Further Analyzing the Sybil Attack in Mitigating Peer-to-Peer Botnets

Tian-Zuo Wang[1, 2], Huai-Min Wang[1, 2], Bo Liu[1], Bo Ding[1, 2], Jing Zhang[1] and Pei-Chang Shi[1]
[1]School of Computer Science, National University of Defense Technology, Changsha 410073, China
[2]National Laboratory for Parallel and Distributed Processing, National University of Defense Technology, Changsha 410073, China
[email: phoenixwtz@163.com]
*Corresponding author: Tian-Zuo Wang

Received December 26, 2011; revised August 16, 201X; accepted September 20, 201X; published October 25, 201X

Abstract

Sybil attack has been proved effective in mitigating the P2P botnet, but the impacts of some important parameters were not studied, and no model to estimate the effectiveness was proposed. In this paper, taking Kademlia-based botnets as the example, the model which has the upper and lower bound to estimate the mitigating performance of the Sybil attack is proposed. Through simulation, how three important factors affect the performance of the Sybil attack is analyzed, which is proved consistent with the model. The simulation results not only confirm that for P2P botnets in large scale, the Sybil attack is an effective countermeasure, but also imply that the model can give suggestions for the deployment of Sybil nodes to get the ideal performance in mitigating the P2P botnet.

Keywords: Sybil attack, P2P botnet, Kademlia, prediction model, mitigation

This research was supported by the National Natural Science Foundation of China under Grant No.90818028
http://dx.doi.org/10.3837/tiis.2012.10.00

1. Introduction

The Sybil Attack, as described by Douceur [1], consists in creating a large number of fake peers called the Sybil nodes and laying them in a strategic way in the DHT to take control over a part of it. The Sybil attack is one of the most significant threats to peer-to-peer (P2P) networks. Although a number of studies has be conducted to defend against Sybil attacks[2][3][4], it is proved by Douceur that the Sybil attack cannot be totally avoided as long as the malicious entity has enough resources to create the Sybil nodes. Nowadays, under the trend that more and more botnets tend to build their command and control (C&C) mechanisms on P2P networks [5][6][7, the Sybil attack can be turned into an effective approach to mitigate P2P botnets.

Holz et al. [8] presented a case study showing how to use Sybil nodes to infiltrate the Storm botnet (a typical P2P botnet based on the Kademlia [9] protocol) and conducted the index poisoning. As a complementary work to that work, Davis et al. [10][11] explored the feasibility of a more general approach using Sybil nodes to mitigate this Kademlia based botnet, and this mitigating method is named the D-method here for short.

The D-method is detailed in **Fig. 1**. The D-method infiltrate the botnet with a large number of Sybil nodes, which seek to disrupt the communication between the bots by inserting themselves in the peer lists of "regular" bots, and eventually reroute or disrupt "real" C&C traffic. For example, according to the protocol of Kademlia, to achieve *C&C INFO* published by the botmaster on bot B6, the bot B1 has to inquire other nodes in the botnet to find <*Ckey*, *C&C INFO* >. Receiving request messages from B1, one bot will return the content indexed by *Ckey* if that is stored locally, otherwise, this bot will return to B1 other K nearest nodes from *Ckey*. Once the content indexed by *Ckey* is received, B1 will stop the search immediately, otherwise, it will inquire other three nodes not inquired yet for the content. As is shown in **Fig. 1**, if there are no Sybil nodes, B1 will get the *C&C INFO* stored in B6 through bots B2, B3, B4 and B5. However, after Sybil nodes are infiltrated in, B3 will return S1 to B1 as one of the K nearest nodes from *Ckey*, which leads B1 to inquire S1 for the content indexed by *Ckey*. The Sybil node S1 will intentionally return the counterfeit content *False INFO* to B1. This makes B1 take *False INFO* as *C&C INFO* and stop further searching iterations. In this way, Sybil nodes strike the C&C mechanism of the P2P botnet.

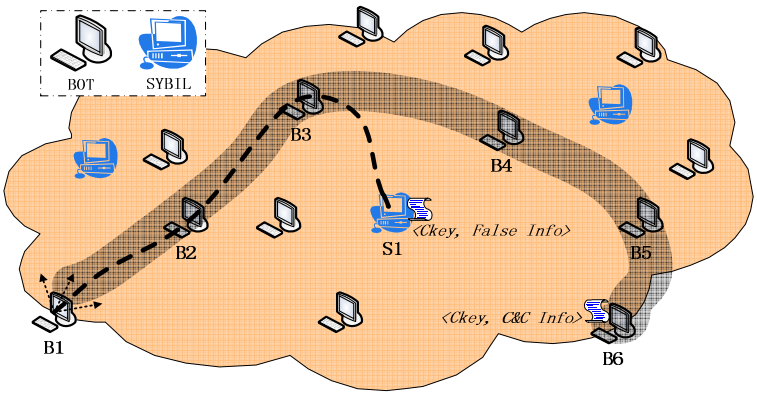

Fig. 1. The D-method

In their work, Davis et al. studied the impacts of some parameters such as the percentage of Sybil nodes, duration of the Sybil attack, size of a bot's peer-list, and so on. However, some other important parameters such as the size of the botnet, the value of K (which determines the number of nodes returned in reply to a node-finding request in Kademlia protocol) [9] were not studies, and also no theoretical guidance for mitigation in practice was given.

In order to help the defenders to decide how many Sybil nodes should be deployed to achieve the goal of mitigation in practice, we proposed a model predicting the upper bound and lower bound of mitigation effect of the D-method, and validate the model and explored the effect of some important parameters such as botnet size, sybil percentage and K value through simulations. In this paper, our study is based on botnets built on Kademlia protocol, so we call these botnets Kademlia-botnets for short.

The rest of this paper is organized as follows. In Section 2, the related works are reviewed. In Section 3, the model estimating the upper and lower bounds of the effect of the D-method is proposed. In Section 4, the simulation experiments and the analysis of results are reported. In section 5, some discussions are made. In Section 6, the whole paper is concluded and the focus of our future research is proposed.

2. Related Work

Mitigating botnets is an urgent task, because they are threatening the Internet seriously, through spam mails [12], denial of service attacks [13][14], and so on. Finding the C&C server of botnets and taking them down is an effective method to mitigate IRC botnets and HTTP botnets. However, to avoid the single point of failure, advanced botnets tend to adopt P2P protocols to construct their C&C mechanisms (such as Storm, Conficker, etc.), which is a big challenge to the mitigation research of botnets. Eliminating bots with anti-virus tools is often considered to counterattack P2P botnets, but the effectiveness is often limited by factors such as variations of bots, absence of AV software, not updated AV software, etc. For example, the Storm botnet was firstly discovered in January of 2007 [15], but in the first quarter of the year 2008, 20% of all spam emails were still Storm-generated [16]. Conficker botnet was discovered in November of 2008, but there were still millions of victims in 2010 and 2011 [17][18]. The first burst of Waledac botnet occurred in December of 2008 [19], while in early 2010, there are still more than 70 thousand zombies [20]. So the mitigation method without the elimination of bots must be focused on.

Holz et al. [8] proposed a method to separate a part of the P2P botnet from the rest. To eclipse a particular keyword *CKey*, they positioned a certain number of fake nodes closely around *CKey*, i.e., the DHT IDs of the nodes are closer to the hash value of *CKey* than the DHT IDs of any real peer. They then announced these nodes to the regular peers in order to "poison" the regular peers' routing tables and to attract all the route requests for keyword *CKey*. However, this method can only be effective to the particular keywords at a moment. In contrast, the D-method can simultaneously disrupt any search requests received by Sybil nodes, not limited to certain keys.

In the work by Wang et al. [21], two kinds of methods ("fake matches" and "stale contacts") are proposed to interrupt the access to correct contents in Kad network which is based on Kademlia protocol. The method of "fake matches" treats the regular nodes to cease the searching with the belief that the fake content they received is the "right" one. The method of "stale contacts" sends a number of stale contacts to regular nodes which forces the regular nodes to stop searching. Different from D-method, these methods employ only one or a few fake nodes. However, they rely on the efficient hijacking of the routing tables, which is now

very hard for a few fake nodes. The D-method utilizes a number of fake nodes, so the effectiveness is not limited.

Holz et al. [8] investigated index poisoning to control particular content. To prevent peers from retrieving search results for a certain key *CKey*, they publish a very large number of files using *CKey*. The goal of the pollution attack is to "overwrite" the content previously published under key *CKey*. To perform this attack, they first crawl the network, and then publish files to all those peers having at least the first 4 bits in common with *CKey*. This method is limited to the selected keys. For the botnets (such as Overbot [22]) whose keys used for searching are unpredictable, index poisoning is not suitable, but the D-method can still work well with these botnets.

In essence, the D-method exploits the sparsity of the nodeID space, the locality of the views of nodes in P2P botnets, and the mechanism of search ceasing. According to the Kademlia protocol, the length of nodeID is set 160bit, which means the network can accommodate 2^{160} nodes. However, the biggest botnet found today has no more than one hundred million bots. So there is adequate nodeID space for Sybil nodes. Because of the locality of views, each node has to find the target content with helps from other nodes, which provide the opportunities for Sybil nodes to launch attacks. The mechanism to cease the search makes bots stop search on receiving counterfeit contents from Sybil nodes, and bots are unable to check whether the content is true with the root nodes. These weaknesses are nearly unavoidable for P2P botnets, so that the Sybil attack of D-method is a more general and prospective approach to mitigate P2P botnets.

3. Prediction Model for the D-method

3.1 Simplifications and Hypotheses

It is hard to establish a precise model to predict the mitigation effect of the D-method, but it is feasible to predict the upper bound and lower bound. According to one conclusion which will be validated in the following, when a Kademlia-botnet is small, the mitigation effect is close to the lower bound, while with the increase of the size of botnet, the effect gets closer to the upper bound. So, with the predicting model for the upper bound and lower bound, the defenders can get a rough estimation for the performance of the D-method, which helps to adjust the intensity of mitigation.

To construct the model, we have to firstly simply the information-searching process through some concepts and hypotheses below.

Inquiry round: A round of inquiry begins with the sending of α requests, and ends with the receiving of α responses or timeouts.

Target space: The smallest subspace of nodeID that contains K root nodes of the content to be searched for is call target space. It can be proved that if K is powers of 2, the target space contains only the K root nodes, otherwise, there may exist non-root nodes, but the number of non-root ones will be less than K. For convenience of computing, the number of nodes in the target space is approximately set K in this paper. K is an important parameter for the Kademlia protocol, because each content is published on K root nodes and each node-finding request will be replied with K other nodes.

Search space: The nodeID subspace in which α nodes to be inquired in one inquiry round are distributed. According to the protocols of structured P2P, the search space shrinks after each inquiry round. So each inquiry round is a shrinking process of search space.

Hypothesis 1: The whole search is composed of inquiry rounds; one round will not start before the end of the previous round. In each inquiry round, α requests are issued.

Hypothesis 2: Through each inquiry round, the size of the search space shrinks by a factor of U. So U is called the shrinking rate in this paper.

Hypothesis 3: In the Kademlia-botnet, bots and Sybil nodes are both distributed randomly in the nodeID space, and the proportion of Sybil nodes is x.

3.2 Prediction Model For The Upper Bound And Lower Bound Of The Effectiveness Of The D-Method

The effectiveness of D-method refers to the failure probability of the search process under D-method. Our model is to predict the upper bound and lower bound of this probability.

In order to calculate the failure probability of one search process, we have to know how many inquiries are needed to get the target of search. According to Subsection 3.1, the process of search is in fact a shrinking process of the search space. If the shrinking rate is U, as is discussed in Hypothesis 2, we can simplify the process of search into the process in the second column of Fig. 3. N_search stands for the number of nodes in the search space, K is the number of nodes in the target space, and R is the number of inquiry rounds. In the initialization, N_search is set to be the number of nodes in the initial search space, and R is initialized as 0. Before the search space shrinks onto the target space, N_search is bigger than K, so at least one more inquiry round is needed. In the next inquiry round, R would be increased by 1 and the N_search would shrink to be N_search/U. Otherwise, if the search space has already shrunk onto the target space, N_search would be no bigger than K, so no more inquiry round would be needed and we get the final value of R. When the search space is equal to the target space, in order to eventually get the target content, one more inquiry is needed. However, among the K nodes in the target space, Sybil nodes may also exist. If the first response in the last inquiry round comes from a Sybil node, the search will fail, otherwise, the search will succeed. So in fact, for the last inquiry round, it is reasonable to say that there is only one useful inquiry. Thus, in the whole process of search, the total number of inquiries is $\alpha \cdot R + 1$. For example, if the number of nodes in the initial search space is M0, then when the search space shrinks onto the target space, the number of inquiry round will be $R = \log_U M0 - \log_U K$, and $\alpha \cdot R + 1$ requests will be sent out in total.

Provided with the number of inquiries during one search process, we can calculate the failure probability of this search process. According to hypothesis 3, the probability that a request message is sent to a Sybil node is always x. So providing that the initial search space is M0, the probability that the search fails because of Sybil nodes would be $1-(1-x)^{\alpha \cdot R+1}$.

If M0 is known, the failure probability will be figured out. The value of M0 for one search process is determined by the location of the target space. The routing table of the Kademlia node is composed of multiple k-buckets [9]. A k-bucket is a segment of the routing table, which can accommodate the pointers to K nodes whose distances from the host node is in $[2^i, 2^{i+1})$ measured by XOR of nodeIDs. The k-bucket corresponding to $[2^i, 2^{i+1})$ is called the NO_(i) k-bucket in this paper, and the set of nodeIDs that are $[2^i, 2^{i+1})$ far away from the host node is called the scope of NO_(i) k-bucket. It can be proved that the length of the path from one node to any node in its NO_(i) k-bucket scope is the same in the sense of statistics, which results in the same probability to meet with Sybil nodes when searching for any node in the NO_(i) k-bucket scope. Thus, for one bot, if only the *Ckey*s of the target contents are located in the same k-bucket scope, the probabilities that the bot fails when searching for the target

contents will be the same. In this sense, for one bot, the k-bucket scope where the *Ckey* of the target content is located should be the initial search space.

Now, provided with the location of the target space, we can get the failure probability of one search process. So, if we know how the target space is distributed, we can get the distribution of the failure probability. Let it be supposed that the nodeID is composed of L bits, and the total number of nodes in the network is M. When a bot B issues a random request for the content indexed by *Ckey*, the probability that *Ckey* is located in the scope of NO_(L-i) k-bucket of B is $P_i = \frac{2^{L-i}}{2^L} = 1/2^i$. Thus, the probability that *Ckey* is located in the scope of NO_(L-1) k-bucket of B is 1/2. The size of the scope of NO_(j) k-bucket will be halved when the value of j is decreased by one. hen the scope shrinks onto the target space, the proportion it takes in the total space will be $\frac{K}{M} = 1/2^{\log_2 \frac{M}{K}}$. Thus the serial number of the smallest k-bucket containing the target space should be $L - \left\lfloor \log_2 \frac{M}{K} \right\rfloor$.

Table 1. The distribution of failure probability

i	No. of k-bucket	Probability that target space is in the k-bucket scope.	Number of nodes in initial search space	Number of iteration rounds	Failure probabilitly
1	L-1	1/2	$M_1=M/2$	$R_1 = \log_U M_1 - \log_U K$	$P_1 = 1-(1-x)^{\alpha \cdot R_1+1}$
2	L-2	$1/2^2$	$M_2=M/2^2$	$R_2 = \log_U M_2 - \log_U K$	$P_2 = 1-(1-x)^{\alpha \cdot R_2+1}$
3	L-3	$1/2^3$	$M_3=M/2^3$	$R_3 = \log_U M_3 - \log_U K$	$P_3 = 1-(1-x)^{\alpha \cdot R_3+1}$
h	L-h	$1/2^h$	$M_h=M/2^h$	$R_h = \log_U M_h - \log_U K$	$P_h = 1-(1-x)^{\alpha \cdot R_h+1}$
. . .	…	…	…	...	...
$\left\lfloor \log_2 \frac{M}{K} \right\rfloor$	$L - \left\lfloor \log_2 \frac{M}{K} \right\rfloor$	$1/2^{\left\lfloor \log_2 \frac{K}{M} \right\rfloor}$	$M_{\left\lfloor \log_2 \frac{M}{K} \right\rfloor} = M/2^{\left\lfloor \log_2 \frac{M}{K} \right\rfloor}$	$R_{\left\lfloor \log_2 \frac{M}{K} \right\rfloor} = \log_U M_{\left\lfloor \log_2 \frac{M}{K} \right\rfloor} - \log_U K$	$P_{\left\lfloor \log_2 \frac{M}{K} \right\rfloor} = 1 - (1-x)^{\alpha \cdot R_{\left\lfloor \log_2 \frac{M}{K} \right\rfloor}+1}$

The distribution of the target space and the probabilities of failure during searching are told in **Table 1**. In addition, the probability that the self of B is located in the target space is $1/2^{\left\lfloor \log_2 \frac{M}{K} \right\rfloor}$, and then the probability to fail in search is certainly 0. So, the cumulative sum of

all the probabilities for the location of B is 1. According to the analysis above, for any node, the probability that it fails in searching for a random content should be as follows.

$$P = \sum_{i=1}^{\left\lfloor \log_2 \frac{M}{K} \right\rfloor} \frac{1}{2^i} P_i + 1 \Big/ 2^{\left\lfloor \log_2 \frac{M}{K} \right\rfloor} \cdot 0 = \sum_{i=1}^{\left\lfloor \log_2 \frac{M}{K} \right\rfloor} \frac{1}{2^i} (1-(1-x)^{\alpha \cdot \log_U \frac{M}{K \cdot 2^i}+1}) \quad (1)$$

According to the protocol of Kademlia, in each inquiry round the search space is minified to a half at most, so U is no less than 2. Thus, we get the model to predict the upper bound of the effectiveness of mitigation.

$$P_{upper} = \sum_{i=1}^{\left\lfloor \log_2 \frac{M}{K} \right\rfloor} \frac{1}{2^i} (1-(1-x)^{\alpha \cdot \log_2 \frac{M}{K \cdot 2^i}+1}) \quad (2)$$

While searching for certain content, one node constructs a set that contains the K nearest nodes known to itself from *Ckey*, which is named K-list here. In each inquiry, the node to inquire is select from K-list, and nodes nearer to *Ckey* are preferred. In the ideal condition, before each inquiry round, all nodes in the K-list will be located in the search space, which means that each round will minify the search space by K, and then U equals K. However, a lot of k-bucket is not full in practice, which leads to the existence of nodes out of the search space in K-list. In the meanwhile, it is also hard to guarantee that K-list will located in the next search space after updates. So the maximum value of U is K, and the model to predict the lower bound of mitigation effectiveness is:

$$P_{lower} = \sum_{i=1}^{\left\lfloor \log_2 \frac{M}{K} \right\rfloor} \frac{1}{2^i} (1-(1-x)^{\alpha \cdot \log_K \frac{M}{K \cdot 2^i}+1}) \quad (3)$$

Two instantiations of the surfaces of P_{upper} and P_{lower} are given in Fig. 2. The mathematical expectations of the mitigation rates should lie between the two surfaces. In Fig. 2 (a), K is set to be 20, α is set to be 3, and the values of P_{upper} and P_{lower} are calculated with different values of x and M. It is obvious that both the upper bound and the lower bound would increase with the growth of x, and both would increase with the growth of M. In Fig. 2(b), M is set to be 32768, α is set to be 3, and the values of P_{upper} and P_{lower} are calculated with different values of x and K. It is obvious that with the growth of K, both the upper bound and the lower bound would decrease.

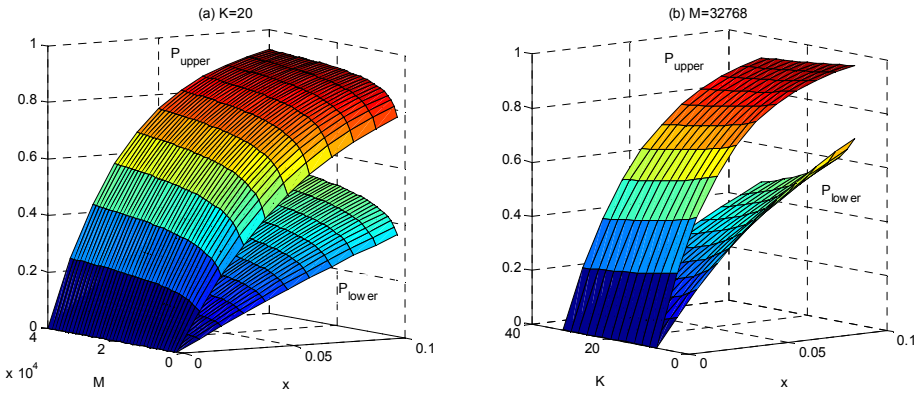

Fig. 2. The surfaces of Pupper and Plower

The diagram of this model is shown in Fig. 3. The first column shows all the k-bucket scopes that the target space may be located in. The second column illustrates the simplified search process for each k-bucket scope. Through the simplified search processes, we can figure out the average numbers of inquiries when searching for targets located in different k-bucket scopes. The third column shows how to calculate the failure probabilities when searching for targets located in different k-bucket scopes, with the results from the simplified search processes. The fourth column tells the probabilities for the target to be located in different k-bucket scopes. Through multiplying the results of the third column and the fourth column, and then summing up them, we get our model. If the value of U is 2, we get the upper bound. If the value of U is K, we get the lower bound.

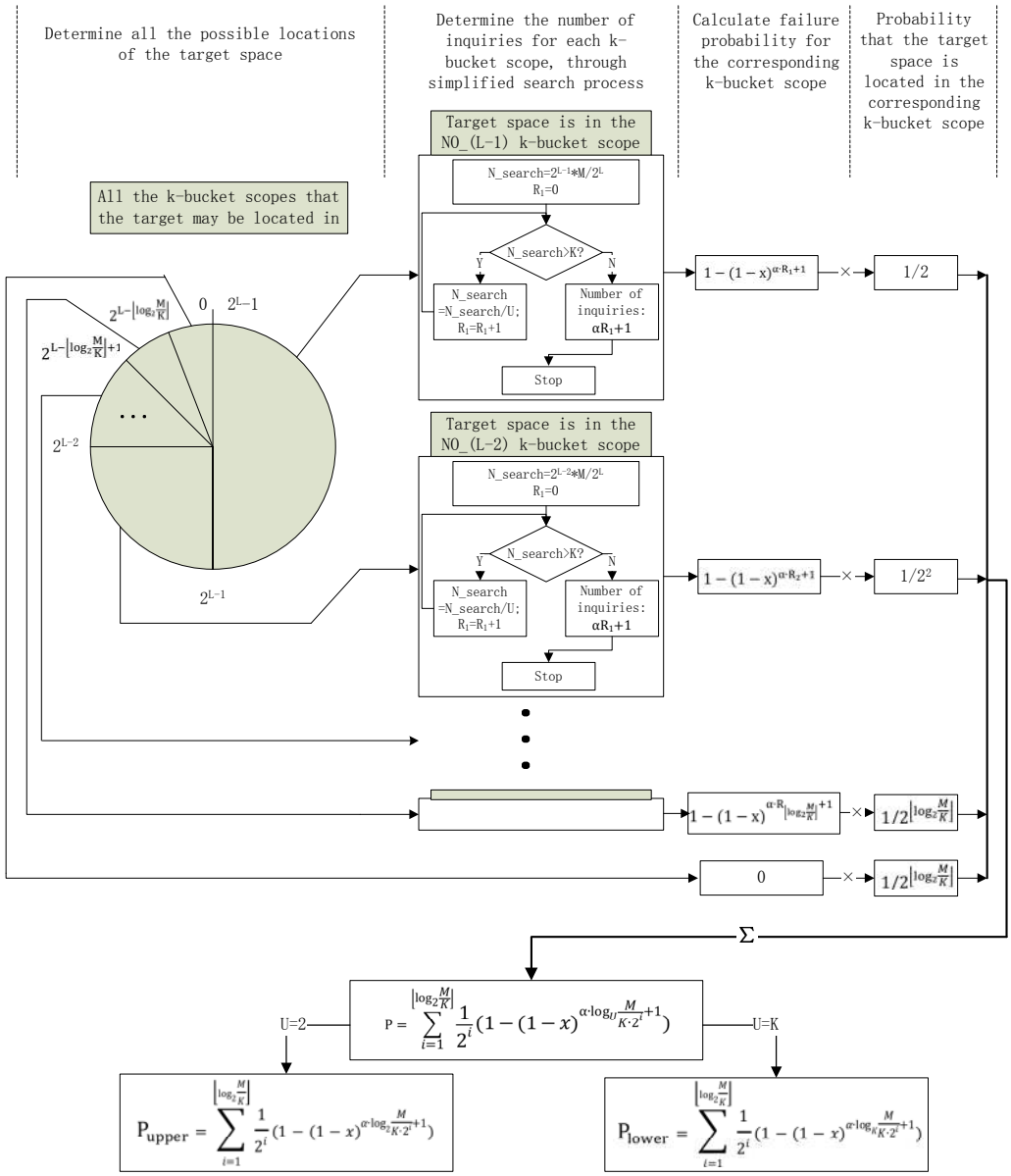

Fig. 3. The model to predict the effectiveness of the D-method

4. Experimental Results and Analysis

In order to validate the model and to find what kind of impact different factors have on the D-method, simulation experiments are conducted.

There are some important variables to consider in experiments, including *Sybil percentage*, *mitigation rate*, size of Kademlia-botnet (*n_size*), number of Sybils (*n_sybil*), number of bots (*n_bot*), number of failed searching (*ns_fail*), total number of searching (*ns_total*), and so on. The following conditions are satisfied.

$$n\_size = n\_sybil + n\_bot \tag{4}$$

$$Sybil\ percentage = n\_sybil / n\_size \tag{5}$$

$$mitigation\ rate = ns\_fail / ns\_total \tag{6}$$

4.1 Simulation Platform

The experiments are carried out on our simulation tool based on PeerSim [23]. PeerSim is a well-known open sourced simulation tool written in JAVA for P2P networks, and Furlan and Bonani [24] implemented Kademlia protocol for PeerSim. Based on these works, we devised the Kademlia protocol implementation and added certain functions into PeerSim. Thus we got a simulation tool for autogenetic Kademlia-botnets, which can simulate D-method and calculate the effectiveness. The PeerSim supports two kinds of simulation modes, one is cycle-based which is efficient but ignores the transport layer in the communication protocol stack, the other is event-based which is less efficient than the former but supports transport layer simulation. Our simulation platform works upon the event-based mode.

In the simulator, there are three parameters that must be set, including Sybil percentage, n_bot and K (which determines the number of bots to return when receiving FIND_NODE message in Kademlia). The settings of these parameters would be detailed in Subsection 4.2. The requests for C&C information are issued concurrently and asynchronously, and the concurrent number is conventionally set 3, which means that at most three requests from a bot can exist simultaneously in the network. Under each set of parameters, the simulator runs for 20 times and average results are adopted.

During each time of the simulations, at least two searching actions are issued by every node in the botnet, and the objectIDs for the searching actions are generated by the simulator randomly, for example, if there are 2000 nodes in the network, 4000 searching actions will be issued. The value of ns_fail and ns_total are recorded accumulatively, and the mitigation rate is computed at the end of each time of simulation. The Kademlia-botnets are generated with Sybils and bots existing at the very beginning, and the initial degree for each node is set 100.

4.2 Results and Analysis

4.2.1 Impact of Sybil percentage

In order to study the impact of *Sybil percentage* on the effectiveness of D-method, simulations are carried out with the numbers of bots being respectively 2048, 4096, 6144, 8192, 16384 and 32768. The value of *K* during the simulation is set 20, and the results are shown in **Fig. 4**. It is obvious that the *mitigation rate* of D-method on C&C information delivery of Kademlia-botnet increases significantly with the *Sybil percentage*s, no matter how large the scale of the botnet is. This is because the bigger the density of Sybil nodes is, the bigger the probability to inquire a Sybil node during searching will be. The experimental result implies that increasing the *Sybil percentage* is an effective approach to enhance the mitigation effect.

It is also shown in **Fig. 4** that the curves of the upper bound model and the lower bound model have the same variation tendency with the simulation result, and the two models are

validated for that the simulation results are always higher than the lower bound and lower than the upper bound predicted by the model.

In **Fig. 4** we see that when bot number = 2048, some simulation results are even a bit lower than the lower bound. One of the possible reasons is that if the objectIDs are close to the searching nodes for most search actions, rather than located evenly in the objectID space, the average number of inquiry rounds would be smaller than the mathematical expectation. Although the possibility for this case is small, this would result in a mitigation rate lower than the lower bound. For the case of bot number = 32768, there are also some simulational results higher than the upper bound. One of the possible reasons is that if the objectIDs are far from the searching nodes for most search actions, rather than located evenly, the average number of inquiry rounds would be bigger than the mathematical expectation. Although the possibility for this case is small, this would result in a mitigation rate higher than the upper bound.

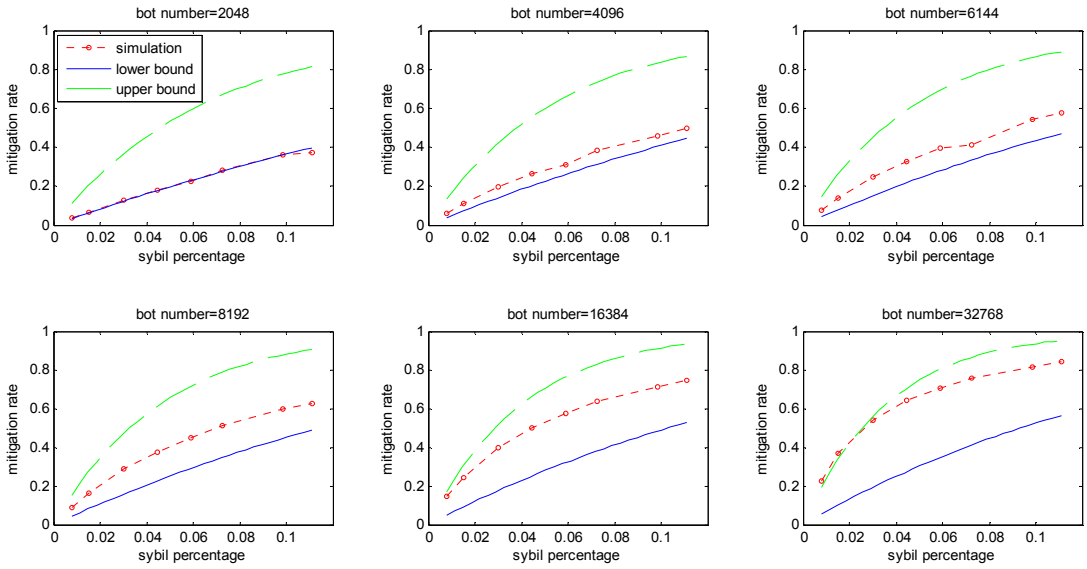

Fig. 4. The impact of *Sybil percentage*

In fact, the prediction model in this paper is a probabilistic model, which means that although the mathematical expectations of experimental results should always lie between the upper bound and the lower bound, there are still chances that a few results deviate far from their mathematical expectations and lie out of the bounds when the number of experiments is limited.

4.2.2 Impact of the Botnet Size

In order to study the impact of the size of the botnet on the effectiveness of the D-method, simulations are carried out with *Sybil percentages* being respectively 3.03%, 4.48%, 5.88%, 7.25%, 9.86% and 11.11%. During our simulations, the value of K is set 20, and the results are shown in **Fig. 5**. It is obvious that the mitigation rate of the D-method increases significantly with the scale of the Kademlia-botnet, whatever the *Sybil percentage* is. This is because the average length of the path increases with the enlarging of the network, which leads to the increase of inquiry rounds and also the increase of the probability to inquire a Sybil node. The curves describing the relationships between mitigation rates and *Sybil percentages* are shown

in **Fig. 6** with bot number respectively being 2048, 4096, 6144, 8192, 16384 and 32768, and it is easy to find that the curves shifts upwards when the network scale increases. When the bot number is 32768, 70.7% of the total searching activities can be mitigated with *Sybil percentage* being only 5.88%, and 81.7% will be mitigated with *Sybil percentage* being 9.86%.

It can be further discovered in **Fig. 5** that the variation tendency of the results of the upper bound and lower bound models are the same with the results of simulation. What is to be noticed is that the simulation results are nearer to the predictions of the lower bound model when botnets are small, while with the increase of the botnet scale, the simulation results tends to be closer to the predictions of the upper bound model. This may implies that the Sybil attack is very suitable to mitigate Kademlia-botnets in large scale.

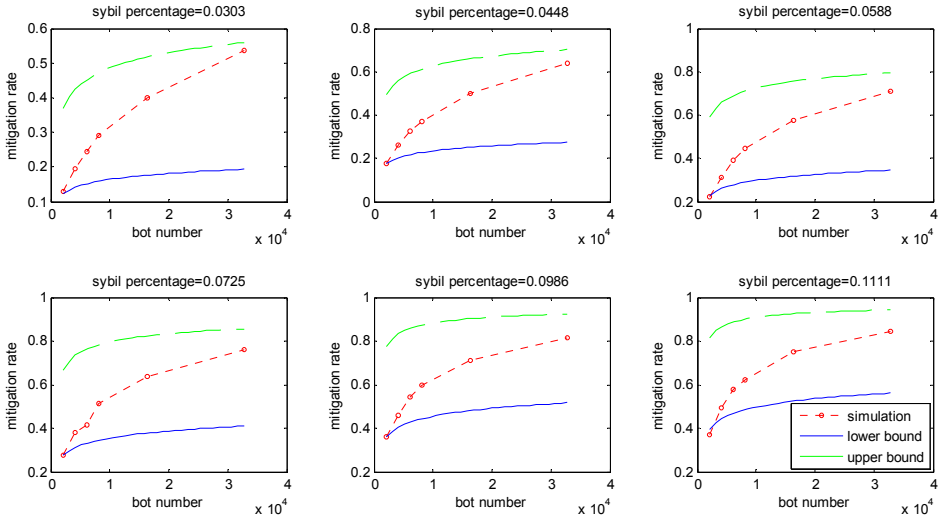

Fig. 5. The impact of the network size

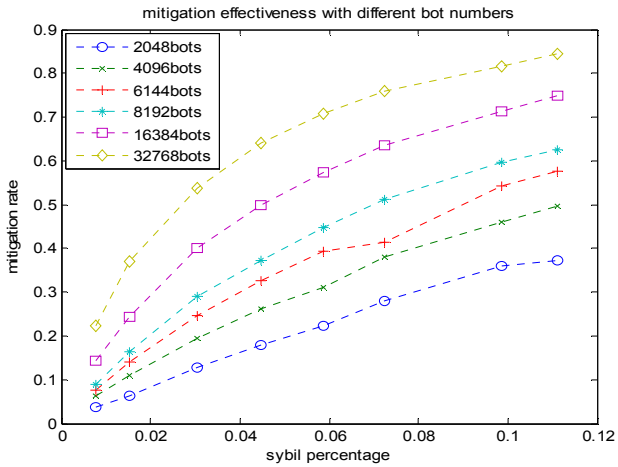

Fig. 6. The comparison of mitigation effects on Kademlia-botnets of different sizes

4.2.3 Impact of K

In order to study the impact of K on the mitigation effect of D-method, with the *Sybil*

percentage being respectively set 0.775%, 1.538%, 3.03%, 4.48%, 5.88% and 7.25%, simulations are carried out to observe the change of mitigation rate when K changes. During our simulations, the bot number is set 4096, and the results are shown in **Fig. 7**. It is obvious that with the increase of K, the mitigation rate decreases significantly. This is because when K increases, on the one hand the target content will be stored on more root nodes, on the other hand the K-list will be enlarged which will lead to a higher value of U and a smaller probability to inquire a Sybil node. The curves describing the relationships between mitigation rates and *Sybil percentages* with K being respectively 8, 10, 16, 20 and 32 are shown in **Fig. 8**, and it is easy to see that with the increase of K, the curves shifts downwards.

In addition, as is shown in **Fig. 7**, the variation tendencies of the simulation results are the same with the predictions of the upper and lower bound model. In **Fig. 7**, whatever the value of *Sybil percentage* is, the mitigation rate is much closer to the prediction of the lower bound model rather than the upper one. This is because the bot number is always 4096, which is a small number. According to the analysis above, when the scale of the botnet increases, the mitigation rate will be closer to the prediction of the upper bound model.

The impact of K observed implies that for the constructer of Kademlia-botnet, a bigger K is a good choice, while for the defender, a smaller K is better.

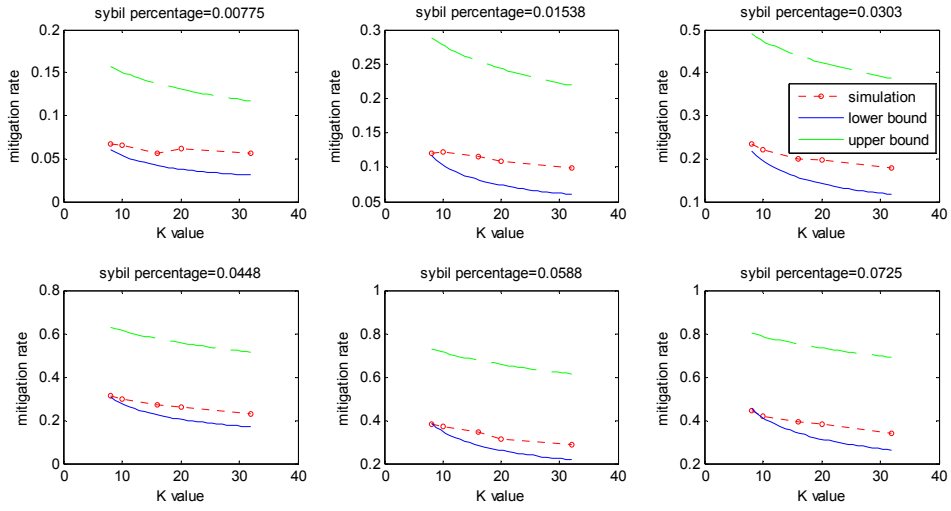

Fig. 7. The impact of K

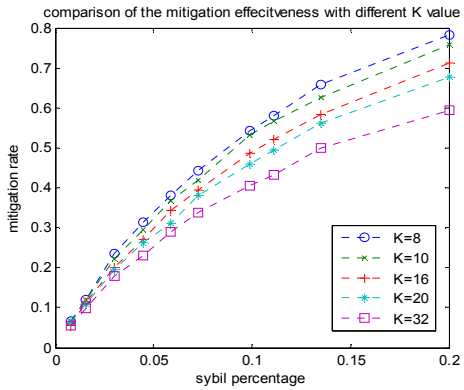

Fig. 8. The comparison of mitigation effect with different K values

4.2.4 Shrinking Marginal Utility

In order to describe the mitigation effectiveness of single Sybil node, we proposed a metric called Single Miti-effect. If each bot in a Kademlia-botnet issues a search request for certain content, then the average number of bots whose searching activities are mitigated by one Sybil node is the Single Miti-effect. It can be proved that the Single Miti-effect E satisfies the equation below.

$$E = \frac{mitigation\ rate}{sybil\ rate} \tag{7}$$

The Single Miti-effects of Sybil nodes under different network scales and different *Sybil percentages* are shown in **Table 2**. It is easy to see that under a fixed *Sybil percentage*, the Single Miti-effect increases with the network scale, which agrees with the above analysis of the relationship between network size and mitigation rate. What is to be noticed is that under a fixed network scale, Single Miti-effect decreases with the increase of *Sybil percentage*. This implies that there is an obvious phenomenon of diminishing marginal utility for Sybil nodes in D-method, which may be a drawback of Sybil attack.

Table 2. Diminishing marginal utility of E

bot number / *E* / Sybil percent	2048	4096	6144	8192	16384	32768
0.00775	4.81	7.99	9.83	11.46	18.46	28.83
0.01538	4.16	7.10	9.19	10.60	15.89	24.06
0.0303	4.23	6.46	8.09	9.60	13.20	17.73
0.0448	3.97	5.83	7.30	8.30	11.12	14.30
0.0588	3.78	5.29	6.69	7.60	9.75	12.03
0.0725	3.85	5.25	5.71	7.05	8.77	10.46
0.0986	3.64	4.66	5.50	6.05	7.22	8.29
0.1111	3.35	4.46	5.18	5.62	6.75	7.60

4.3 Summary

According to the analysis on the impact of the botnet scale, for Kademlia-botnets in large scale, Sybil attack is a good choice for defenders. According to the analysis on the impact of the *Sybil percentage*, promoting the percentage of Sybil nodes is an effective approach to enhance the mitigation. According to the analysis on the impact of K value, the smaller K is, the better D-method works. However, there is an obvious phenomenon of diminishing marginal utility for Sybil's Single Miti-effect, which may be a defect of Sybil attack.

Anyway, Sybil attack is an effective approach to mitigate Kademlia-botnets in large scale.

5. Discussion

5.1 Sybil Attack and the Trend of Miniaturization

The work by Davis *et al.* [25] points out that structured P2P is an ideal structure to build botnets for its robustness, while according to the analysis above, the enlargement of botnets make them weaker in front of Sybil attack, just as the situations shown in **Fig. 9**. Then there is a question: is Kademlia really suitable to build an autogenetic botnet in large scale?

Our opinion is that structured P2P protocol is suitable to build parasitic botnets in large scale, because the P2P structure have a high robustness and the benign nodes give a good

protect against Sybil attacks. While for autogenetic botnets, this is a question need further studies.

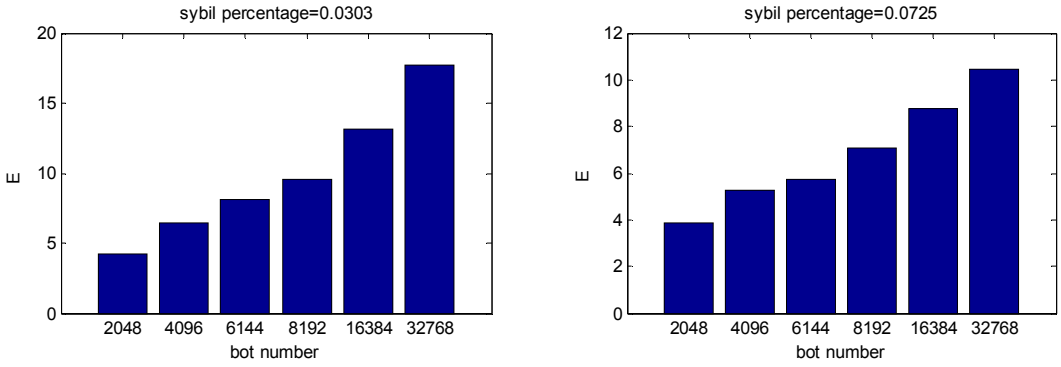

Fig. 9. *E* changes with the number of bots

Now there is a trend for botnets to be smaller. As is shown in the report from Damballa [26] which monitored more than 600 botnets for 3 months, only 5% of all the botnets are of the size larger than 10000 bots, and 57% are of the size smaller than 100 bots. The work by Cooke *et al.* [27] considered that one of the reasons for this trend is the improvement of defending technology which make the building and surviving of large botnets a harder job, and the other reason is that the higher bandwidth makes the small botnet powerful enough to complete some work which needs a larger botnet before. There is also a reason that the delivery of C&C information in a large P2P botnet tends to be slower. From our analysis in this paper, the challenges from the Sybil attack that the value of E increases with the size of the botnet may also be a force for the trend of miniaturization.

5.2 Deployment of the D-method

Although Davis et al. only analyzed their D-method through simulations [10, 11] rather than deploying it in real systems, in fact there are at least two ways to deploy the Sybil nodes in real networks. One is centralized, utilizing the local one or more servers of high performance and the adequate bandwidth. Each Sybil node is implemented as an application instance on the server: it contacts with other bots and mitigates their searching activities, using local resources and the nodeID assigned by the defender. The other is decentralized, utilizing computing and network resources in many distributed computers. Each computer runs one or more Sybil nodes just like the case in the first way, and all the Sybil nodes together are conducted by the defender.

The first way is easy to implement, and for the botnets such as StormNet [8], it works well as is shown by the simulation. However, for botnets which checking IP addresses of bots (e.g. different bots must have different IP addresses), it may be necessary to exploit IP harvesting, which is not an easy work. Further, once the local servers are discovered by the botmasters, the chances are that they will be attacked by a large number of bots.

For the second way, there are not those limitations. Sybil nodes are distributed, so the probability of survival under the attack of botnets is much higher than the first way. The more important advantage is that there are a lot of IP addresses and computing resources, thus the bottleneck no longer exists. The cost of this way is obviously bigger than the first one.

We favor the second way. Firstly, the decentralized way is nearly capable to mitigate all kinds of P2P botnets. Secondly, the capacity of IP harvesting may be limited. Last but not the least, although the cost seems higher than the first one, it may be not high when considering

the fact that the decentralized Sybil network is reusable in mitigating other P2P botnets. Let's take a look at the ongoing techniques. When a new botnet is discovered, defenders may try to update the anti-virus software, but because of the limitations discussed in related work above, during the long term before the most of bots can be eliminated by the anti-virus software, defenders must take measures to reduce the damages of botnets. This may involves development and deployment of a particular tool suited to a particular botnet, cooperation with different organizations including ISPs and governments, and so on. What's more, for each one different botnet, those activities may need to be conducted once again. That is a big cost. So if there is a method that can be reusable to mitigate different botnets, it is definitely a big reduce to the cost. Thus, the second way has much higher performance than the first way and much lower cost than other methods.

In fact, in front this kind of serious threats which utilizes huge number of infected computers, it is hard to mitigate them effectively every time utilizing only comparatively a handful of resources. So, we believe that in order to mitigate P2P botnets, the decentralized way is a good choice.

5.3 Possible Countermeasures

A CA for a secure nodeID assignment can prevent faulty nodes from selecting nodeIDs freely. However, the existence of CA brings in the single point of failure for botnets, which damages the robustness of botnets. So most important methods that focusing on how to solve the Sybil problem are decentralized and may be adopted by P2P botnets to defend against the D-method.

SybilGuard [28] and SybilLimit [29] make use of social relationships between nodes to detect Sybil nodes. In these methods, before accepting an unknown node, an honest node checks whether this unknown node has enough social relationships with other honest nodes. If this honest node has enough random routes that have intersection nodes with the random routes of the unknown node, the unknown one is accepted. However, these methods are not suitable for P2P botnets. First, SybilGuard and SybilLimit need the out-of-band authentication (e.g. via phone calls) between nodes, which is crucial to ensure the basic assumption that the number of attack edges (the edges connecting the Sybil nodes and the honest nodes) is small. It is hard for P2P botnets to ask bots to get through the out-of-band authentication (via phone calls for example) when connecting to the botnets. Rather, the work by Yang et al. [30] pointed out that in a real system, most Sybil nodes only form attack edges, and only 20% of Sybils are connected with other Sybils, which contested another basic assumption of SybilGuard and SybilLimit. Second, for every bot, each acceptance of a neighbor involves more network and local activities, which reduces the covert of botnets.

Hyeong S. Kim et al. [31] proposed a method called ELiSyR, which can realize efficient, lightweight and Sybil-resilient file search in P2P networks. Considering that the vulnerability of each node depends on its degree [32], ELiSyR favors lower-degree nodes to higher-degree ones when forwarding messages and this reduces the chances of Sybil nodes responding to search queries. This method is more suitable to unstructured P2P networks whose degree distribution is unbalanced. However, for structured P2P botnets such as Storm, the distribution of degrees of nodes is much more balanced, which limits the effect of ELiSyR against the D-method.

In the work by Kapadia et al. [33], the algorithm HA_locate is proposed. This algorithm generates multiple redundant messages to search for one key, and tries to prevent the routing paths of these separate messages from overlapping. HA_locate enables the searching node to obtain the correct results with the existence of faulty nodes. Although the redundant messages make the botnets less covert, it can limit the effect of the D-method.

6. Conclusion

Botnet mitigation is a relatively new and a challenging research area, especially for P2P botnets. In this paper, we analyzed the principal of the D-method, and pointed out that the feasibility relies on the sparsity and locality of bots. According to the study of Kademlia protocol, based on some rational simplification and hypotheses, the model to predict the upper and lower bound of mitigation effectiveness are proposed. Through simulation experiments, the impacts of different factors on the mitigation effectiveness are observed, and we found that the bigger the *Sybil percentage*, the network scale or the value of K is, the better the mitigation effectiveness of the Sybil attack is. The work in this paper shows that for structured P2P botnets, Sybil attack is an effective approach to mitigate them, and the prediction model helps when deciding how many Sybil nodes are needed to achieve the goal of mitigation.

However, we must realize that the D-method is still not excellent enough: the Sybil nodes worked separately and no cooperation exists, and rather, some anti-Sybil methods (e.g. HA_locate) can limit the effect of mitigation by the D-method. So, as a future work, we will focus on how to make Sybil nodes automatically cooperate together in order to achieve better effect of mitigation against P2P botnets and counterattack the anti-Sybil methods that may be adopted by P2P botnets.

References

[1] John R. Douceur, "The Sybil Attack," *Peer-to-Peer Systems Lecture Notes in Computer Science*, Vol. 2429/2002, pp. 251-260, 2008. Article (CrossRef Link)

[2] Miguel Castro, Peter Druschel, Ayalvadi Ganesh, Antony Rowstron, Dan S. Wallach, "Secure routing for structured peer-to-peer overlay networks," in *Proc. of 5th symposium on Operating Systems Design and Implementation*, Dec 2002. Article (CrossRef Link)

[3] Hosam Rowaihy, William Enck, Patrick McDaniel, and Thomas La Porta, "Limiting Sybil Attacks in Structured P2P Networks," in *Proc. of 26th IEEE International Conference on Computer Communications*, pp. 2596-2600, Jun 2007. Article (CrossRef Link)

[4] T. Cholez, I. Chrisment and O. Festor, "Evaluation of sybil attacks protection schemes in kad," Scalability of Networks and Services, pp. 70-82, 2009. Article (CrossRef Link)

[5] P. Wang, L. Wu, B. Aslam, and C. C. Zou, "A systematic study on peer-to-peer botnets," in *Proc. of* 18th *International Conference on Computer Communications and Networks*, pp.1-8, Aug 2009. Article (CrossRef Link)

[6] Natalya Fedotova, Luca Veltri, "The case for in-the-lab botnet experimentation: creating and taking down a 3000-node botnet," in *Proc. of 26th Annual Computer Security Applications Conference*, pp. 141-150, Dec 2010. Article (CrossRef Link)

[7] JB Grizzard, V. Sharma, C. Nunnery, BB Kang, and D. Dagon, "Peer-to-peer botnets: Overview and case study," in *Proc. of* 1st *Hot Topics in Understanding Botnets*, Apr 2007. Article (CrossRef Link)

[8] T. Holz, M. Steiner, F. Dahl, E. W. Biersack, and F. Freiling, "Measurements and mitigation of peer-to-peer-based botnets: a case study on Storm worm," in *Proc. of 1st USENIX Workshop on Large-Scale Exploits and Emergent Threats*, Apr 2008. Article (CrossRef Link)

[9] Petar Maymounkov and David Mazières, "Kademlia: A Peer-to-Peer Information System Based on the XOR Metric," *Peer-to-Peer Systems Lecture Notes in Computer Science*, Vol. 2429/2002, pp. 53-65, 2002. Article

(CrossRef Link)

[10] C. Davis, J. Fernandez, S. Neville, and J. McHugh, “Sybil attacks as a mitigation strategy against the Storm botnet,” in *Proc. of 3rd International Conference on Malicious and Unwanted Software*, pp.32-40, Oct 2008. Article (CrossRef Link)

[11] C.Davis, J. Fernandez, S. Neville, and B. Victoria, “Optimising sybil attacks against p2p-based botnets,” in *Proc. of 4th International Conference on Malicious and Unwanted Software*, pp. 78-87, Oct 2009. Article (CrossRef Link)

[12] HyunCheol Jeong, Huy Kang Kim, Sangjin Lee and Eunjin Kim, "Detection of Zombie PCs Based on Email Spam Analysis", *KSII Transactions on Internet and Information Systems*, vol. 6, no. 5, May 2012. Article (CrossRef Link)

[13] Kai Chen, HuiYu Liu and XiaoSu Chen, "Detecting LDoS attacks based on abnormal network traffic", *KSII Transactions on Internet and Information Systems*, vol. 6, no. 7, Jul 2012. Article (CrossRef Link)

[14] Jing Zhang, Huaping Hu and Bo Liu, "Robustness of RED in Mitigating LDoS Attack", *KSII Transactions on Internet and Information Systems*m, vol. 5, no. 5, may 2011. Article (CrossRef Link)

[15] Raimund Genes, Anthony Arrott, David Sancho, “Stormy Weather: A Quantitative Assessment of the Storm Web Threat in 2007,” Dec 2011. Article (CrossRef Link),

[16] “MessageLabs Intelligence: Q1/March 2008 --One Fifth of All Spam Springs from Storm Botnet”, http://www.messagelabs.co.uk/mlireport/MLI_Report_March_Q1_2008.pdf, December, 2011.

[17] Seungwon Shin and Guofei Gu, "Conficker and Beyond: A Large-Scale Empirical Study," in *Proc. of 26th Annual Computer Security Applications Conference*, Dec 2010. Article (CrossRef Link)

[18] “infection tracking,” Dec 2011. Article (CrossRef Link)

[19] Gilou Tenebro, “W32.Waledac Threat Analysis,”, Nov 2011. Article (CrossRef Link)

[20] Dan Goodin, “Waledac botnet 'decimated' by MS takedown,” Oct 2011. Article (CrossRef Link)

[21] P. Wang, J. Tyra, ames, E. Chan-Tin, T. Malchow, D. F. Kune, N. Hopper, Y. Kim, “Attacking the kad network,” in *Proc. of 4th International Conference on Security and Privacy in Communication Networks*, Sep 2008. Article (CrossRef Link)

[22] Guenther Starnberger, Christopher Kruegel, Engin Kirda, “Overbot-A botnet protocol based on Kademlia,” in *Proc. of 4th International Conference on Security and Privacy in Communication Networks*, Sep 2008. Article (CrossRef Link)

[23] Montresor A, Jelasity M, “PeerSim: A Scalable P2P Simulator,” in *Proc. of 9th International Conference on Peer-to-Peer Computing*, pp. 99-100, Sep 2009 Article (CrossRef Link)

[24] http://peersim.sourceforge.net/code/kademlia.zip, Apr 2011.

[25] C. Davis, S. Neville, J. Fernandez, J.M. Robert, and J. McHugh, “Structured peer-to-peer overlay networks: Ideal botnets command and control infrastructures?,” in *Proc. of 13th European Symp. on Research in Computer Security*, pp. 461-480, Oct 2008. Article (CrossRef Link)

[26] Gunter Ollmann, “Botnet Size within the Enterprise,” http://blog.damballa.com/?p=361, Mar2011.

[27] E. Cooke, F. Jahanian, and D. McPherson, “The zombie roundup: understanding, detecting, and disrupting botnets,” in *Proc. of Steps to Reducing Unwanted Traffic on the Internet Workshop*, pp. 39-44, Jul 2005. Article (CrossRef Link)

[28] H. Yu, M. Kaminsky, B. P. Gibbons and A. Flaxman, “SybilGuard: Defending against sybil attacks via social

networks," in *Proc. of ACM Special Interest Group on Data Communication*, Sep 2006. Article (CrossRef Link)

[29] H. Yu, P. Gibbons, M. Kaminsky and F. Xiao, "SybilLimit: A near-optimal social network defense against sybil attacks," in *Proc. of 29th IEEE Symposium on Security and Privacy*, May 2008. Article (CrossRef Link)

[30] Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai, "Uncoveringsocial network sybils in the wild," in *Proc. of the ACM Internet Measurement Conference*, pp. 259-268, Nov 2011. Article (CrossRef Link)

[31] Hyeong S. Kim,Eunjin Jung and Heon Y.Yeom, "ELiSyR: Efficient, Lightweight and Sybil-Resilient File Search in P2P Networks ," *KSII Transactions on Internet and Information Systems*, vol. 4, no. 6, Dec 2010. Article (CrossRef Link)

[32] Lazaros K Gallos, Reuven Cohen, Panos Argyrakis, Armin Bunde and Shlomo Havlin, "Stability and Topology of Scale-Free Networks under Attack and Defense Strategies," *Physical Review Letters*, vol. 94, no. 18, pp. 188701, May 2005. Article (CrossRef Link)

[33] Apu Kapadia, Nikos Triandopoulos, "Halo: High-Assurance Locate for Distributed Hash Tables," in *Proc. of 15th Annual Network and Distributed System Security Symposium*, Feb 8-11, 2008. Article (CrossRef Link)

Tian-Zuo Wang, is currently a Ph. D. candidate in School of Computer Science, National University of Defense Technology, Changsha, China. His current research interests include distributed computing and information security.

Huai-Min Wang, Ph.D., professor in School of Computer Science, National University of Defense Technology, Changsha, China.. His research interests include distributed computing, information security and software engineering.

Bo Liu, Ph.D., associate professor in School of Computer Science, National University of Defense Technology, Changsha, China. His research interests include information security.

Bo Ding, Ph.D., research assistant in School of Computer Science, National University of Defense Technology, Changsha, China. His research interests include distributed computing.

Jing Zhang, is currently a Ph.D. candidate in School of Computer Science, National University of Defense Technology, Changsha, China. Her research interests include Network and information security, cryptography.

Pei-Chang Shi, research assistant in School of Computer Science, National University of Defense Technology, Changsha, China. His current research interests include distributed computing.