Fool Me If You Can: Mimicking Attacks and Anti-Attacks in Cyberspace

Shui Yu, *Senior Member*, *IEEE*, Song Guo, *Senior Member*, *IEEE*, and Ivan Stojmenovic, *Fellow*, *IEEE*

Abstract—Botnets have become major engines for malicious activities in cyberspace nowadays. To sustain their botnets and disguise their malicious actions, botnet owners are mimicking legitimate cyber behavior to fly under the radar. This poses a critical challenge in anomaly detection. In this paper, we use web browsing on popular web sites as an example to tackle this problem. First of all, we establish a semi-Markov model for browsing behavior. Based on this model, we find that it is impossible to detect mimicking attacks based on statistics if the number of active bots of the attacking botnet is sufficiently large (no less than the number of active legitimate users). However, we also find 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 define a new fine correntropy metrics and show its effectiveness compared to others. Our real world data set experiments and simulations confirm our theoretical claims. Furthermore, the findings can be widely applied to similar situations in other research fields.

Index Terms—Mimicking, flash crowd attack, detection, second order metrics

1 Introduction

In this paper, we attempt to answer the following question: can we detect legitimate cyber behavior mimicking attacks from large scale botnets? The answer is: it depends. We first demonstrate this by proving 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 need to satisfy one critical condition: they have to possess a sufficiently large number of *active* bots, with no fewer than the number of active legitimate users of the simulated events. By active bots, we mean the bots that botnet owners can manipulate at the time they initiate attacks.

Botnets are the main drivers of cyber attacks, such as distributed denial of service (DDoS), information phishing and email spamming. These attacks are pervasive in the Internet, and often cause great financial loss [1], [2]. Motivated by huge financial or political reward, attackers find it worthwhile to organize sophisticated botnets for use as attack tools. There are numerous types of botnets in cyberspace, such as DSNXbot, evilbot, G-Sysbot, sdbot, and Spybot [3]. On one hand, researchers have studied botnets from various perspectives, including botnet probing events [4], Internet connectivity [5], size [6], and domain fluxing [7], [8]. On the other hand, botnet owners have at their disposal state-of-the-art techniques, such as stepping stones, reflectors, IP spoofing [1], [9], code obfuscation, memory encryption [10], and peer-to-peer implementation technology [9], [11], [12] to sustain their botnets and disguise their malicious activities and traces.

Moreover, sophisticated hackers are trying their best to mimic legitimate cyber behavior to fly under the radar [13], with popular websites becoming the major victims of cyber attacks. Experienced attackers usually simulate the phenomenon of flash crowds to disable intrusion detection systems (referred to as a *flash crowd attack*) [14], [15]. There are many other examples of mimicking attacks, such as email spamming and botnet membership recruitment. In this paper, we use legitimate web browsing and flash crowd attacks to study mimicking attack and the anti-attack issue.

Discriminating flash crowd attacks from genuine flash crowds has been explored for approximately a decade. Previous work [16]–[18] 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 defence against flash crowd attacks is the use of graphical puzzles to differentiate between humans and bots [19]. This method involves human responses and can be annoying to users. Another common method is detecting anomalies by modeling legitimate behavior, in which Markov models are the popular tools. For example, Xie and Yu [20] used the hidden semi-Markov model, and Awad and Khalil [21] employed the all-Kth Markov model to describe web browsing dynamics. Oikonomou and Mirkovic tried to discriminate mimicking attacks from real flash crowds by modeling human behavior [22]. These behavior based discriminating methods work at the application layer, and are therefore limited to the potential victim's location. An ideal detection method should be feature independent and work on a large scale, e.g. at the network layer.

- *S. Yu is with the School of Information Technology, Deakin University, Burwood, Victoria 3125, Australia. E-mail: syu@deakin.edu.au.*
- *S. Guo is with the School of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu, Fukushima 965-8580, Japan. E-mail: sguo@u-aizu.ac.jp.*
- *I. Stojmenovic is with the School of Information Technology, Deakin University, Australia; King Abdulaziz University, Jeddah, Saudi Arabia; the School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, Ontario K1N N5, Canada. E-mail: ivan@site.uottawa.ca.*

Manuscript received 23 Jan. 2013; revised 08 Aug. 2013; accepted 06 Sep. 2013. Date of publication 16 Sep. 2013; date of current version 12 Dec. 2014. Recommended for acceptance by J. Wu.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TC.2013.191

In this paper, we study mimicking attacks and detections from both sides, as attackers and defenders, which is a significant extension based on our preliminary work in [23]. From the botnet programmers' 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). Based on the research on web browsing dynamics, there are three distributions in place for the three key pieces of information. Namely, the web page popularity follows the Zifp-Mandelbrot distribution [24], the page requesting time interval follows the Pareto distribution [25], and the browsing length follows the inverse Gaussian distribution [26]. Furthermore, Borgnat et al. [27] observed a backbone of the Internet for 7 years (2001 to 2008), and compared their observation with previous ones (1998-2003) [28]. They concluded that the Internet is consistent in terms of traffic although the Internet has developed significantly. Therefore, the properties of the Internet we use in this paper are reliable. If botmasters have a sufficient number of active bots (here we mean the number of active bots is no fewer than the number of active users of a genuine flash crowd, which we will refer to as the *sufficient number* condition), then each bot can simulate one legitimate user using the three statistical distributions. As a result, it is impossible to differentiate mimicking attacks from the legitimate web browsing of a large number of browsers. We will analyze and prove this as the first goal of this paper.

However, it is hard for botnet owners to meet the sufficient number condition for certain mimicking attacks, such as flash crowd attacks. In general, people have an inappropriate and exaggerated image of hackers. They are usually described as extremely smart individuals who can easily compromise and control a large number of computers. However, this is simply not true, as there are many factors constraining the number of active bots a hacker can use, such as, widely used anti-virus software, software patching, or power off of host computers. Rajab et al. [6] discovered that the number of active bots for a given botnet is usually at the hundreds or a few thousands level, although the number of alive bots may be much larger. On the other hand, the number of legitimate users for a flash crowd is generally in the hundreds of thousands [29]–[31]. As a result, in order to execute a flash crowd attack, one bot has to carry out the task of simulating tens or even hundreds of legitimate users. To the best of our knowledge, this is a new feature to the cyber security community. By taking advantage of this new feature, defenders can detect legitimate behavior mimicking attacks. This is our second goal.

We note that in order to execute a flash crowd attack, all bots have to act exactly as a legitimate user aside from their malicious aims, e.g., using real IP addresses, submitting genuine page requests and so on. Otherwise, the attack can be easily identified by existing detection algorithms, such as detection strategies based on IP spoofing [32], hop-count [33] and packet score [34].

This paper makes the following contributions.

- 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. We prove this conclusion in theory and confirm it with real world data experiments. Under this circumstance, the current feature, statistics or browsing behavior based detection algorithms will be disabled.
- We find a new feature of today's botnets, namely, the number of active bots is usually much lower than the number of legitimate users of a genuine flash crowd. Based on this new finding, we can detect the mimicking attacks when the sufficient number condition does not hold for botmasters. We also prove this claim in theory and confirm it with simulations for the flash crowd attack cases.
- 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 flash crowd attack. Moreover, this model can be used for simulations and performance analysis for related research communities.
- We propose a new second order statistical metric for the detection purpose. 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 a fine detection accuracy.

The rest of this paper is organized as follows. Related work is discussed in Section 2, followed by the modeling, analysis and algorithm design of the browsing behavior simulation in Section 3. The effectiveness of mimicking attacks is evaluated in Section 4. In Section 5, we discuss the possibility of differentiating mimicking attacks using second order statistical metrics when the sufficient number condition does not hold, and also present the detection algorithm and simulations in this section. Further discussions are presented in Section 6. Finally, we summarize this paper and discuss future work in Section 7.

2 Related Work

2.1 Botnet and Mimicking Attacks

Cyber attackers are organizing more and more botnets to carry out their illegal tasks [12], such as launching DDoS attacks, sending spam emails, performing information phishing and collecting sensitive information.

A botnet is usually established by a botnet writer developing a program, called a bot or agent, and installing the program on compromised computers on the Internet using various techniques. All the bots from a botnet are controlled by a botmaster. The hosts running these programs are known as zombies [1], [3], [35]. For a botnet, there is one or a number of command and control (C&C) servers to communicate with bots and collect data from them. In order to protect the C&C servers and sustain the botnet, the IP address of the URL of the C&C is rapidly changed by botmasters. This is known as fast IP fluxing [36], [37]. The latest strategy for this purpose is domain fluxing, namely changing the URL of the C&C on a frequent basis [7], [8]. If these two methods are used together, it is more difficult to detect the botnet.

The majority of current DDoS attacks are performed by botnets [9]. DDoS attackers aim at exhausting the victim's

resources, such as network bandwidth, computing power, operating system data structures [3]. In order to sustain their botnets, botnet owners employ various strategies against detection and traceback. For example, using a stepping stone to protect their C&C centers, IP spoofing and reflectors to hide the real addresses of bots [1], [9], code obfuscation, memory encryption to guard against reverse engineering on bot programs [10], and peer-to-peer implementation technology to improve the sustainability of the whole botnet [11], [9], [12]. They also simulate the phenomenon of a flash crowd to fool detection algorithms [20], [22]. With the establishment of a botnet, it is easy for attackers to execute flash crowd attacks, as attacking bots are distributed all over the world, and some of the host computers are also used by genuine viewers to access the victims. Defenders do not have an effective method to deal with botnets on large scale networks.

Determining the size of a botnet is important to both attackers and defenders. Researchers have employed various methods to attain the size of botnets, such as botnet infiltration [3], DNS redirection [38] and external information [6]. A direct method to count the number of bots is to perform botnet infiltration and count the bot IDs or IP addresses. Stone-Gross et al. used this method and reported that the footprint of the Torpig botnet is 182,800, and the median and average size of the Torpig's live population is 49,272 and 48,532, respectively [3]. The live population means the active bots counted during the whole observation period. Another method is to use DNS redirection. Dagon, Zou and Lee [38] analyzed captured bots using honeypot, identified the C&C server using source code reverse engineering tools, manipulated the DNS entry which is related to a botnet's IRC server, and then redirected the DNS requests to a local sinkhole. This meant they could count the number of bots in a botnet. They reported that the number of footprints of a botnet can reach 350,000. However, there are far less active bots that a botmaster can use to initiate a flash crowd attack than its footprint and live population. There are a number of reasons for this, such as host power off, anti-virus patching and system reinstallation. Rajab et al. investigated this issue, and pointed out that the number of active bots for a given botnet is usually at the hundreds or a few thousands level [6].

2.2 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 [24]. These findings are widely used in research papers, such as [39] and [40].

For a given website, we assume that there are $N(N > 0)$ web pages in total, and they are sorted in terms of popularity from the most to the least as $w_1, w_2, \ldots, w_N$. Let random variable W be the requested web page, and $Pr[W = w_i]$ be the request probability of page w_i. Then the Zipf-Mandelbrot distribution can be formulated as

$$Pr[W = w_i] = \frac{\Omega}{(i+q)^{\alpha_z}}, \tag{1}$$

where $\alpha_z(\alpha_z > 0)$ is the *skewness* factor, which dominates the skewness of the distribution, and $q(q \geq 0)$ is the *plateau* factor, which makes the probability of the highest ranked pages flat. The Zipf-Mandelbrot distribution becomes the Zipf distribution when $\alpha_z = 1$, and it becomes the Zipf-like distribution when $q = 0$. Since $\sum_{i=1}^{N} Pr[W = w_i] = 1$, $\Omega = (\sum_{i=1}^{N} \frac{1}{(i+q)^{\alpha_z}})^{-1}$.

If all the bots of a botnet use $Pr[W = w_i]$ to decide the pages to browse, then the page request distribution on the victim's side follows the Zipf-Mandelbrot law, and we are unable 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 [25] (confirmed also by [41] and [42]). Let random variable T be the page viewing time for a given web page, and t_m be the minimum page viewing time for a given website. For a given web page with viewing time t, the probability of the viewing time distribution is defined as follows.

$$Pr(T = t) = \alpha_p t_m^{\alpha_p} t^{-(\alpha_p+1)}, \tag{2}$$

where $t_m \leq t$, and α_p is also called the *Pareto index*.

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 L, namely the number of pages a user generally views during a browsing session. Huberman et al. indicated that the probability of L follows the two-parameter inverse Gaussian distribution [43], formulated as follows.

$$Pr[L = l] = \sqrt{\frac{\lambda}{2\pi l^3}} \exp\left[\frac{-\lambda(l-\mu_l)^2}{2\mu_l^2 l}\right], l \in \mathbb{N}, \tag{3}$$

where μ_l is the mean, and λ is the shape parameter. The inverse Gaussian distribution approximates the Gaussian distribution when $\lambda \to \infty$.

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 botmasters to possess a sufficient number of active bots to carry out flash crowd attacks.

2.3 Similarity Measurement

Similarity measurement has been extensively explored for many years, with researcher inventing many metrics, including first order and second order metrics. For example, mean and the Kullback-Leibler distance are first order metrics, while standard deviation and correntropy [44] are second order metrics. It is not difficult for attackers to exhaust their active bots to generate the same average number of page requests as a flash crowd. Therefore, first order metrics are vulnerable to sophisticated mimicking attacks. However, when the sufficient number condition is not met for attackers, the flow feature of standard deviation or second order statistics will reveal the difference between a genuine flash crowd and a flash crowd attack.

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

$$\hat{V}_{m,\sigma}(A,B) = \frac{1}{m}\sum_{j=1}^{m} k_\sigma(A_j - B_j), \tag{4}$$

where $k_\sigma(\cdot)$ is the Gaussian kernel, which is usually defined as follows.

$$k_\sigma(\cdot) \triangleq \exp\left(-\frac{x^2}{2\sigma^2}\right). \tag{5}$$

Correntropy is widely used in various disciplines, such as face recognition [45].

3 Legitimate Browsing Behavior Modeling

In this section, we act as attackers in order to study how to mimic the browsing behavior of a legitimate web viewer. We establish a mathematical model based on the semi-Markov Chain process for this purpose, and describe a mimicking algorithm.

3.1 An Individual View of Web Browsing Behavior

If a botnet owner has a sufficient number of active bots (the sufficient number condition holds), then he can use one bot to act as one legitimate viewer. The problem for botnet writers is how to statistically simulate the behavior of a legitimate browser. A browsing session for a user is shaped by three factors: which pages to request, time duration of viewing a page and how many pages to go through. As discussed in Section 2, these three factors are determined by the Zipf-Mandelbrot, the Pareto and the inverse Gaussian distributions, respectively.

We suppose that a given potential victim web site has $N(N > 0)$ web pages in total. Furthermore, we can obtain the ranking $w_1, w_2, \ldots, w_N$ of the N pages in terms of popularity by observing legitimate page requests, from the most popular one to the least popular.

Definition 1 (Flow). *A flow is a group of HTTP requests that share the same source IP and destination IP addresses.*

For a given observation point, we count the number of HTTP requests of each flow for the given time intervals. As a result, a flow is a sequence of numbers. We denote this flow as $F(\mu, \sigma)$, where μ is the mean of the flow and σ is the standard deviation of the flow. Physically, μ is the average number of HTTP requests for a web page over the observation time intervals. Furthermore, we denote a genuine flash crowd as $F_c(\mu_c, \sigma_c)$ (c stands for flash crowd), an original attack flow as $F_a(\mu_a, \sigma_a)$ (a stands for attack), and an aggregated flow of attack flows as $F_g(\mu_g, \sigma_g)$ (g stands for aggregated).

To describe the browsing behavior of a legitimate web viewer, we extend the classical Markov model to a four parameter semi-Markov model as follows.

$$\Lambda = (P, T, L, \pi),$$

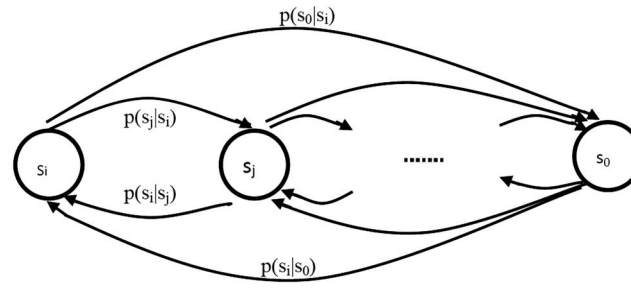


Fig. 1. The state transition of the four parameter semi-Markov Chain model for browsing behavior.

where P, T, L, π represents the state transition matrix, duration at the current state, browsing length, and the initial probability distribution of the states, respectively.

In our model, we treat every web page as an unique state. We use s_i to denote the state of web page w_i. In general, a viewer may start his browsing session with any page of the web site, and follow the hyperlinks of the current page to access other pages. At the same time, there is a possibility during the browsing session that a viewer may key in a URL or choose a URL from his favorite list. This results in a jump from page i to page $j(j \neq i)$ where there is no hyperlink for page j in page i. In order to deal with this situation, we define a special page, *null page* (denoted as page w_0, and state s_0 in the state space). As a result, the jump from page i to page j can be interpreted as: page i transfers to the null page, and then the null page transfers to page j. If a viewer terminates his browsing session at this web site or leaves the web site then the viewer stays at state s_0.

We use variable q_t to represent the state at time t.

Let p_{ij} represent the transition probability from state i to state j, namely,

$$p_{ij} \triangleq Pr[q_{t+1} = s_j | q_t = s_i], 0 \leq i, j \leq N.$$

Then the state transition matrix can be represented as

$$P = \{p_{ij}\}, 0 \leq i, j \leq N,$$

where $\sum_{j=0}^{N} p_{ij} = 1$, and $\sum_{i=0}^{N} p_{ij} = 1$.

The state transition is shown in Fig. 1.

We use a set $\{h_i\}(0 \leq i \leq N)$ to represent all the hyperlinks of page w_i (including the special hyperlink to page w_0). When a bot progresses its browsing from a current web page w_i, suppose the next web page is w_j, then the transition probability p_{ij} can be calculated as follows.

$$p_{ij} = \frac{Pr[W = w_j]}{\sum_{k \in \{h_i\}} Pr[W = w_k]}. \tag{6}$$

The second parameter of the model, T, represents the time duration a viewer stays at the current state.

$$T = \{t_i\}, i = 0, 1, \cdots, N; 0 \leq t_i \leq +\infty,$$

where t_i denote the time duration of state s_i. Physically, it is the time interval of viewing the current page. T follows the Pareto distribution, which has been defined in equation (2).

The third parameter of the model, L, represents the browsing length of the current session.

$$L = \{l_i\} = \{0, 1, \cdots, N\}.$$

It follows the inverse Gaussian distribution, which is defined by equation (3).

The last parameter π is the probability that a viewer selects a page as the first page of his browsing session, and parameter $\pi_i (0 \leq i \leq N)$ indicates the probability of the initial state s_i,

$$\pi_i \triangleq Pr[q_1 = s_i | q_0 = s_0], i = 0, 1, \ldots, N,$$

where $\pi_0 = 0$, $\sum_{i=0}^{N} \pi_i = 1$, and $\pi_i (1 \leq i \leq N)$ follows the Zipf-Mandelbrot distribution, which has been defined by equation (1).

In order to find the four parameters for the semi-Markov model, we should observe the potential victim for sufficient time in attack free cases, and based on the data collected, we can extract the four parameters. Of course, this training should be taken periodically to update the parameters to reflect the ever changing web browsing behavior.

With this four parameter semi-Markov model in place, every bot can independently simulate a legitimate web viewer's browsing behavior.

3.2 A System View of Web Browsing Behavior

In a genuine flash crowd scenario, we are interested to see the various phenomenon in a system viewpoint. For example, 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.

In order to answer these questions, we need one more parameter: the number of active web viewers for a given time point t, which we denote as $n(t)$. $n(t)$ varies against the time point of a day. Intuitively, there are more web viewers during working time than early morning. We have conducted a 30 days observation on $n(t)$ for every 30 minutes, and found that $n(t)$ was stable day after day. In Fig. 2, we present our observation of $n(t)$ on June 1, 2010 of a popular news web site.

From Fig. 2, we find there is significant variation among the number of web viewers. For example, there are less than 100 concurrent viewers at 5 or 6 am, however, it soars to more than 1,000 around 11 or 12am, and is relatively stable in the afternoon and into the middle night, with around 400 viewers. There are many factors that impact $n(t)$, e.g. time zone, holidays, weekdays or weekends. Therefore, it is hard to have a closed form of $n(t)$. However, this does not impact our modeling and analysis.

Following the properties of the Pareto distribution, when $\alpha_p > 1$, the mean of the viewing time is

$$\overline{T} = \frac{t_m \alpha_p}{\alpha_p - 1}. \tag{7}$$

Therefore, we can obtain an average of frequency F (the number of pages a browser reads for a unit of time) as

$$\overline{F} = \frac{1}{\overline{T}} = \frac{\alpha_p - 1}{t_m \alpha_p}. \tag{8}$$

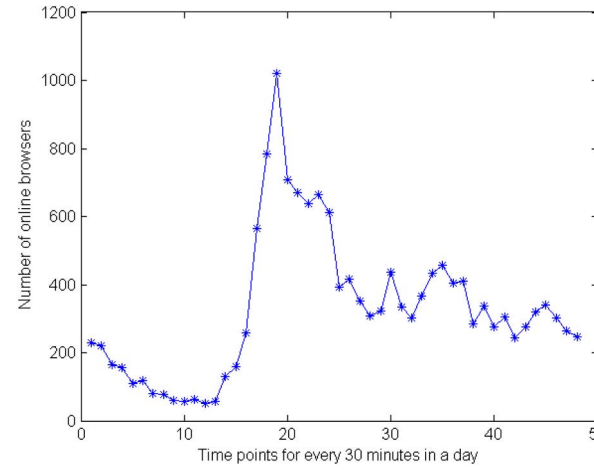


Fig. 2. The distribution of the number of browsers for every 30 minutes on June 1, 2010 for a popular news web site.

The number of page requests for a given time point t, $R(t)$, observed at the server end can be expressed as

$$R(t) = n(t) \cdot \frac{\alpha_p - 1}{t_m \alpha_p}. \tag{9}$$

If we break this down further, for the same scenario, the number of requests for page $w_i (1 \leq i \leq N)$ at time point t is

$$\begin{aligned} R(t, w_i) &= Pr[W = w_i] \cdot R(t) \\ &= \frac{\Omega}{(i+q)^{\alpha_z}} \cdot n(t) \cdot \frac{\alpha_p - 1}{t_m \alpha_p}. \end{aligned} \tag{10}$$

Moreover, the duration of a browsing session for a user is dominated by

$$\begin{aligned} Pr[D = t \cdot l] &= Pr[T = t, L = l] \\ &= \alpha_p t_m^{\alpha_p} t^{-(\alpha_p + 1)} \cdot \sqrt{\frac{\lambda}{2\pi l^3}} \exp\left[\frac{-\lambda (l - \mu_l)^2}{2\mu_l^2 l}\right]. \end{aligned} \tag{11}$$

Based on Wald's theorem, the mean of the duration of browsing sessions is

$$\overline{D} = \overline{T} \cdot \overline{L} = \frac{t_m \alpha_p}{\alpha_p - 1} \cdot \mu. \tag{12}$$

Suppose we observe the number of users every Δt ($\Delta t < \overline{D}$) time interval, and we have conducted k observations, then during this time interval $\Delta T = k\Delta t$ ($\Delta T > \overline{D}$), the number of unique users accessing the system is

$$n(\Delta T) = \frac{\Delta t}{\overline{D}} \cdot \sum_{i=1}^{k} n(t_i). \tag{13}$$

Equation (13) indicates the number of unique bots that a botmaster has to possess in order to carry out a mimicking attack for a duration of ΔT.

Based on the four parameter semi-Markov model and the analysis, we can make the following conclusion.

Theorem 1. *If a botnet owner has a sufficient number of active bots, then he can successfully mimic a given flash crowd.*

Proof. Suppose there is a genuine flash crowd flow, $F_c(\mu_c, \sigma_c)$ (as defined in Definition 1), and it is generated by $n(t)$ legitimate browsers at any given time point t.

In order to successfully simulate it, we firstly observe the flash crowd, and collect the page requests data. Once the collected data set is sufficient, we can extract the parameters for the semi-Markov model $\Lambda = (P, T, L, \pi)$. A page request software package can be programmed, and injected to all bots. For any give time point t, the botnet owner activates $n(t)$ bots to execute their semi-Markov model based programs.

From a system and statistical viewpoint, the flash crowd attack is the same as the mimicked genuine flash crowd, no matter what perspective we use, such as total number of page requests, or number of requests for a specific web page, number of unique browsers for a given time interval. □

Theorem 1 indicates that a flash crowd can be successfully simulated in terms of statistics. At the same time, we note that a critical element for the simulation is that the sufficient number condition holds, which is dominated by equation (13).

3.3 The Legitimate Behavior Mimicking Algorithm

In order to simulate legitimate flows or a flash crowd of a web site, attackers have to firstly study the subject and extract related browsing dynamic parameters. There are practical methods to obtain the parameters. For example, attackers can observe the traffic to the victim on one or a few compromised routers. Due to the scale free property of power law, which includes both the Zipf-Mandelbrot distribution and the Pareto distribution, they can obtain the parameters appropriately based on a partial observation.

When all the parameters are in hand, we can arrange active bots to carry out a mimicking attack. We present the implementation detail of the mimicking attack in Algorithm 1.

Algorithm 1: The mimicking attack algorithm

1. Observe the target web site, and extract the related browsing dynamic parameters $\alpha_z, q, \alpha_p, \lambda, \mu_l, n(t)$.
2. Initialize the parameters of the semi-Markov model Λ.
3. Take $n(t)$ bots from a set of active bots, $\{bots\}_t$, and instruct these bots to run independently.
4. **foreach** *bot* $\in \{bots\}_t$ **do**
   - 4.1. Generate a random number rnd.
   - 4.2. Identify an initial page according to equation (1) with rnd;
   - 4.3. Decide the browsing length L for this bot using equation (3) with rnd;
   - 4.4. $j = 1$;
   - **while** $j \leq L$ **do**
     - a. Submit the request and discard the downloaded content;
     - b. Wait for a time interval decided by equation (2) and rnd;
     - c. $j = j + 1$;
     - d. Identify the a new page to request following the semi-Markov model Λ.
   - **end**
   - 4.5. Remove the current bot from the set $\{bots\}_t$;
   - **end**
5. Introduce new bots and update $n(t)$;
6. Go to step 4.

This algorithm can be used to launch a flash crowd mimicking attack if we have a target flash crowd to obtain the browsing dynamic parameters. This methodology can be applied to other types of mimicking attacks, such as email spamming, botnet membership recruitment or virus spreading.

4 Effectiveness of the Mimicking Attack Model

In order to demonstrate the effectiveness of the proposed mimicking attack algorithm, we collected the web dynamic data of a popular news web site for thirty days (from June 1 to 30, 2010) at a major backbone network center. The data for two days (the 1st and the 30th of June 2010) has been explicitly extracted for our experiments. We used the data from day 1 as a training data set, and extracted the key parameters from the data set to populate the parameters of the semi-Markov model. We call these as *training data* and *target data*, respectively. We arrange the data sets into a matrix for both days: each web page of the web site is a row in the matrix, and every column denotes the number of requests in a thirty minute duration.

Firstly, we simulate the number of legitimate browsers at different time points for the target data. As the training data has offered us this information as $n(t)$, we simulate the target $n'(t)$ as follows.

$$n'(t) = n(t) + (0.5 - x) \cdot |n(t) - n(t-1)|, \qquad (14)$$

where x is an uniform random variable between 0 and 1, and $t = 0, 1, \ldots, 47$. Due to the fact that the variation could be positive or negative, we therefore use $0.5 - x (-0.5 \leq 0.5 - x \leq 0.5)$ multiple the term $|n(t) - n(t-1)|$ to reflect the fact.

A comparison between the real data of requests for day 30 and the mimicking requests of the target data is shown in Fig. 3. We found that the mimicking data and the real data were very close.

Secondly, we needed to identify the α_z and α_p for the semi-Markov model. Both Zipf-Mandelbrot and the Pareto distributions are power law distributions. For the sake of simplicity, we assume $\alpha_p = \alpha_z$, and denoted it as α. We applied the Least Square Summary (LSS) method on the training data set, and found that the best α was 1.31. Moreover, we found that $q = 4$ under the condition of $\alpha = 1.31$.

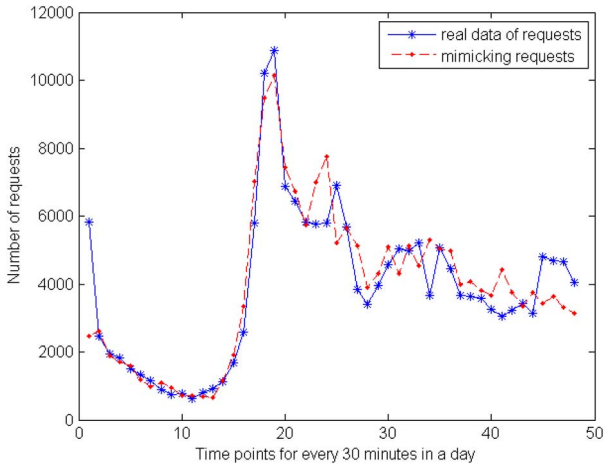


Fig. 3. The comparison of the total number of requests for every 30 minutes between the real data of the target data (day 30) and the data we simulated based on the training data (day 1).

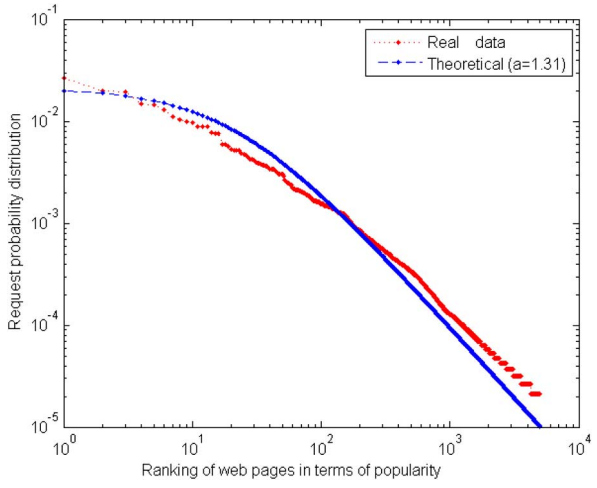


Fig. 4. The web page request distribution of the target data and that generated by our mimicking model for a whole day.

Following the previous research [20], [25], [42], [46], we took the minimum viewing time $t_m = 30$ seconds. Following the properties of the Pareto distribution, we obtained the mean viewing time for a page as 126.77 (seconds). Humerman et al. indicated that $\mu_l = 15$ in their experiments for browsing length [43]. Therefore, the mean life span for a browsing session is $\overline{T} = 15 \times 126.77 = 1901.55$ (seconds) following equation (12), which is around 31.69 minutes.

With these parameters in place, we can populate them into the semi-Markov model, and execute individual bots independently to carry out the mimicking. At the same time, we also have the real data of the target day.

We studied the request distributions for a twenty four hour period to test the effectiveness of our simulation. We compared the distribution generated by our mimicking model against the target data, with the results presented in a log-log graph in Fig. 4. Once again, the difference between the mimicking attack algorithm generated data and the real data is very limited. We are unable to differentiate them through statistical methods.

On the other hand, we further investigated the effectiveness of the simulation model in terms of a short time period.

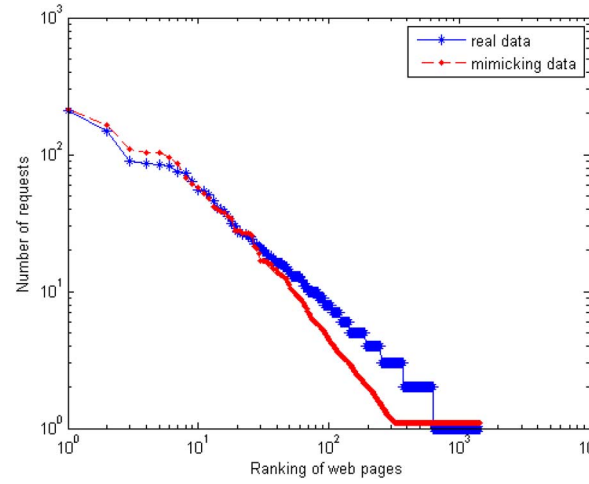


Fig. 5. A comparison of web page request distribution between the real target data and that generated by our mimicking model in a short time period (30 minutes).

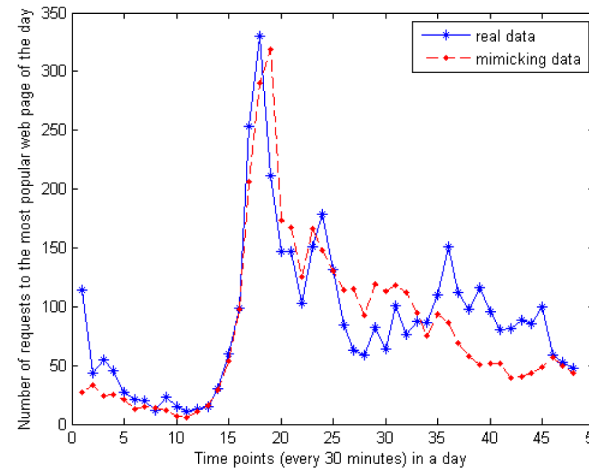


Fig. 6. A comparison of a whole day's web page request distribution for the most popular page of the day between the real target data and that generated by our mimicking model.

We took just one sample duration (6:00 - 6:30 pm) of the data set as our study object. For this case, our simulation model indicates that we need 326 bots based on equation (14). In Fig. 5, we present the result of this comparison, further confirming that the mimicking model is effective. Once again, we are unable to discriminate it in terms of statistics.

Furthermore, we investigated the whole day request distribution for a specific web page. We identified the most popular web page of the target day, and simulated the request distribution based on our model with the extracted parameters from the training day data set. The result is shown in Fig. 6. From statistics perspective, we believe the simulation was successful.

In summary, the mimicking attack algorithm was effective on different scales, whether twenty four hours or thirty minutes, and also effective for the whole request distribution or the request distribution for a single wb page. We conclude that it is impossible to discriminate such attacks from their statistics.

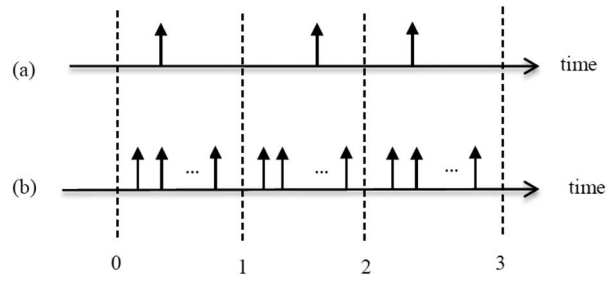


Fig. 7. The comparison on page requesting sequence between a legitimate flow (a) and a mimicking flow (b) when $\rho < 1$.

5 Mimicking Attack Detection

As we discussed previously, if the sufficient number condition holds for a botnet owner, then he can successfully simulate a cyber event. However, the sufficient number condition is hard to meet in some instances, such as flash crowd mimicking attacks. The problem is how to detect flash crowd attacks when the sufficient number condition is not met by botnet owners.

5.1 When the Sufficient Number Condition Does Not Hold

We notice that the number of active bots that a botmaster possibly has is far less than the number of legitimate users of a flash crowd. The number of legitimate browsers of a genuine flash crowd is usually in the hundreds to thousands level. For example, from the World CUP 98 data set [29] and the Auckland data set [30], we see more than 3,000 requests per second. If a browser views a page for 2 minutes on average (based on the knowledge of the previous section), then we roughly have more than 360,000 concurrent browsers. However, based on previous research [6], the number of active bots of a botnet is usually only at the hundreds or a few thousands level.

This finding is new and has never been used by researchers, to the best of our knowledge. To obtain the same number of requests for a given time interval of a flash crowd, botnet owner has to exhaust every active bot to generate tens, even hundreds times of page requests than that of a legitimate user for a given time interval. This action results in the time interval between two consecutive requests of a simulating flow being much shorter than that of a legitimate flow, and this difference can be uncovered using second order metrics.

Before we progress on the detection of mimicking attacks, we confine our discussion within the following assumptions in order to make our analysis feasible. Of course, the conditions can be removed in practice, which will make it more complex to deal with mimicking attacks. We will further discuss about these in the limitation and further discussion Section.

- We suppose attackers can obtain the browsing dynamics of the victim. They can obtain the statistics through observing a close router to the victim, or using other strategies, such as internal attacks and social engineering.
- We assume the attacking botnet is homogeneous.

For a given time point t, let $\rho(t)$ be the ratio at time t, $n_c(t)$ be the number of legitimate users that a botmaster expects to simulate, and $n_a(t)$ be the number of active bots a botmaster has.

In order to specify this new found feature, we make a definition of *ratio* as follows.

Definition 2 (Ratio). *A ratio ρ is the percentage of the number of active bots over the target number of legitimate users for a given time point. It can be expressed as*

$$\rho = \frac{n_a(t)}{n_c(t)}. \tag{15}$$

We only discuss the case $0 < \rho < 1$. Otherwise, we cannot detect the mimicking attack as mentioned in the previous discussion.

In order to explain the difference between a legitimate flow and an aggregated mimicking flow, we present the new found feature using Fig. 7 (a upward arrow represents a page request at the time point). In the diagram, (a) represents a page requesting sequence of a legitimate user. We suppose there is one page request per time unit for the user, and the time interval between two consecutive requests is a random variable. (b) represents the sequence of page requests of a bot. A bot has to generate $\frac{1}{\rho}$ page requests for a time unit.

Let the mean and standard deviation of a legitimate flow be μ_c and σ_c, and the mean and standard deviation of a mimicking flow be μ_a and σ_a. Suppose σ_c and σ_a are instances of a same distribution. Then

$$\begin{cases} \mu_a = \frac{1}{\rho} \cdot \mu_c \\ \sigma_a = \frac{1}{\rho} \cdot \sigma_c. \end{cases} \tag{16}$$

Secondly, we expect to see the difference at the destination computer (the victim) between a genuine flash crowd and a mimicking attack. The flows (either legitimate flow or mimicking flow) merge together at the destination computer, and both have the same mean; however, the standard deviation should be different as the mimicking flows have a much smaller standard deviation than that of legitimate flows. To illustrate this, we provide an example here. Suppose aggregated attack traffic comes from 50 attack flows following Gaussian distributions with $\mu_a = 10$ and $\sigma_a = 1\% \cdot \mu_a = 0.1$. A genuine single distribution is a Gaussian distribution with $\mu'_f = 500$ and $\sigma'_f = 1\% \cdot \mu'_f = 5$. We observe the phenomenon at a destination point according to the number of requests, and the results are shown in Fig. 8. We can see from Fig. 8 that although the single flash crowd and the aggregated attack flow share the same mean (number of requests), they possess a different standard deviation.

Lemma 1. *Suppose we have k ($k > 0$) flows, $F_a^i (i = 1, 2, \ldots, k)$, which are generated by any function with mean μ_a and standard deviation σ_a. If we merge these k flows together in any style, to obtain an aggregated flow $F_g(\mu_g, \sigma_g)$, then $\sigma_g = \sigma_a$.*

Proof. We transform any flow $F_a^i (1 \leq i \leq k)$ as

$$F_a^i = \mu_a^i + f(\sigma_a^i),$$

where μ_a^i is the mean of flow F_a^i, and $f(\sigma_a^i)$ is a random variable depending on variable σ_a. Therefore, $E[(f(\sigma_a^i)] = 0$ for any $i (1 \leq i \leq k)$.

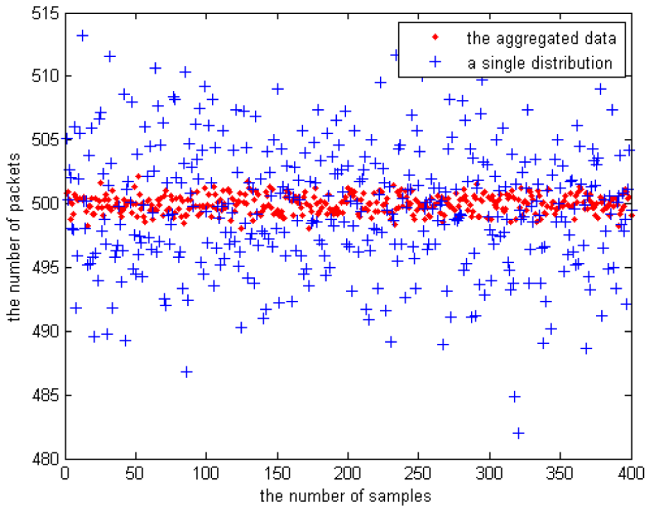


Fig. 8. The difference between a mimicking attack and a genuine flash crowd from the viewpoint of the destination computer.

Flow $F_g(\mu_g, \sigma_g)$ is the sum of k different flows. Therefore,

$$F_g(\mu_g, \sigma_g) = \sum_{i=1}^{k} F_a^i.$$

Then

$$\begin{aligned}
\sigma_g &= \left(E[(F_g - \mu_g)^2]\right)^{\frac{1}{2}} \\
&= \left(E\left[\left(\sum_{i=1}^{k} F_a^i - \sum_{i=1}^{k} \mu_a^i\right)^2\right]\right)^{\frac{1}{2}} \\
&= \left(E\left[\left(\sum_{i=1}^{k} f(\sigma_a^i)\right)^2 - 0\right]\right)^{\frac{1}{2}} \\
&= \left(E\left[(f(\sigma_a) - E[(f(\sigma_a)])^2\right]\right)^{\frac{1}{2}} \\
&= E[\sigma_a] = \sigma_a. \qquad \square
\end{aligned}$$

Theorem 2. *When the sufficient number condition does not hold for botnet owners, defenders can discriminate mimicking attacks from genuine flash crowds based on the standard deviation of flows.*

Proof. Based on our previous discussion, 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'.

The flash crowd $F_c'(\mu_c', \sigma_c')$ is a single distribution, and its mean μ_c' is much larger than μ_a. For the same reason, $\sigma_c' \gg \sigma_a$. In order to match the request volume of the flash crowd, a mimicking flow is actually an aggregated flow of a number of $F_a(\mu_a, \sigma_a)$, as lemma 1 indicated, the standard deviation of the aggregated flows is the same as a single attack flow, σ_a, namely, $\sigma_a < \sigma_c'$. Therefore, we can differentiate them. $\square$

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 propose a new second order metric based on the correntropy as follows.

$$V_{m,\sigma}(A, B) = \frac{1}{m}\sum_{j=1}^{m} \frac{1}{2\sigma} \cdot exp[(A_j - B_j)^2]. \tag{17}$$

We name the proposed metric as *fine correntropy*. Compared with the definition of correntropy in (4), we can see that the fine correntropy inherits the symmetric, positive properties of the correntropy metric. The only difference is that the proposed metric possesses a much smaller granularity, which is what we need. The advantage of the fine correntropy will be shown through experiments in the next section.

We note that our proposed detection method possesses an accuracy problem as other statistical methods, such as false negative and false positive. For example, for unexpected events or some incidents, the variation of legitimate flows may larger than the given threshold of detection, this causes a false positive. On the other hand, a mimicking attack with a variation smaller than the given threshold of detection results in a false negative.

5.2 The Mimicking Attack Detection Algorithm

In order to detect the flash crowd mimicking attacks, we have to establish a profile of the fine correntropy of flows for the non-attack cases, and identify an anomaly when the variation of flow fine correntropy is sufficiently different from the normal value. We can manually supervisor the network traffic of the web site, which we try to protect, for a number of given periods (we use twenty four hours as one period). We take the periods that are attack free as benchmark for anomaly detection. As a result, we can establish a map of the number of page request $R(t)$ against time t for a twenty four hour period. The granularity of time t could be at the second or minute level in order to detect attacks in time.

The detection algorithm is shown in detail in Algorithm 2.

Algorithm 2: The mimicking attack detection algorithm

1. Establish the profile of $R(t)$ for a 24 hour period;
2. Establish a mapping of the variation of flow fine correntropy of page request flows against $R(t)$, and denote as $V_f(n(t))$;
3. **while** {*true*} **do**

   Monitor the volume of page requests of the web site, denote as $R'(t)$;

   while $\{R'(t) \geq R(t)\}$ **do**

   a. Following statistical methodology, sample request flows for sufficient sample points;

   b. Calculate the flow fine correntropy $V_f'(t)$;

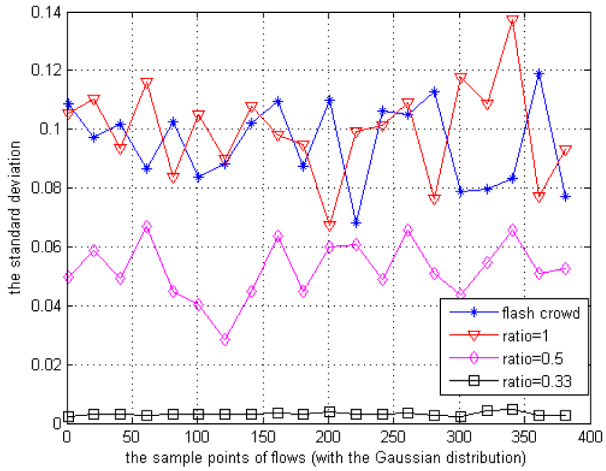


Fig. 9. The variation of standard deviation as a metric between mimicking flows under different ratios compared to that of flows from a flash crowd.

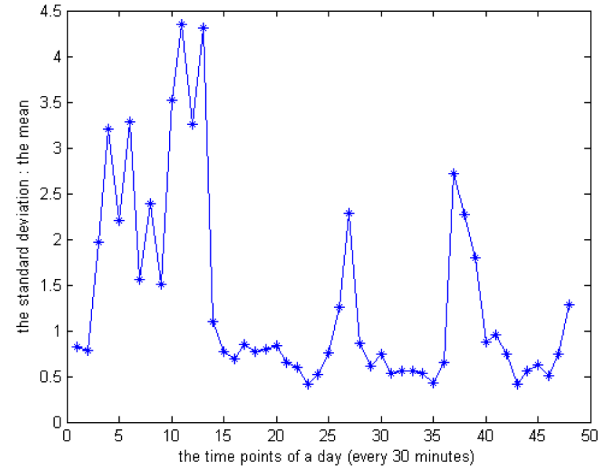


Fig. 10. The ratio of standard deviation over mean of legitimate users of a popular news web site for every thirty minutes for a twenty-four hour duration.

c. $\Delta V_f(t) = |V_f(t) - V'_f(t)|$;
d. **if** $\Delta V_f(t)$ is sufficient **then**
it is mimicking attack
else
do nothing
end
end
end

We note that the goal of the proposed method is to detect flash crowd mimicking attacks, rather than identify attack sources, which is referred to as *traceback*. We refer interested readers to the recent work of [47], [48], and [49] for DDoS traceback.

5.3 Effectiveness of the Detection Method

As we have proven that it is possible to discriminate a mimicking attack from a genuine flash crowd, we demonstrate the effectiveness of the proposed detection method using simulations in this section. We note that the proposed detection method is independent of any specific flow distribution because the parameters that we use are the second order statistical data, e.g. the standard deviation. Without loss generality, we use the Gaussian distribution for the traffic flows in the following experiments.

A direct and simple metric is the standard deviation of flows. We then investigated the variation of standard deviation for different ratio ρ. In detail, we set a target flow as legitimate flash crowds. For a given ρ, the mean and the standard deviation of a mimic flow is then $\frac{1}{\rho}$ of that of the target flows, respectively. The result is shown in Fig. 9.

We found that there is no way to detect the mimicking attack when $\rho = 1$ (namely, the number of active bots is the same as the number of active legitimate users), and it is hard to do so when $\rho = 0.5$ (namely, the number of active bots is half of the number of active legitimate users). However, we can clearly detect mimicking attacks when $\rho = 0.33$. Namely, we can detect when the number of active bots is no more than $\frac{1}{3}$ of the number of legitimate users $(n_a(t) \leq \frac{1}{3} n_c(t))$.

However, the number of legitimate users varies from day to day, and this fact has a critical impact on our detection. To find out the variation of the number of legitimate users for a given web site, we collected the number of users for every 30 minutes for 30 days of a popular news web site, and obtained the mean and the related standard deviation of each sample point in a 24 hour scale. In Fig. 10, we present the ratio of standard deviation over the mean in terms of the number of users for every 30 minutes for a 24 hour duration. Based on Fig. 10, we find that the variation of users could be as high as 5 times in the early morning. In other words, it is possible that the number of user varies to only 20% as we expected.

We define *detection effectiveness* as the threshold (in terms of ratio) that we can successfully detect mimicking attacks, in other words, if the ratio is lower than the threshold, then we can effectively detect the mimicking attack using the proposed second order metric method. We know that the detection effectiveness is decided by two elements: detection accuracy of the selected metric and the possible variation of the number of legitimate users.

Combined the metric accuracy of using standard deviation (which is $\frac{1}{3}$ in this case) and the observation on variation of the number of legitimate users that we gain from Fig. 9 (which is around $\frac{1}{5}$), we find that the detection effectiveness that we can achieve is $\frac{1}{15}(\frac{1}{3} \cdot \frac{1}{5} \approx 6.7\%)$. This means if a botmaster can organize a sufficient number of active bots, for example, just more than to 6.7% of the number of active legitimate users, then he can fly under the radar, or cause a false negative for our detection systems.

In order to improve the detection effectiveness of the proposed method, we will use the proposed fine correntropy to replace the standard deviation as a metric to repeat the same experiments as we conducted in Fig. 9. The result is shown in Fig. 11. We can see that it is impossible to

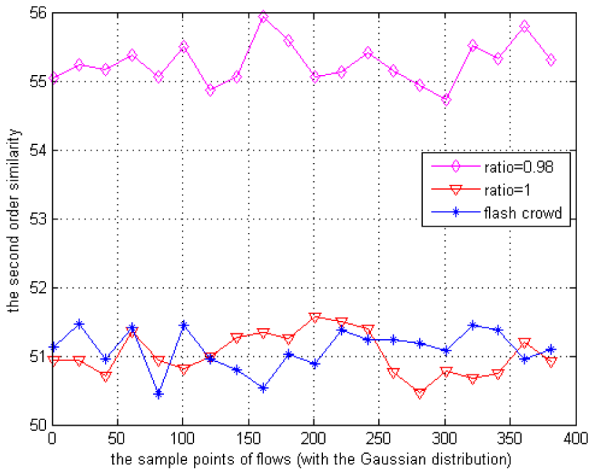


Fig. 11. The variation of the second order similarity between mimicking flows and that of a single genuine flow.

discriminate the attack when $\rho = 1$, but we can clearly differentiate them when $\rho = 0.98$. Combined with the variation of the number of legitimate users we found in Fig. 10, we obtain the threshold for effective detection is around $19.6\% = \frac{1}{5} \cdot 0.98$, which means a botmaster has to possess more than 19.6% of the number of legitimate users to fly under the radar. In other words, the detection accuracy improves around 3 times using the proposed fine correntropy metric compared to using the standard deviation as the metric.

6 Limitations and Further Discussion

In this paper, we have explored the possibility of successful mimicking attack and detection, which tackles only a small part of the whole problem. We discuss the limitation of the current paper as follows.

Firstly, there are many legitimate events with small number of active users in cyberspace, which make it easy for botnet owners to meet the critical number condition to successfully mimicking those kind of events to carry out their malicious goals. Based on our analysis, our method is unable to detect them. However, the existing graphic puzzle method [19] is effective in eliminating them at the potential victim location although this method is annoying to users.

Secondly, the visual graph based method is not strictly reliable to judge a mimicking is successful or not. In practice, our benchmarks for page popularity, viewing time interval and browsing length are statistical data, therefore, when a mimicking result is very close to the benchmark, we cannot differentiate it. Moreover, to the best of our knowledge, researchers use visual graphs to judge whether a distribution follows the power law or not. Although this method is feasible in practice, it lacks mathematical rigorousity.

Thirdly, similar to many other modeling cases, we only studied the case where the active bots were homogeneous in order to simplify the modeling and analysis. This is a feasible approximation in practice as the majority of active bots are in the same or neighboring time zone, and the majority of the computers and their bandwidth are similar. We can relax this to a heterogeneous environment, however, it needs a lot of further effort.

Fourthly, the proposed detection method falls in the statistical category, and inherits the disadvantages of the methodology. For example, the parameters for a detection are not available for a newly created web site, thus it is impossible to carry out a detection. We believe there are effective and better methodologies to address the inherited shortcomings of the statistical techniques from different perspectives. Moreover, similar to other statistical methods, our detection method is vulnerable to deliberate traffic profile taint from attackers before their attacks.

Fifthly, attack and anti-attack is an endless loop between attackers and defenders. Whenever a new defence technique or strategy is known to attackers, they may invent new methods or strategies to circumvent the defence. For example, in this paper, we only consider that the attack botnet is homogeneous. It is difficult for defenders if different botnet owners collaborate with each other to establish a super heterogeneous botnet to carry out their attacks. This is an important and interesting topic to explore.

To help defenders win against hackers, one essential rule is to reduce the active number of bots that hackers can use. Education should be offered to Internet users, such as executing anti-virus software periodically and frequently, patching software packages in time, and turning off computers when they are not in use.

7 Summary and Future Work

In this paper, we tried to answer an important question in cybersecurity: can we detect legitimate behavior mimicking attacks? The answer is both yes and no. Firstly, it is very hard to detect this kind of attack using existing methodologies, e.g. feature based or statistics based methods. We have established a mathematical model to simulate the browsing dynamics of legitimate web browsers. Both of our theoretical analysis and real world data experiments demonstrated that we cannot detect this kind of simulation in statistics. However, there is a critical condition for a successful mimicking attack: the number of active bots of the botnet must not be lower than the number of active legitimate users. Secondly, we note that current botnet owners would find it difficult, if not impossible, to satisfy this sufficient number condition in the instance of performing large scale attacks, such as flash crowd attacks. Based on this new finding, we therefore proposed a second order statistics based discrimination algorithm to detect this kind of attack. Our theoretical analysis and simulations confirmed the effectiveness of the proposed detection method.

Our future work will follow two directions. First, there are a lot of legitimate network events that do not involve a large number of users. Therefore, botnet owners do have the capability to perform perfect mimicking attacks, such as membership recruitment, performance degradation attacks, and so on. We have a significant interest in addressing this problem by finding new methodologies. Secondly, we are also interested in tackling the problem of botnet owners who may cooperate with each other to establish a super botnet to satisfy the sufficient number condition to execute mimicking attacks.

Acknowledgment

Dr. Yu's work is partially supported by the National Natural Science Foundation of China (Grant no. 61379041), Prof. Stojmenovic's work is partially supported by NSERC Canada Discovery grant (Grant No. 41801-2010), and KAU Distinguished Scientists Program.

References

[1] T. Peng, C. Leckie, and K. Ramamohanarao, "Survey of network-based defense mechanisms countering the DOS and DDoS problems," *ACM Comput. Surv.*, vol. 39, no. 1, 2007.

[2] M. Edman and B. Yener, "On anonymity in an electronic society: A survey of anonymous communication systems," *ACM Comput. Surv.*, vol. 42, no. 1, 2009.

[3] 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," in *Proc. ACM Conf. Comput. Commun. Security*, 2009.

[4] Z. Li, A. Goyal, Y. Chen, and V. Paxson, "Towards situational awareness of large-scale botnet probing events," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 1, pp. 175–188, Mar. 2011.

[5] C. A. Shue, A. J. Kalafut, and M. Gupta, "Abnormally malicious autonomous systems and their internet connectivity," *IEEE/ACM Trans. Netw.*, vol. 20, no. 1, pp. 220–230, Feb. 2012.

[6] M. A. Rajab, J. Zarfoss, F. Monrose, and A. Terzis, "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)*, 2007.

[7] N. Jiang, J. Cao, Y. Jin, L. E. Li, and Z.-L. Zhang, "Identifying suspicious activities through DNS failure graph analysis," in *Proc. IEEE Int. Conf. Netw. Protocols*, 2010, pp. 144–153.

[8] S. Yadav, A. K. K. Reddy, A. L. N. Reddy, and S. Ranjan, "Detecting algorithmically generated malicious domain names," in *Proc. Internet Meas. Conf.*, 2010, pp. 48–61.

[9] V. L. L. Thing, M. Sloman, and N. Dulay, "A survey of bots used for distributed denial of service attacks," in *SEC*, 2007, pp. 229–240.

[10] N. Ianelli and A. Hackworth, "Botnets as vehicle for online crime," in *Proc. 18th Annu. 1st Conf.*, 2006.

[11] P. Wang, S. Sparks, and C. C. Zou, "An advanced hybrid peer-to-peer botnet," *IEEE Trans. Dependable Secure Comput.*, vol. 7, no. 2, pp. 113–127, Mar./Apr. 2010.

[12] M. Bailey, E. Cooke, F. Jahanian, Y. Xu, and M. Karir, "A survey of botnet technology and defenses," in *Proc. Cybersecurity Appl. Technol. Conf. Homeland Security*, 2009.

[13] S. Yu, W. Zhou, and R. Doss, "Information theory based detection against network behavior mimicking DDoS attack," *IEEE Commun. Lett.*, vol. 12, no. 4, pp. 319–321, Apr. 2008.

[14] 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 Comput.*, vol. 4, no. 1, pp. 56–70, Jan./Mar. 2007.

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

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

[17] G. Carl, G. Kesidis, R. Brooks, and S. Rai, "Denial-of-service attack-detection techniques," *IEEE Internet Comput.*, vol. 10, no. 1, pp. 82–89, Jan./Feb. 2006.

[18] Y. Chen and K. Hwang, "Collaborative detection and filtering of shrew DDoS attacks using spectral analysis," *J. Parallel Distrib. Comput.*, vol. 66, no. 9, pp. 1137–1151, 2006.

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

[20] Y. Xie and S.-Z. Yu, "A large-scale hidden semi-Markov model for anomaly detection on user browsing behaviors," *IEEE/ACM Trans. Netw.*, vol. 17, no. 1, pp. 54–65, Feb. 2009.

[21] M. A. Awad and I. Khalil, "Prediction of user's web-browsing behavior: Application of Markov model," *IEEE Trans. Syst. Man Cybern. B*, vol. 42, no. 4, pp. 1131–1142, Feb. 2012.

[22] G. Oikonomou and J. Mirkovic, "Modeling human behavior for defense against flash-crowd attacks," in *Proc. IEEE Conf. Comput. Commun.*, 2009.

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

[24] Z. K. Silagadze, "Citations and the Zipf-Mandelbrot's law," *Complex Syst.*, vol. 11, p. 487, 1997.

[25] M. E. Crovella and A. Bestavros, "Self-similarity in world wide web traffic: Evidence and possible causes," *IEEE/ACM Trans. Netw.*, vol. 5, no. 6, pp. 835–846, 1997.

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

[27] P. Borgnat, G. Dewaele, K. Fukuda, P. Abry, and K. Cho, "Seven years and one day: Sketching the evolution of internet traffic," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2009, pp. 711–719.

[28] M. Fomenkov, K. Keys, D. Moore, and K. Claffy, "Longitudinal study of internet traffic in 1998-2003," in *Proc. Int. Symp. Inf. Commun. Technol. (WISICT)*, 2004.

[29] WorldCup98 [Online]. Available: http://ita.ee.lbl.gov/html/contrib/WorldCup.html

[30] NLANR. *Passive measurement and analysis (PMA) project auckland-viii* [Online]. Available: http://pma.nlanr.net/Special/auck8.html

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

[32] Z. Duan, X. Yuan, and J. Chandrashekar, "Controlling IP spoofing through interdomain packet filters," *IEEE Trans. Dependable Secure Comput.*, vol. 5, no. 1, pp. 22–36, Jan./Mar. 2008.

[33] H. Wang, C. Jin, and K. G. Shin, "Defense against spoofed IP traffic using hop-count filtering," *IEEE/ACM Trans. Netw.*, vol. 15, no. 1, pp. 40–53, Feb. 2007.

[34] Y. Kim, W. C. Lau, M. C. Chuah, and H. J. Chao, "Packetscore: A statistics-based packet filtering scheme against distributed denial-of-service attacks," *IEEE Trans. Dependable Secure Comput.*, vol. 3, no. 2, pp. 141–155, Apr./Jun. 2006.

[35] C. Y. Cho, J. Caballero, C. Grier, V. Paxson, and D. Song, "Insights from the inside: A view of botnet management from infiltration," in *Proc. USENIX Workshop Large-Scale Exploits Emergent Threats (LEET)*, 2010.

[36] C. V. Zhou, C. Leckie, and S. Karunasekera, "Collaborative detection of fast flux phishing domains," *J. Netw.*, vol. 4, no. 1, pp. 75–84, 2009.

[37] S. Yu, S. Zhou, and S. Wang, "Fast-flux attack network identification based on agent lifespan," in *Proc. IEEE Int. Conf. Wireless Commun. Netw. Inf. Security (WCNIS)*, Jun. 2010, pp. 658–662.

[38] D. Dagon, C. Zou, and W. Lee, "Modeling botnet propagation using time zones," in *Proc. 13th Netw. Distrib. Syst. Security Symp. (NDSS)*, 2006.

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

[40] M. Hefeeda and O. Saleh, "Traffic modeling and proportional partial caching for peer-to-peer systems," *IEEE/ACM Trans. Netw.*, vol. 16, no. 6, pp. 1447–1460, Dec. 2008.

[41] M. Mitzenmacher, "A brief history of generative models for power law and lognormal distributions," *Internet Math.*, vol. 1, 2004.

[42] W. J. Reed and M. Jorgensen, "The double pareto-lognormal distribution—A new parametric model for size distributions," *Commun. Stat. Theory Methods*, vol. 33, no. 8, pp. 1733–1753, 2003.

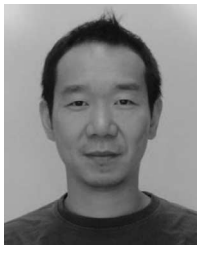
[43] B. A. Huberman, P. L. T. Pirolli, J. E. Pitkow, and R. M. Lukose, "Strong regularities in world wide web surfing," *Science*, vol. 280, no. 3, 1998.

[44] W. Liu, P. P. Pokharel, and J. C. Principe, "Correntropy: Properties and applications in non-Gaussian signal processing," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5286–5298, Nov. 2007.

[45] R. He, W.-S. Zheng, and B.-G. Hu, "Maximum correntropy criterion for robust face recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1561–1576, Aug. 2011.

[46] S. Burklen, P. J. Marron, S. Fritsch, and K. Rothermel, "User centric walk: An integrated approach for modeling the browsing behavior of users on the web," in *Proc. 38th Annu. Symp. Simul. (ANSS'05)*, 2005 pp. 149–159.
[47] Y. Xiang, W. Zhou, and M. Guo, "Flexible deterministic packet marking: An IP traceback system to find the real source of attacks," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 4, pp. 567–580, June 2009.
[48] S. Yu, W. Zhou, R. Doss, and W. Jia, "Traceback of DDoS attacks using entropy variations," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 3, pp. 412–425, Mar. 2011.
[49] S. Yu, W. Zhou, S. Guo, and M. Guo, "A dynamical deterministic packet marking scheme for DDoS traceback," in *Proc. IEEE Global Telecommun. Conf. (Globecom)*, 2013.

Shui Yu (M'05–SM'12) received the BEng and MEng degrees from the University of Electronic Science and Technology of China, Chengdu, China, in 1993 and 1999, respectively, and the PhD degree from Deakin University, Victoria, Australia, in 2004. He is currently a senior lecturer with the School of Information Technology, Deakin University. He has published nearly 100 peer review papers, including top journals and top conferences, such as IEEE TPDS, IEEE TIFS, IEEE TFS, IEEE TMC, and IEEE INFOCOM. His research interests include networking theory, network security, and mathematical modeling. He actively serves his research communities in various roles, which include the editorial boards of *IEEE Transactions on Parallel and Distributed Systems*, and three other International journals, IEEE INFOCOM TPC members, symposium co-chairs of IEEE ICC 2014, IEEE ICNC 2013 and 2104, and many different roles of international conference organizing committees. He is a member of AAAS.



Song Guo (M'02–SM'11) received the PhD degree in computer science from the University of Ottawa, Canada in 2005. He is currently a senior associate professor at School of Computer Science and Engineering, the University of Aizu, Japan. His research interests include the areas of protocol design and performance analysis for reliable, energy-efficient, and cost-effective communications in wireless networks. He is an associate editor of the *IEEE Transactions on Parallel and Distributed Systems* and an editor of *Wireless Communications and Mobile Computing*. He is a senior member of the ACM.

Ivan Stojmenovic received the PhD degree in mathematics. He is full professor at the University of Ottawa, Canada. He held regular and visiting positions in Serbia, Japan, USA, Canada, France, Mexico, Spain, U.K. (as chair in applied computing at the University of Birmingham), Hong Kong, Brazil, Taiwan, China and Australia. He published over 300 different papers, and edited 7 books on wireless, ad hoc, sensor and actuator networks, and applied algorithms with Wiley. He was the editor-in-chief of *IEEE Transactions on Parallel and Distributed Systems* (2010–2013), and founder and editor-in-chief of three journals. He is an associate editor-in-chief of *Tsinghua Journal of Science and Technology*, steering committee member of *IEEE Transactions on Emergent Topics in Computing*, and associate editor of *IEEE Network*, *IEEE Transactions on Cloud Computing*, *IEEE Transactions on Computers*, *ACM Wireless Networks*, and some other journals. He is on Thomson Reuters list of Highly Cited Researchers (from 2013; <300 computer scientist), has h-index 59, top h-index in Canada for mathematics and statistics, and $>13\,000$ citations. He received five best paper awards and the Fast Breaking Paper for October 2003, by Thomson ISI ESI. He received the Royal Society Research Merit Award, UK (2006) and Humboldt Research Award, Germany (2012). He is Tsinghua 1000 Plan Distinguished Professor (2012–2015). He is Fellow of the IEEE (Communications Society, class 2008), and Canadian Academy of Engineering (since 2012), and Member of the Academia Europaea (The Academy of Europe), from 2012 (section: Informatics). He was an IEEE CS Distinguished Visitor 2010–2011 and received 2012 Distinguished Service award from IEEE ComSoc Communications Software TC. He received Excellence in Research Award of the University of Ottawa, 2009. He chaired and/or organized >60 workshops and conferences, and served in >200 program committees. He was a program co-chair at IEEE PIMRC 2008, IEEE AINA-07, IEEE MASS-04&07, founded several workshop series, and is/was Workshop Chair at IEEE ICDCS 2013, IEEE INFOCOM 2011, IEEE MASS-09, and ACM Mobihoc-07&08.