Soft Comput (2017) 21:3713–3721
DOI 10.1007/s00500-015-2025-6

METHODOLOGIES AND APPLICATION

A protocol-free detection against cloud oriented reflection DoS attacks

Le Xiao[1,2] · **Wei Wei**[2] · **Weidong Yang**[2] · **Yulong Shen**[3] · **Xianglin Wu**[1]

Published online: 18 January 2016

Abstract Distributed denial of service (DDoS) attack presents a critical threat to cloud infrastructure, where many manipulated hosts flood the victim cloud with plenty of packets, which will lead to the exhaustion of bandwidth and other system resources. As one type of DDoS attack, in reflection DoS (RDoS) attack, legitimate servers (reflectors) are fooled into sending a large number of packets to the victim cloud. Most of the existed RDoS attack detection mechanisms are protocol-specific, thus low in efficiency. It is inspected that because of being triggered by the same attacking flow, intra-unite correlation exists among the packet rate of attacking flows. Based on the phenomenon, a flow correlation coefficient (FCC)-based protocol-free detection (PFD) algorithm is proposed. The simulation results show that PFD can detect attacking flows efficiently and effectively and is not protocol-specific, thus can be used as effective supplement to existed algorithms.

Keywords Cloud computing · Reflection DoS · Flash crowds · Flow correlation coefficient · Protocol-free detection · Botnets

Communicated by V. Loia.

✉ Le Xiao
lorexiao@163.com

[1] School of Automation, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China

[2] College of Information Science and Engineering, Henan University of Technology, Zhengzhou, Henan 450001, China

[3] School of Computer Science and Technology, Xidian University, Xi'an, Shaanxi 710071, China

1 Introduction

The cloud computing technology provides a solid foundation for cloud based application. A cloud platform makes resource provisioning elastic, reliable and cost-effective, thus significantly alleviates the effort of building dynamic and scalable network services (Wei et al. 2014a, b; Liu and Wei 2015). However, even with huge amount of resources, cloud is still vulnerable to many kinds of attacks. As one of them, denial of service (DoS) attack constitutes a critical threat to cloud infrastructure. DoS attack is the kind of attack to make a service unavailable to target users, e.g. make a server disconnected from Internet, or make a network link congested by a large number of packets. Distributed denial of service (DDoS) is the kind of attack where attacking packets come from thousands of computers. Usually, there are many paralleled attacking flows be used to exhaust the resource in targeted server or network. To initialize the attack, a lot of computers are compromised to be used as attacking source. In one occurrence of attack, computers in a botnet receive attacking order from remote attacker and send attacking packets to victim. If the target is a network, the incoming bandwidth will be exhausted and future incoming packets will be dropped, and if the target is a server, the computing/memory resource of the server will be exhausted by the network stack module in the server.

The computers involved in DDoS are often organized as botnet. Botnet is usually constructed using malware, the key step of installing malware on target computer is breaking into systems. E.g., by scanning target computer's ports, the attacker can break into the target computer by exploiting the vulnerability of the software listening on the given port, then install malware to help take further action.

One well-known malware is MyDoom. Before the launch of one attack, a new version of malware is produced by

including target information and attacking date in the source code. Another kind of malware is trojan, which allow the attacker to manipulate the compromised computer as his will, e.g., the attacker can order the compromised computer to download another malware agent software. In addition to previous ones, Stacheldraht can be viewed as a classic tool to organize botnet. In Stacheldraht, all compromised computers are organized to form a layered structure. The attacker connect to handlers, and handlers connect to compromised computers. The intrusion to compromised computers is implemented via the handlers. The number of computers each handler controls can be up to more than one thousand.

After botnet is ready, many kinds of attacks can be launched, e.g., resource starvation attacks like SYN flood, DNS targeted attack like DNS base DoS, or bandwidth consumption attacks like IP spoofing or smurf. In SYN flood, a small number of compromised computers send a lot of TCP SYN packets with a large number of faked source IP address, when these packets arrived at victim server, it will consume memory and computer resource, for the victim server to sustain the state information for the connection linked with each TCP SYN packet. When further TCP packet will not arrive for these links, a lot of system resources are wasted. Common countermeasure for SYN flood is SYN cookie, but it can only defend attack not saturating incoming bandwidth. Furthermore, computers can be compromised intentionally by their owners, just trading for economic returns. If only one computer is involved to launch attack, the attack can be classified as DoS, or if more than one computer is involved, the attack can be classified as DDoS.

The inherent power of DDoS is the huge difference between the resource owned by the attacking side and the victim side, i.e., the number of attacking computers is significantly larger than the number of victim. And since attacking computers are distributed sparsely in the Internet, it is hard to shut down the attacking computers one by one. Actually, the victim can only wait for the stop of attack. Currently, there is still no effective defense mechanism for DDoS, e.g., increasing the bandwidth of incoming link or adding more resources in victim will not help since it is easy for attacker to add more attacking computer from botnet.

Moreover, DDoS can be launched by one attacking computer, which send fake requests of some protocols to a large number of innocent computers, with the source address of these faked packets point to victim. Depending on the request–response mechanism in the given protocol, these innocent computers will reply to the requests by sending response packet to the source address embedded in the faked request packet, consequently flood the victim. This kind of attack is called distributed reflection DoS (DRDoS). DRDoS is more difficult to detect. Due to the inherent mechanism of RDoS, it can exploit any connectionless request–response-based protocols. Some common forms of DRDoS include Smurf, in which the attacker sends ECHO requests to the broadcast address of the network, each computer in the network will receive the request and flood ECHO REPLY packets to victim. A lot of known services can be exploited by DRDoS. Because the volume of reflected traffic is low in source network, the detection near single host maybe useless (Paxson 2001). Since ingress filtering is not largely deployed, it is also not a hopeful solution (Ferguson 2000).

This paper provide ideas of solving the problem. We inspect the basic traffic correlation near the victim cloud under RDoS attack, and present a universal detection method: the protocol-free detection against cloud oriented reflection DoS attacks (PFD). PFD is protocol-free and its computation cost will not be affected by network throughput. In PFD, packet rate is sampled in upstream router and correlation of flows are tested using flow correlation coefficient (FCC), the detection result is given by considering current FCC value and historical information. As far as we know, it is the first time that RDoS attack be analyzed and detected by Flow Correlation Coefficient.

2 Related work

There are many literatures for the topic of cloud based DoS attacks. One kind of special cloud-related attack is energy-oriented DoS attack, which is analyzed under different scenarios with different constraints (Palmieri et al. 2011, 2014a, 2015; Ficco and Palmieri 2015). Palmieri et al first investigated the impacts of network-based DoS attacks under the energy consumption perspective (Palmieri et al. 2011). Impacts of different DoS attacks are analyzed and several aspects are revealed to show the inherent affects of DoS attacks. They then introduced a orchestrated strategy achieve maximal cost-effectiveness ratio (Palmieri et al. 2014a). Palmieri et al analyzed different types of energy-oriented DoS attack and modeled their behavior (Palmieri et al. 2015). The introduced model quantified how attack can manipulate network traffic to raise the target facility emissions and costs. They focused on attacks that are specifically tailored to originate the worst-case energy demands by leveraging properly crafted low-rate traffic patterns to ensure stealth operations. Special kind of DoS attacks, e.g., the low-rate DoS, can cause worst-case energy demand. They identified the strategies attackers use to increase the overall energy consumption in a fraudulent way, and analyze their impact within large-scale cloud infrastructures (Ficco and Palmieri 2015).

For detection of DoS, many packet-level defense methods already exist in traditional and emerging cloud platform. Blocking all incoming responsive packets is not applicable, since as a result, the protected server can not actively connect to any remote servers (Paxson 2001). It is also computation-

ally expensive to examine packet content and record protocol status, and the detection itself is also vulnerable to attacks (CHKP 2010; Rooj 2011; Tsunoda et al. 2008). And more application-level information is utilized for detection, such as the user browsing dynamics based detection (Xie and Yu 2009a, b; Kandula et al. 2005). With more new protocols can be utilized to launch RDoS attack (Drakos 2002), the list of protocols need to be considered with each protocol tackled specifically for detecting RDoS attack, and the length of list will grow rapidly. Moreover, reflectors can be controlled to send traffic mimicking flash crowds, which are legitimate traffic in the form of dramatic surges of access to a server, and are inclined to be detected as attacking traffic due to similarity with attacking traffic (Jung et al. 2002; Scherrer et al. 2007). Flash crowds were differentiated from attacking traffic by inspecting user browsing dynamics (Xie and Yu 2009a, b) and human behavior (Oikonomou and Mirkovic 2009; Yu et al. 2013), but these methods are inherently protocol-specific. As a result, we urgently expect to discover protocol-free methods to enable universal detection of RDoS attack, and differentiate flash crowds from attacking traffic. There are already some preliminary works. Palmieri et al developed a two-stage anomaly detection strategy based on multiple distributed sensors located throughout the network (Palmieri et al. 2014b). By solving a Blind Source Separation problem, fundamental traffic components are extracted from network traffic. Using the baseline traffic profile built from these components, detection is transformed into an anomalous/normal classification problem and solved by machine learning-inferred decision trees. Wei et al found linear correlation existed among attacking flows, they use a rank correlation-based detection algorithm to locate and filter incoming attacking packets (Wei et al. 2013). Since correlation can be classified as one kind of distance measurement, to further investigate correlation-based method, we list several kinds of related distance measurement as follows:

2.1 Hellinger distance

(1) Definition based on measure theory

Based on a probability measure χ, we define two continuous probability measures A and B, with the Hellinger distance between A and B is defined as below.

$$H(A, B) = \sqrt{\frac{1}{2}\int\left(\sqrt{\frac{\mathrm{d}A}{\mathrm{d}\chi}} - \sqrt{\frac{\mathrm{d}B}{\mathrm{d}\chi}}\right)^2 \mathrm{d}\chi}. \tag{1}$$

We can see that in above definition, if χ is replaced with another probability measure, the Hellinger distance between A and B will not change. As a result, in most circumstances, the definition can be rewritten as

$$H(A, B) = \sqrt{\frac{1}{2}\int\left(\sqrt{\mathrm{d}A} - \sqrt{\mathrm{d}B}\right)^2}. \tag{2}$$

(2) Definition based on elementary probability theory

If we deduce definition from elementary probability theory, χ is assumed to be a Lebesgue measure, the $\mathrm{d}A/\mathrm{d}\chi$ and $\mathrm{d}B/\mathrm{d}\chi$ can be viewed as probability density functions, which can be rewritten as $f^a(\cdot)$ and $f^b(\cdot)$. Accordingly, since integral of a probability density is 1, the definition of Hellinger distance can be reformed as a standard calculus integral

$$\begin{aligned} H(A, B) &= \sqrt{\frac{1}{2}\int\left(\sqrt{f^a(x)} - \sqrt{f^b(x)}\right)^2 \mathrm{d}x} \\ &= \sqrt{1 - \int\sqrt{f^a(x) f^b(x)}\,\mathrm{d}x} \end{aligned} \tag{3}$$

Based on Cauchy Schwarz inequality, the value of $H(A, B)$ follows between the range [0,1]:

$$0 \le H(A, B) \le 1. \tag{4}$$

(3) Definition based on discrete distributions

Set $A = (a_1, \ldots, a_k)$ and $B = (b_1, \ldots, b_k)$ as two discrete probability distributions, the definition of Hellinger distance can be formed as

$$H(A, B) = \frac{1}{\sqrt{2}}\sqrt{\sum_{i=1}^{k}(\sqrt{a_i} - \sqrt{b_i})^2}, \tag{5}$$

which can be rewritten as:

$$H(A, B) = \frac{1}{\sqrt{2}}\left\|\sqrt{A} - \sqrt{B}\right\|_2. \tag{6}$$

2.2 Kullback-Leibler distance

Kullback-Leibler distance is also known as relative entropy (also known as KL divergence, KLIC, information divergence, information gain), which is just a special case of a broader class of divergences called f-divergences, and can be derived from a Bregman divergence.

For two probability distribution A and B, when B is used to approximate A, the Kullback-Leibler distance of B from A can be used to measure information lost caused by the approximation. One important feature is that Kullback-Leibler distance is non-symmetric, which means that $\mathrm{D}_{\mathrm{KL}}(A\|B) \neq \mathrm{D}_{\mathrm{KL}}(BA)$. Here $\mathrm{D}_{\mathrm{KL}}(A\|B)$ is used to represent Kullback-Leibler between A and B. Fortunately, Fisher information metric is symmetric, which is the infinitesimal form of Kullback-Leibler distance. Suppose in a common scenario that we use a theoretical distribution B to approximate the real distribution A of observed data, the Kullback-Leibler

distance represented the extra bits needed to precisely reconstruct samples of A from B.

The definition of Kullback-Leibler distance of B from A is given by:

$$D_{\mathrm{KL}}(A\|B) = \sum_i A(i)\,\ln\frac{A(i)}{B(i)}. \tag{7}$$

As shown in equation above, the Kullback-Leibler distance can be viewed as the expected logarithmic difference between A and B, and if $A(i)$ is zero, the i-th part of Kullback-Leibler distance is zero. Note that $A(i) = 0$ is true only if $B(i) = 0$.

If A and B are distribution of continuous random variable, the Kullback-Leibler distance can be defined as the integration of functions of A and B's PDF (probability density function) (i.e., $p^a(x)$ and $p^b(x)$):

$$D_{\mathrm{KL}}(A\|B) = \int_{-\infty}^{\infty} p^a(x)\,\ln\frac{p^a(x)}{p^b(x)}\,\mathrm{d}x, \tag{8}$$

In general, if for a set S, we get its probability measures A and B, with A is continuous with respect to B, then the Kullback-Leibler distance can be defined as:

$$D_{\mathrm{KL}}(A\|B) = \int_S \ln\frac{\mathrm{d}A}{\mathrm{d}B}\,\mathrm{d}A, \tag{9}$$

where $\frac{\mathrm{d}A}{\mathrm{d}B}$ is the Radon-Nikodym derivative, and above definition can be rewritten as follows:

$$D_{\mathrm{KL}}(A\|B) = \int_S \ln\left(\frac{\mathrm{d}A}{\mathrm{d}B}\right)\frac{\mathrm{d}A}{\mathrm{d}B}\,\mathrm{d}B, \tag{10}$$

where above definition can be views as the entropy of A relative to B. Consequently, if there is a measurement ν on S, and we define $a = \frac{\mathrm{d}A}{\mathrm{d}\nu}$ and $b = \frac{\mathrm{d}B}{\mathrm{d}\nu}$, and if p and q exists, the Kullback-Leibler distance can be finally defined as:

$$D_{\mathrm{KL}}(A\|B) = \int_S a\,\ln\frac{a}{b}\,\mathrm{d}\nu. \tag{11}$$

If we want to measure information in units of bits, we can use a logarithms of 2. Similarly, if we want to measure in nats, we can use logarithms e.

3 System model

For length limits, two typical scenarios are considered here, which involves many reflectors and one attacker:

(a) Constant rate attack (CA): the attacker send faked requests to reflectors randomly, following uniform distribution, at a constant rate, e.g., the outgoing bandwidth.

(b) Variable rate attack (VA): the attacker send faked requests to reflectors randomly, following uniform distribution, at a varying and low rate.

3.1 Definition of network flow

For a given sampling point in network (e.g., a community network), one network flow is defined as all packets with same destination IP address and the sampling period is T. Set the start time of a sampling period is t, then for a pair of flows X_a and X_b in the time span $[t, t+T]$, their corresponding set of reflectors are defined as R_a and R_b, and the corresponding numbers of reflectors are M_a and M_b. The set of other reflectors is defined as R_0 as shown in Fig. 1. where corresponding set of reflectors of one flow is all the reflectors that send responsive packets to the victim cloud.

If the length of X_a is N, then we define network flow as follows:

$$X_a = \{x_a[1];\ x_a[2];\ \ldots;\ x_a[N]\} \tag{12}$$

In Eq. 12, $x_a[k]\ (1 \le k \le N)$ is the number of packets sampled in the $k-$th time interval.

Let X_a and X_b be two network flows with the same length N, then correlation is defined as:

$$r_{X_a,X_b} = \frac{1}{N}\sum_{n=1}^{N} x_a[n]\,x_b[n] \tag{13}$$

Correlation is usually used to describe the relation of two flows. But in sometime, correlation value is zero for two correlated flows due to their phase difference. Therefore, a new definition is given by considering phase difference:

$$r_{X_a,X_b}[k] = \frac{1}{N}\sum_{n=1}^{N} x_a[n]\,x_b[n+k] \tag{14}$$

Where $k = 1, 2, 3, N-1$ is the phase shift of flow X_b. Then to erase the magnitude difference, it is necessary to make some unification, as shown in Sect. 3.2.

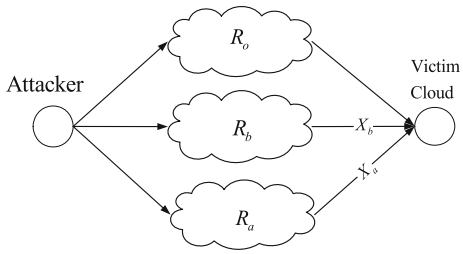

Fig. 1 Attacking scenario

3.2 Definition of flow correlation coefficient (FCC)

Then the correlation coefficient between two flows can be defined as:

$$\rho_{X_a,X_b}[k] = \frac{r_{X_a,X_b}[k]}{\frac{1}{N}\left[\sum_{n=1}^{N} x_a^2[n] \sum_{n=1}^{N} x_b^2[n]\right]^{1/2}} \tag{15}$$

Based on FCC, we can construct the protocol-specific detection algorithm, which is introduced in Sect. 4.

4 Algorithm

Ideally, for two attacking flow X_a and X_b with no background noise, the correlation coefficient $\rho_{a,b}$ should be 1. Although background noise is not avoidable in Internet, after some kind of denoising, correlation between two attacking flows should be significantly high compared with non-correlated flow pairs.

Then in RDoS attack scenario, we can use thresholds δ to judge whether both flows are malicious or not. If it exists $\rho_{a,b} > \delta$ for flow pair (a, b), it means that both flows are reflection flows and attack is detected. And then we can make the final decision R by inspecting all flow pairs.

$$R = \begin{cases} 0, & \rho_{a,b} < \delta \ \forall (a,b) \\ 1, & \rho_{a,b} > \delta \ \exists (a,b) \end{cases} \tag{16}$$

There is no universal threshold for all scenarios, a feasible way is to calculate thresholds from history data of different scenarios. We will give an example of threshold calculation in our simulations in Sect. 5.

The accuracy of PFD can be improved by using multiple flow pairs. Set the false negative rate per flow pair is q, by using multiple flow pairs, q can be decreased further, e.g., when the number of flow pairs is m then the false negative rate will be decreased towards q_m. And when $q = 0.1$ and $m = 3, q_m = 0.1\%$, which is low enough. The PFD algorithm is shown in Table 1.

The time complexity of PFD algorithm is given as follows. For i-th flow pairs, N coefficients need be calculated with each one is given by Eq. 15 using different $k(1 \le k \le N)$, where N is the length of sampling vector. Then the time complexity is $O(N^2)$ for each flow pairs, and the total time complexity will be $O(mN^2)$ if m flow pairs are tested.

In cloud environment, the detection algorithm can be deployed to the entrance of a protected virtual LAN, i.e., only paid custom can deploy the algorithm to protect themselves. To get large processing throughput, the entrance needs to be a virtual switch with all packets stored in memory. And the detection algorithm runs on a distributed computing infrastructure like MapReduce platform. By exploiting cloud flexibility, high throughput can be achieved easily with low cost.

Table 1 Steps of PFD algorithm

Input: sampling sequence of each flow

Output: a binary value indicating whether the victim cloud is under attack.

```
For (each network entry){
    Get recent average flow rate of background traffic,
which can be calculated from history data.
}
If ( alarm raised from pre-deployed detection method){
    // pre-deployed detection method can be efficient
    // burst-based detection method.
    Sample incoming flows and get packet rate of each flow;
    For ( each sampling time interval){
        Convert sampled data into value sequence for each flow;
        Subtract their value sequence by their respect recent average
flow rate;
        Enumerate each flow pairs and calculate FCC
for each pair;
        Make decision by inspecting all flow pairs;
    }
}
```

5 Analysis

With the presence of transmitting latency, attacking packets arrived at the victim cloud was generated a little earlier at source reflectors. Set latency as τ, if per unit of time T is far greater than τ, then $C_{a,t}$ and $C_{b,t}$, the number of packets at victim cloud in period $[t, t + T]$ can be approximated by the number of sent packets in reflectors in $[t - \tau, t + T - \tau]$: $C_{a,t-\tau}$ and $C_{b,t-\tau}$, which is shown as follows:

$$C_{a,t} \approx C_{a,t-\tau} \tag{17}$$

$$C_{b,t} \approx C_{b,t-\tau} \tag{18}$$

The arrived packets from the attacker will trigger reflectors to send responsive packets immediately. In most of protocols, without loss of generality, one arrived request packet will trigger one responsive packet. So in $[t - \tau, t + T - \tau]$, the number of arrived packets at reflectors are also $C_{a,t-\tau}$ and $C_{b,t-\tau}$. Then the total number of reflectors (with reflectors not included in set R_a and R_b) involved in the attack is M_r, with number of reflectors are M_a and M_b for set R_a and R_b, respectively. The total count of arrived faked request packets is $C_{r,t-\tau}$. As faked requests from the attacker are distributed randomly, there are:

$$C_{a,t-\tau} \approx \frac{M_a}{M_r} C_{r,t-\tau} \tag{19}$$

$$C_{b,t-\tau} \approx \frac{M_b}{M_r} C_{r,t-\tau} \tag{20}$$

Then we have:

$$\frac{C_{a,t}}{C_{b,t}} \approx \frac{C_{a,t-\tau}}{C_{b,t-\tau}} \approx \frac{M_a}{M_b} \tag{21}$$

It is shown above that in $[t, t+T]$, for flow X_a and X_b, the ratio of number of packets is close to the number of their respective reflectors. If R_a and R_b do not change significantly between adjacent time units, M_a/M_b can sustain constant for a short period. Consequently, the packet rates for X_a and X_b is proportional.

On top of that, if faked requests are sent at full speed, $C_{r,t-\tau}$ will be limited by the bandwidth of the attacker, then:

$$C_{a,t} + C_{b,t} \approx C_{a,t-\tau} + C_{b,t-\tau} \approx \frac{M_a + M_b}{M_r} C_{r,t-\tau} \tag{22}$$

So, summation of packet rates for X_a and X_b approximates a constant. In the two typical scenarios above, the packet rate of X_a and X_b presents correlation and can be characterized by FCC.

The task of detection algorithm is to differentiate attacking and legitimate traffic, and the main challenge is to differentiate legitimate traffic resembling attacking traffic, i.e., the flash crowds. Flash crowds are legitimate, but unexpected, plenty of access to a server, such as web requests to a breaking event.

We now investigate the FCC of flash crowd flows. It is shown by previous research that http traffic follows the Pareto law (Paxson and Floyd 1995; Crovella and Bestavros 1997), hence, we can judge that whether the abnormal traffic result from flash crowds or not through observing that whether the traffic conforms to the Pareto distribution.

Its distribution is defined as follows. With a random variable Z, and minimal time interval β for arrived packets. For a time interval z, the probability density function (PDF) of the Pareto distribution is defined as:

$$P_r[Z = z] = \alpha \cdot \beta^{\alpha} \cdot z^{-(\alpha+1)} \tag{23}$$

where $\beta < z$ and α is the Pareto index. Then it has Theorem 1:

Theorem 1 *Given two same length instances, X_a and X_b ($a \neq b$) with same length N, of a flash crowd produced by the same function with same parameters, $\lim_{N\to\infty} \rho_{X_i,X_j}[k] = 0$.*

The proof of Theorems in the paper can be found in appendix. It can be easily deduced from Theorem 1 that:

Corollary 1 *For two independent flash crowds X_a and X_b ($a \neq b$) with same length N, $\forall\delta\,(\delta < 1)$, $\exists N'$ when $N > N'$, $\rho_{X_i,X_j}[k] < \delta$.*

It is shown by Corollary 1 that the FCC can be low enough with large enough N. Then we will investigate the FCC of attacking traffic in RDoS.

Theorem 2 *If there is no background noise and network delay, let X_a and X_b be two traffic flows in VA and CA scenarios, there is $\rho_{X_i,X_j}[k] = 1$.*

Based on Theorem 2, it is probably that we can differentiate RDoS attack flows from flash crowds, as the FCC of these two kind of traffic are different in perfect condition. However, the background noise may affect FCC. The strategy in Algorithm 1 is to subtract value sequence by the recent average rate of the flow, whose effectiveness is stated as follows.

Corollary 2 *Let Y_a and Y_b be the noises for two attacking flows X_a and X_b with sufficiently large length N $\forall\delta(\delta < 1)$, $\exists\Delta$, $\rho_{X_i,X_j}[k] \geq \delta$ holds when $\frac{E(X_a)}{E(Y_a)} > \Delta$ and $\frac{E(X_b)}{E(Y_b)} > \Delta$.*

Combining Theorems and corollaries above, we can come to the conclusion in Theorem 3.

Theorem 3 *if the length of the sampled flow is large enough, and the attack strength is strong enough, then the reflection DoS attack flow can be discriminated from flash crowds by the flow correlation under two conditions.*

With Theorem 3, if we remove background traffic to strengthen attack traffic and get sample flow with long enough length, the FCC between attack flows will be different enough with other flow pairs, thus can be a useful indicator for detection.

6 Experiments

As shown in Fig. 2, the typical network used in our NS2 simulation topology includes 15 routers and 800 hosts with

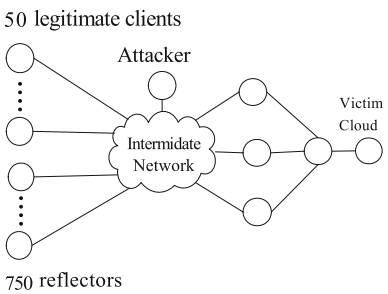

Fig. 2 Simulation topology

750 hosts are reflectors and 50 hosts are innocent ones. Normal requests to victim cloud follow a Pareto distribution. The latency in network is between 10 and 200 ms, which is consistent to the average Internet RTT of 200 ms.

We test the two typical scenarios mentioned in the first paragraph of Sect. 3. To figure out the effectiveness of PFD, we compare PFD with relative entropy method (Yu et al. 2008), which detect attacks by inspecting relative entropy between flow pairs. In relative entropy based method, if intrusion detection system (IDS) found existence of attack and issued an alarm, it can sample suspicious flows f_a and f_b for a sufficient long time span T. After accumulating enough data, IDS can calculate distributions for these two flows:

$$\begin{aligned} A(X) &= a(x_1, x_2, x_3 \ldots x_n) \\ B(X) &= b(x_1, x_2, x_3 \ldots x_n) \end{aligned} \tag{24}$$

If attackers trigger botnet to use the same function $f(\cdot)$ to generate attacking flows f_a and f_b. And suppose the transformation of flows from reflectors to IDS can be views as some kind of system functions, and if we define the system functions for f_a and f_b as $g_a(\cdot)$ and $g_b(\cdot)$, respectively. Then we can have:

$$\begin{aligned} a(x) &= g_a(f(x)) \\ b(x) &= g_b(f(x)) \end{aligned} \tag{25}$$

In ideal situation, if g_a and g_b are linear, the relative entropy between f_a and f_b is zero, i.e., $\mathrm{D}(A \parallel B) = 0$. However, the assumption of linearity may not holds in Internet due to the existence of background traffic. The relative entropy can be small enough compared to other unrelated flow pairs. Still, the existence of attack can be judged using below equation:

$$\mathrm{Result}(A, B) = \begin{cases} 1 & D(A \parallel B) \leq \delta \\ 0 & D(A \parallel B) > \delta \end{cases} \tag{26}$$

We also present steps of relative entropy method for comparison:

1. IDS locate suspicious attacking flows in IDS.
2. Sample flow rate of suspicious flow using time unit t.
3. Collect data for a sufficient time span T, and get sampling data $x_1, x_1 \ldots x_n$, extract distinct values and their frequencies. Then calculate the approximated distribution of flows:

$$p(x_i) = \frac{f_i}{\sum_{i=1}^{n} f_i}, \quad i = 1, 2 \ldots n. \tag{27}$$

4. Use the distribution information to calculate relative entropy.

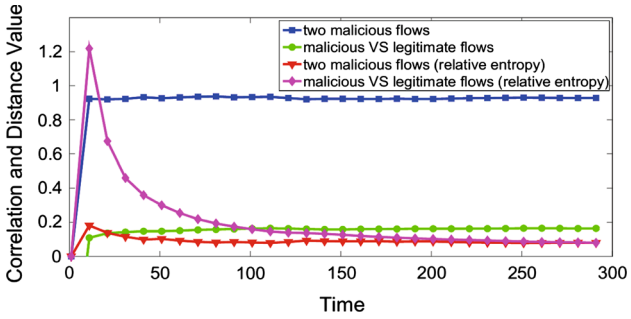

Fig. 3 Comparison of methods for constant rate attack

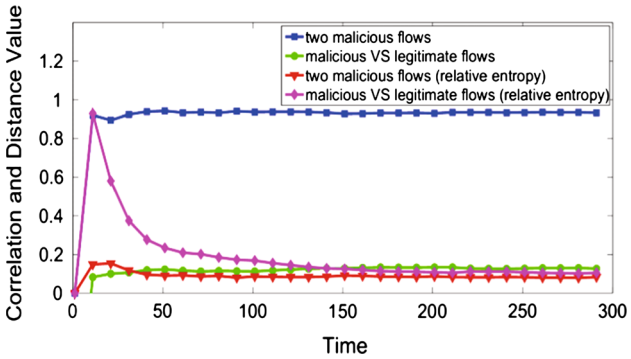

Fig. 4 Comparison of methods for variable rate attack

5. For different flows pairs, make the decision based on Eq. 26.
6. IDS can response to attack by blocking packets in attacking flows.

The comparison result is shown in Figs. 3 and 4, and it is shown that:

1. In PFD, high correlation between two attacking flows is clearly captured, even with a high rate of background traffic, and the correlation is low between one attacking and one normal flow (the solid circle in Figs. 3 and 4).
2. In relative entropy method, the relative entropy of two malicious flows (the solid triangle in Figs. 3 and 4) is similar to the relative entropy of one malicious and one legitimate flow (the solid diamond in Figs. 3 and 4). Thus RDoS cannot be easily detected by relative entropy.
3. In PFD, FCC becomes stable after the sampling has lasted for about 100 units of time (the solid rectangle in Figs. 3 and 4). When the unit of time is 0.1 s, we need 10 s to give the final detection result.

It is shown from above experiments that in the typical RDoS scenario, the correlation between attacking flows can only

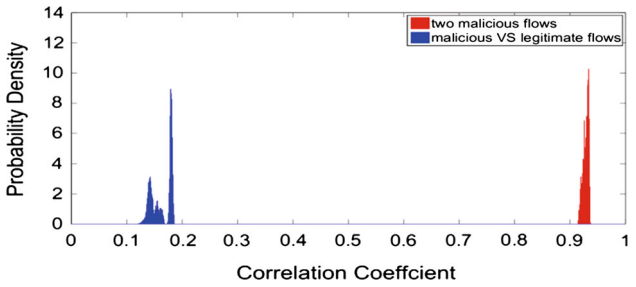

Fig. 5 PDF of correlation coefficient in two typical attacking scenarios

be captured by PFD, and the detection delay is small enough (10 s), thus PFD can be used as a helpful indicator.

We use 200 different RDoS cases to get the threshold for our scenarios, where packet rate of attacking flow is 100–1000 % of that of legitimate flows, which covers a broad range of cases with high and low rates. Figure 5 shows the PDF of correlation coefficient in PFD, it is found that:

1. PFD can clearly distinguish between the two kinds of correlations with a broad range of different packet rate.
2. The threshold is chosen using the x-axis of cross point of fitted curves. As in Fig. 5, we choose the corresponding threshold $\delta = 0.5$.

7 Conclusion

The paper concentrates on detecting RDoS attack against cloud without considering the protocol used in attack, and proposes the Protocol-Free Detection (PFD) algorithm. Once suspicious flows are located, PFD calculates the Flow Correlation coefficient (FCC) between flow pairs and issue a final warning. The simulation results show that it is a helpful indicator for RDoS attack detection. It can also help us to find and isolate attacking flows.

There are a lot of interesting works to do in the future, including:

1. More simulations and experiments against real RDoS attack in Internet. Due to the scarcity of public RDoS data, we need to investigate how to collect real traffic in some network test bed.
2. Test our algorithm in more sophisticated scenarios, e.g., with more sophisticated topology by modeling real networks.
3. The method attackers can use to escape detection and our countermeasures, i.e., the attackers try to decrease the FCC of their flows, and what we can do to detect this kind of Byzantine attack.
4. Investigating how to use PFD algorithm to protect application oriented cloud, e.g., the cloud based grain storage information system.

Acknowledgements This study was funded by National Natural Science Foundation of China (Grant Number 61202099, number U1504607), Plan of Nature Science Fundamental Research in Henan University of Technology (Grant Number 2014JCYJ04).

Compliance with ethical standards

Conflict of interest The authors declare that there is no conflict of interests regarding the publication of this paper.

Ethical approval This article does not contain any studies with human participants or animals performed by any of the authors.

8 Appendix

Proof of Theorem 1 X_a and X_b follow the Pareto distribution if they are flash crowds. The probability of $x_a[n] = x_b[n] = x$ is:

$$Pr([x_a[n] = x_b[n] = x) = \left(\frac{\alpha \cdot \beta^{\alpha}}{x^{\alpha+1}}\right)^2 < 1$$

If $X_a = X_b$, i.e., $x_a[n] = x_b[n]$ for each $n(1 \le n \le N)$, then we have:

$$Pr(X_a = X_b) = (Pr([x_a[n] = x_b[n] = x))^N = \left(\frac{\alpha \cdot \beta^{\alpha}}{x^{\alpha+1}}\right)^{2N}$$

It can be concluded that:

$$\lim_{N\to\infty} \rho_{X_i,X_j}[k] = \lim_{N\to\infty} Pr(X_a = X_b) = 0$$

□

Proof of Theorem 2 With no background noise and network delay, there is $x_a[n] = kx_b[n](1 \le n \le N)$, where $k = M_a/M_b$ in Eqs. 8 and 9. Then we have: $\rho_{X_a,X_b}[k]$

$$= \frac{\frac{1}{N}\sum_{n=1}^{N} x_a[n]x_b[n]}{\frac{1}{N}\left[\sum_{n=1}^{N} x_a^2[n] \sum_{n=1}^{N} x_b^2[n]\right]^{1/2}}$$

$$= \frac{\sum_{n=1}^{N} kx_b^2[n]}{\left[\sum_{n=1}^{N} k^2 x_b^2[n] \sum_{n=1}^{N} x_b^2[n]\right]^{1/2}}$$

$$= \frac{k\sum_{n=1}^{N} x_b^2[n]}{\left[(k\sum_{n=1}^{N} x_b^2[n])^2\right]^{1/2}} = 1$$

□

Proof of Theorem.3 Let X_a and X_b be two random flash crowds, X_c and X_d be two RDoS flooding attack flows, and Δ be a very small real number. Based on Theorem 1, for a given$N$, it has:

$$Pr(\rho_{X_a,X_b}[k] < \Delta|N) = 1$$

Based on Theorem 2, given N and signal-noise-rate (SNR), the following equation holds. Here SNR is the ratio of attacking traffic rate to background traffic rate.

$$Pr(\rho_{X_c,X_d}[k] \geq \Delta|N, SNR) = 1$$

Since $\rho_{X_a,X_b}[k]$ is decreasing along with increasing of N(the length of flow). In perfect condition, $\rho_{X_c,X_d}[k] = 1$ and $\rho_{X_a,X_b}[k]$ decreases with increasing of SNR. As a result, there must exist a point where both above two equations hold, i.e., $\rho_{X_a,X_b}[k] < \Delta \leq \rho_{X_c,X_d}[k]$, thus reflection DoS attacking flow can be isolated from flash crowds, and Theorem holds as well. □

References

CHKP (2010) Stateful inspection technology (the industry standard for enterprise class network security solutions). http://www.checkpoint.com/products/downloads/Stateful_Inspection

Crovella M, Bestavros A (1997) Self-similarity in world wide web traffic: Evidence and possible causes. IEEE/ACM Trans Netw. 5(6):835–846

Drakos RN (2002) Application-level reflection attacks. http://www.lemuria.org/security/application-drdos.html

Ferguson P (2000) rfc2827:network ingress filtering: defeating denial of service attacks which employ ip source address spoofing

Ficco M, Palmieri F (2015) Introducing fraudulent energy consumption in cloud infrastructures: a new generation of denial-of-service attacks. IEEE Syst J 99:1–11

Jung J, Krishnamurthy B, Rabinovich M (2002) Flash crowds and denial of service attacks: characterization and implications for cdns and web sites. In: Proc. 11th Intl Conf. World Wide Web (WWW), pp 252–262

Kandula S, Katabi D, Jacob M, Berger A (2005) Botz-4-sale: surviving organized ddos attacks that mimic flash crowds. In: Proceedings of the 2nd conference on Symposium on Networked Systems Design, vol 2, pp 287–300

Liu Y, Wei W (2015) A replication-based mechanism for fault tolerance in mapreduce framework. Math Probl Eng 2015(1):1–7. http://www.hindawi.com/journals/mpe/2015/408921/

Oikonomou G, Mirkovic J (2009) Modeling human behavior for defense against flash-crowd attacks. In: Proc. IEEE Intl Conf. Comm

Palmieri F, Ricciardi S, Fiore U (2011) Evaluating network-based dos attacks under the energy consumption perspective: new security issues in the coming green ICT area. International Conference on Broadband. Wireless Computing, Communication and Applications (BWCCA), pp 374–379

Palmieri F, Ficco M, Castiglione A (2014a) Adaptive stealth energy-related dos attacks against cloud data centers. In: International Conference on Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), pp 265–272

Palmieri F, Fiore U, Castiglione A (2014b) A distributed approach to network anomaly detection based on independent component analysis. Concurr Comput Pract Exp 26(5):1113–1129

Palmieri F, Ricciardi S, Fiore U, Ficco M, Castiglione A (2015) Energy-oriented denial of service attacks: an emerging menace for large cloud infrastructures. J Supercomput 71(5):1620–1641

Paxson V (2001) An analysis of using reflectors for distributed denial-of-service attacks. ACM SIGCOMM Comput Commun Rev 31(3):38–47

Paxson V, Floyd S (1995) Wide area traffic: the failure of poisson modeling. IEEE/ACM Trans Netw 3(3):226–244

Rooj G (2011) Real stateful tcp packet filtering in ip filter. In: 10th USENIX Security Symposium

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

Tsunoda H, Ohta K, Yamamoto A, Ansari N, Waizumi Y, Nemoto Y (2008) Detecting drdos attacks by a simple response packet confirmation mechanism. Comput Commun 31(14):3299–3306

Wei W, Chen F, Xia Y, Jin G (2013) A rank correlation based detection against distributed reflection dos attacks. IEEE Commun Lett 17(1):173–175

Wei W, Liu Y, Zhang Y (2014a) TRLMS: two-stage resource scheduling algorithm for cloud based live media streaming system. IEICE Trans Inf Syst 97-D(7):1731–1734

Wei W, Zhang Y, Liu Y (2014b) A time-efficient solution to the general resource placement problem in cloud. Math Prob Eng 2014(1):1–10. http://www.hindawi.com/journals/mpe/2014/760458/

Xie Y, Yu S (2009a) A large-scale hidden semi-markov model for anomaly detection on user browsing behaviors. IEEE/ACM Trans Netw 17(1):54–56

Xie Y, Yu S (2009b) Monitoring the application-layer ddos attacks for popular websites. IEEE/ACM Trans Netw 17(1):15–25

Yu S, Zhou W, Doss R (2008) Information theory based detection against network behavior mimicking ddos attacks. IEEE Commun Lett 12(4):319–321

Yu S, Zhou W, Jia W, Guo S, Xiang Y, Tang F (2013) Discriminating ddos attacks from flash crowds using flow correlation coffcient. IEEE Trans Parallel Distribut Syst 23(6):1073–1080