challenge in anomaly detection. In this work, study of mimicking attacks and detections from both sides, as attackers and defenders is done. First of all, a semi-Markov model for browsing behavior is established. Based on this model, a botmasters can simulate flash crowd successfully in terms of statistics, with a sufficient number of active bots (not less than the number of active legitimate users). But it is hard for botnet owners to satisfy the condition to carry out a mimicking attack most of the time. With this new finding, we conclude that mimicking attacks can be discriminated from genuine flash crowds using second order statistical metrics. We detect the mimicking attacks when the sufficient number condition does not hold for botmasters. Detection is proclaimed to the user. Furthermore, the findings can be widely applied to similar situations in other research fields.

Keywords: mimicking, flash crowd attack, detection, second order metrics.

1. INTRODUCTION

The exponential growth of World Wide Web (WWW) is making it the standard information system for an increasing segment of the world's population. The Internet was originally designed for openness and scalability. For example, the Internet Protocol (IP) was designed to support ease of attachment of hosts to networks, and provides little support for verifying the contents of IP packet header fields. This makes it possible to fake the source address of packets, and hence difficult to identify the source of traffic. Moreover, there is no inherent support in the IP layer to check whether a source is authorized to access a service. Packets are delivered to their destination, and the server at the destination must decide whether to accept and service these packets. A denial of service (DoS) attack aims to deny access by legitimate users to shared services or resources. This can occur in a wide variety of contexts, from operating systems to network-based services. When the traffic of a DoS attack comes from multiple sources, it is called a distributed denial of service (DDoS) attack [1].

However, nowadays Botnets are the main drivers of cyber attacks, such as distributed denial of service (DDoS), information phishing and anti attacks. Other examples of mimicking attacks, such as email spamming , membership recruitments of botnet, etc., The term *bot* (derived from the word *robot*) is used in industry terminology to describe an machine or automated process in both the real world and the computer world. A bot generally supports a communication channel with the attacker, as well as the ability to execute particular tasks, for example, launching mimicking attacks, according to the attacker's instructions [1]. Well experienced attackers usually simulate the phenomenon of flash crowds to disable intrusion detection systems (referred to as a flash crowd attack) [6], [7].

Discriminating flash crowd attacks from genuine flash crowds has been explored for approximately a decade. Previous work [8]-[10] has focused on extracting DDoS attack features, followed by detecting and filtering DDoS attack packets using the known features. However, these methods cannot actively detect DDoS attacks. The current popular defense against flash crowd attacks is the use of graphical puzzles to differentiate between humans and bots [11]. This method involves human responses and can be annoying to users. From the botnet programmer's perspective, in order to simulate the legitimate behavior of a web browser, we need three key pieces of information: web page popularity of the target website, web page requesting time interval for a user, and number of pages a user usually browses for one browsing session (referred to as browsing length).

In this paper, we attempt to demonstrate that legitimate cyber behavior can be successfully simulated; therefore, it is not possible to discriminate mimicking attacks from legitimate cyber events using statistical methods. However, in order to achieve this, attackers have to possess a sufficiently large number of active bots, than the number of active legitimate users. Based on the study of mimicking attacks, we establish a four parameter semi-Markov model to represent browsing behavior. Using this model, we successfully simulate browsing behavior of victim client. We find that the first order statistical metric does not serve our discrimination task, and the traditional second order metric (e.g. the standard deviation) is not good enough in terms of detection granularity. We therefore invent a new second order statistical metric based on the traditional correntropy to serve the detection tasks with fine detection accuracy.

ANALYSIS

In 2007, TAO PENG analysed and found that, in contrast to direct attacks, indirect attacks can exploit insecure actions that may be performed by genuine users. These attacks generally require human interaction.

Traffic model analysis

In 2009, Ke Li, Wanlei Zhou gave an analysis of traffic model. Firstly, Flash crowds and DDoS attacks are very similar in traffic behavior from macroscopic observation; however there are also several essential differences in the aspects of access intents, distributions of source IP address and speed of the increased and decreased traffic. Flash crowds are the results of the legitimate users respond to special events such as breaking news or popular products (movies, music and software) release. All the users just want to obtain the information or material quickly from the server. If the server is slowed down they will even shut down. However, DDoS attacks are not social events and all the requests are launched by attackers and are illegitimate. Secondly, the distributions of the source IP address are also quite different between Flash crowds and DDoS attacks [2].If we aggregate the IP addresses of flash crowd attack, the distribution of source IP addresses will be subject to the fractional Gaussian noise distribution [5] However, If we aggregate these source the distribution of source IP addresses of DDoS attack, it will subject to the Poisson distribution [4]. Thirdly, there is a big difference in the increased and decreased speed of traffics between them.

Web browsing dynamics

Breslau *et al.* analyzed web accessing behavior and found that page popularity follows the Zipf-like distribution [26]. A general form of the popularity distribution is called the Zipf-Mandelbrot distribution [13]. These findings are widely used in research papers, such as [17] and [18].

For a given website, if all the bots of a botnet requests page based on the Zipf-Mandelbrot law,we will not be able to identify which ones are attack requests. Therefore, attackers can easily disable statistics based detection algorithms using this strategy in their bot programs.

Crovella and Bestavros found that viewing time distribution on web pages follows the Pareto distribution [14] (confirmed also by [19] and [20]). This information is very useful for botnet writers. Once a browsing page has been decided, a bot submits the page request to the victim and downloads the page to the host computer without displaying it (e.g. discarding it or depositing it to the cache). When the requested page has been downloaded, the bot decides a “reading" time interval following the Pareto distribution before requesting another web page.

The last element for browsing dynamics is browsing length namely the number of pages a user generally views during a browsing session. Huberman *et al.* indicated that the probability follows the two-parameter inverse of guassian distribution. This information can be employed by botnet writers to decide how many pages to request for a bot, otherwise, the defender may notice that many “clients” have a long browsing length, and therefore detect the attack. This fact forces bot-masters to possess a sufficient number of active bots to carry out flash crowd attacks.

Mimcking attack detection

Shui Yu in 2012 gave a mechanism for attack detection. If the sufficient number condition holds for a botnet owner, then botmaster can perfectly simulate a cyber event such as mimicking attacks. He took World CUP 98 data set [12] and the Auckland data set [13],and based on research [3], concluded that the number of active bots of a botnet is usually only at the hundreds or a few thousands level. Therefore, in order to carry out a flash crowd attack, the sufficient number condition is hard to meet. He demonstrated the effectiveness with two examples such as the Gaussian distribution and the Poisson distribution as both of them are typical and widely used for network traffic modeling [16].

PROPOSED SYSTEM

Mimicking attacks and detections are analyzed both as botmasters and victim clients. From the botnet developers (botmaster) point of view, in order to simulate the legitimate behavior of a web browser, we need three information such as popularity of target website, time interval between two consecutive request for a web page, number of pages the browser usually browses for one browsing session.

If attackers have a sufficient number of active client nodes, then each bot can simulate one genuine user using these three statistical distributions. However, it is difficult for botnet owners to meet the adequate number condition for certain mimicking attacks.

Following are the contributions of our paper

a) We demonstrate that botmasters can simulate a flash crowd successfully in terms of statistics. With a sufficient number of active bots, a botmaster can use one bot to simulate one legitimate user using the knowledge of web browsing dynamics.
b) We establish a four parameter semi-Markov model to represent browsing behavior. Using this model, we

can successfully simulate browsing behavior, and therefore can successfully initiate a cyber event.

c) We propose a new second order statistical metric for the detection purpose. We therefore invent a new second order statistical metric based on the traditional correntropy to serve the detection tasks with fine detection accuracy.

d) Thus the attack is detected and the event is proclaimed to the user.

Browsing behavior tracking

We count the number of HTTP requests of each flow for the given time intervals and to describe the browsing behavior of a legitimate web viewer or user. This training should be taken periodically to update the parameters to reflect the ever changing web browsing behavior.To describe the browsing behavior of a legal web viewer, we extend the classical Markov model to a four parameter semi-Markov model as shown in Figure-1.

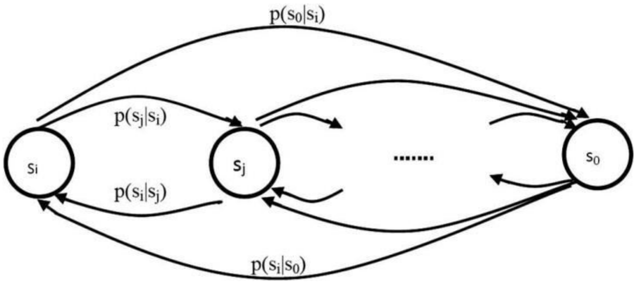


Figure-1. Markov process model four parameter semi-Markov model is as follows.

$\Delta = (S, M, l, \pi)$

where (S, M, l, π) the state transition matrix, duration at the current state, browsing length, and the initial probability distribution of the states, respectively. For a given point of time, we expect to know the number of total page requests to a web site, and number of requests for a specific web page of the web site.

We need one more parameter: the number of active web viewers for a given time point t, which we denote as m(t). m(t) varies against the time point of a day. Intuitively, there are more web viewers during working time than early morning.

Implementation of Mimicking attack:

In this part, using the collected details about the victim, the botmaster will successfully generate the mimicking attack. If we perform any modification in the botmaster side, it will automatically get reflect in the victim client page. Observe the target website for extracting all the markov model parameters and initialize these parameters. Choose sample set of bots s, from the set of active bots S and instruct these bots to run independently. Generate a random number rm. As per the client browsing behavior tracking discussed earlier in this paper, first decide the initial page .Next decide the browsing length L of the current bot based on clients browsing history. When the browsing length is within L,request a page and download the content. Calculate the time interval taken between two requests. After the time interval, discard the downloaded content and again give a new page request. Identify next page to request based on markov model. Remove the current bot and introduce another set of bots and continue the above steps again and again. Thus mimicking attack is implemented in our process.

Attack detection

After analyzing the client response from the server, we can able to detect the mimicking attack.A bot has to generate many more requests compared to a legitimate browser for a given time interval in order to generate the same number of requests to the web site, and therefore, the standard deviation of the attack flow is much smaller than that of legitimate browsers'. Therefore, we can differentiate them. In general, we can use any second order statistical metric to carry out the detection task. The only difference is the accuracy of the result, which depends on the granularity of the metric. Mimicking attacks can be detected using the standard deviation under the circumstance that the sufficient number condition is not held for attackers. However, there exists a problem of how accurately we detect mimicking attacks. Accuracy depends on the metric that we choose. In this paper, we have to employ second order statistical metrics. There are many candidates, such as the standard deviation, or the traditional correntropy. However, in our experiments, we found that both of them are not as good as we expected, therefore, we proposed a new second order metric based on the correntropy. Correntropy is a recently invented local tool for second-order similarity measurement in statistics. It works independently on measuring pair-wise arbitary samples. Correntropy metrics are symmetric, positive, and bounded. For any two finite data sequences A and B, suppose we have sample

$\{(A_j, B_j)\}_{j-1}^{m}$,
$m \in$ N, then the similarity of the sequences are estimated as,

$$C_{m,\sigma} = \frac{1}{m} \sum_{i=1}^{m} k_{\sigma}\,(A_i - B_i)$$

where $k(\cdot)$ is the Gaussian kernel, which is usually defined as follows:

$$k(\cdot)\ \square\ \exp\left(-\frac{x}{\quad}\right), \qquad 2\sigma^2$$

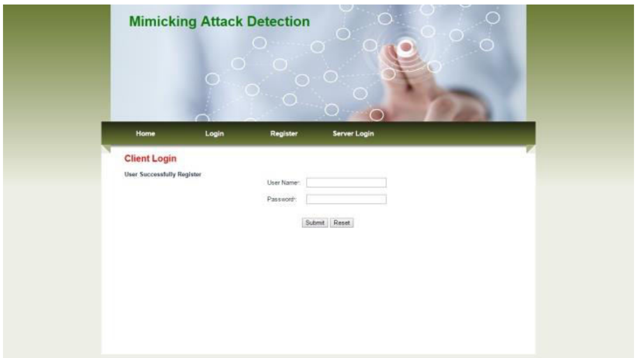
Algorithm for discriminating Mimicking attack

```
        1. Monitor the clients number of page request for a 24 hour period. Denote it C(t)
        2. Establish a mapping of the difference of flow fine correntropy of page request flows against C(t) and denote as Tf(n(t))
3. while {true}do
        Monitor the number of page request of current website. Denote it as C'(t).
While C (t)<C'(t)
        a. Collect the sample points of request flows based on our statistics methodology
        b. Claculate the flow fine correntropy C'(t)
c. $\Delta C(t) = C(t) - C'(t)$

if $\Delta C(t)$ is minimum then it is detected as mimicking attack
else
do nothing
end
end
end
```

4. EXPERIMENT RESULTS

We have done an experiment by designing both server side and the attacker side. At server side we designed a web page to provide service for uploading image files. The client can login into the server after registering and can upload the files as shown in Figure-2.



Mimicking Attack Detection

Client Login

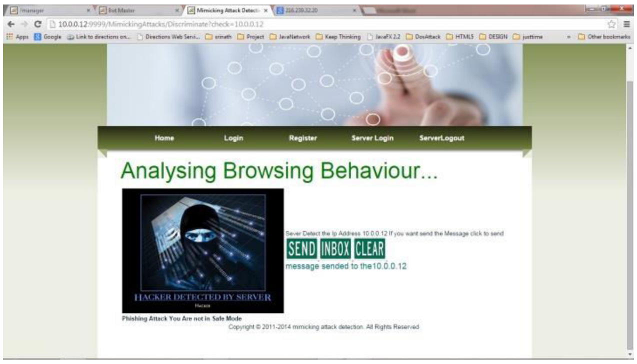
User Successfully Register

User Name:

Password:

Figure-2. Client login page.

We also designed an attacker page with different attacks as shown in Figure-3.



10.0.0.12:9999/MimickingAttacks/Discriminate?check=10.0.0.12

Analysing Browsing Behaviour...

Sever Detect the Ip Address 10.0.0.12 If you want send the Message click to send

message sended to the10.0.0.12

HACKER DETECTED BY SERVER

Phishing Attack You Are not in Safe Mode

Copyright © 2011-2014 mimicking attack detection. All Rights Reserved

Figure-3. Attack detection.

The attacker will attack the server, intruding as legitimate user based on the browsing behavior of genuine user. We performed attacks such as mimicking attack, flash crowd attack, DDos attack and phishing attack.

At server side we analysed the browsing behavior of genuine client as per statistical method and Current behavior. The similarity measure was calculated by applying fine correntropy. We also calculated based on standard deviation and other metrics. The result is shown in Figure-4.

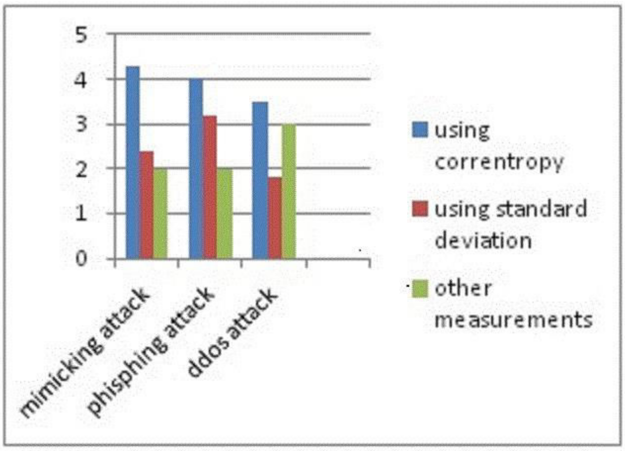


Figure-4. Similarity measure.

CONCLUSIONS

Detection of legitimate mimicking attacks is taken as a major analysis in this paper. We have established markov process model to simulate the browsing dynamics of genuine web browsers. The theoretical analysis and real world data experiments demonstrated that we cannot detect this kind of simulation in statistics. However, there is a significant condition for a successful mimicking attack. That is, the number of active bots of the botnet must not be lower than the number of active genuine users. We find that it is impossible for botnet owners to satisfy this sufficient number condition in the case of performing large scale attacks. Based on this new finding, we proposed a second order statistics based differentiation algorithm to detect this kind of attack. We done theoretical analysis and confirmed the effectiveness of the proposed detection method.

FUTURE WORK

Mimicking attacks such as membership recruitment, performance degradation attacks etc can be performed in networks which have less number of users. We can address this kind of problem by finding new methodologies. Botnet owners can also interact with other botner owners to establish a super botnet for satisfying the sufficient number condition to execute mimicking attacks. We can analyse this kind of attacks in future.

REFERENCES

[1] T. Peng, C. Leckie, and K. Ramamohanarao. 2007. Survey of network-based defense mechanisms countering the DOS and DDoS problems. ACM Comput. Surv. 39(1).

[2] Ke Li, Wanlei Zhou, Ping Li, Jing Hai and Jianwen Liu. 2009. Distinguishing DDoS Attacks from Flash Crowds Using Probability Metrics. Third International Conference on Network and System Security

[3] M. A. Rajab, J. Zarfoss, F. Monrose and A. Terzis. 2007. My botnet is bigger than yours (maybe, better than yours): Why size estimates remain challenging. In: Proc. 1st Conf. Workshop Hot Topics Understanding Botnets (HotBots'07).

[4] W. Willinger. 1995. Traffic modeling for high-speed networks: Theory versus practice. In Stochastic Networks. Springer-Verlag.

[5] S. Ledesma and D. Liu. 2000. Synthesis of Fractional Gaussian Noise Using Linear Approximation for Generating Self-Similar Network Traffic. Computer Communication Review. Vol. 30.

[6] A. Scherrer, N. Larrieu, P. Owezarski, P. Borgnat, and P. Abry. 2007. Non-Gaussian and long memory statistical characterizations for internet traffic with anomalies. IEEE Trans. Dependable Secure Comput. 4(1): 56-70.

[7] A. El-Atawy, E. Al-Shaer, T. Tran, and R. Boutaba. 2009. Adaptive early packet filtering for protecting firewalls against DOS attacks. In: Proc. IEEE Conf. Comput. Commun. (INFOCOM).

[8] J. Jung, B. Krishnamurthy and M. Rabinovich. 2002. Flash crowds and denial of service attacks: Characterization and implications for CDNS and web sites. In: Proc. World Wide Web (WWW). pp. 252-262.

[9] G. Carl, G. Kesidis, R. Brooks and S. Rai. 2006. Denial-of-service attack-detection techniques. IEEE Internet Comput. 10(1): 82-89.

[10] Y. Chen and K. Hwang. 2006. Collaborative detection and filtering of shrew DDoS attacks using spectral analysis. J. Parallel Distrib. Comput. 66(9): 1137-1151.

[11] S. Kandula, D. Katabi, M. Jacob and A. Berger. 2005. Botz-4-sale: Sur-viving organized DDoS attacks that mimic flash crowds (awarded best student paper). In: Proc. Symp. Netw. Syst. Des. Implement. (NSDI).

[12] S. Yu, S. Guo, and I. Stojmenovic. 2012. Can we beat legitimate cyber behavior mimicking attacks from botnets. In Proc. IEEE Conf. Comput. Commun. (INFOCOM). pp. 3133-3137.

[13] Z. K. Silagadze. 1997. Citations and the Zipf-Mandelbrot's law. Complex Syst. 11: 487.

[14] M. E. Crovella and A. Bestavros. 1997. Self-similarity in World Wide Web traffic: Evidence and possible causes. IEEE/ACM Trans. Netw. 5(6): 835-846.

[15] L. Breslau, P. Cao, L. Fan, G. Phillips and S. Shenker. 1999. Web caching and Zipf-like distributions: Evidence and implications. In: Proc. IEEE Conf. Comput. Commun. (INFOCOM). 126-134.

[16] S. Yu, G. Zhao, S. Guo, Y. Xiang and A. Vasilakos. 2011. Browsing behavior mimicking attacks on popular websites. In: Proc. IEEE Conf. Comput. Commun. (INFOCOM) Workshops.

[17] A. Klemm, C. Lindemann, M. K. Vernon, and O. P. Waldhorst. 2004. Characterizing the query behavior in peer-to-peer file sharing systems. In: Proc. 4th ACM SIGCOMM Conf. Internet Meas. 55-67.

[18] M. Hefeeda and O. Saleh. 2008. Traffic modeling and proportional partial caching for peer-to-peer systems. IEEE/ACM Trans. Netw. 16(6): 1447-1460.

[19] M. Mitzenmacher. 2004. A brief history of generative models for power law and lognormal distributions. Internet Math. Vol. 1.

[20] W. J. Reed and M. Jorgensen. 2003. The double pareto-lognormal distribution-A new parametric model for size distributions. Commun. Stat. Theory Methods. 33(8): 1733-1753. http://scitechnol.com/submitmanuscript-computer-engineering-and-information-technology.php.