Learning-based Threshold Estimation Model for Detection of Botnet Malicious Activities

Do-Hoon Kim\(^1\), Hoh Peter In\(^2\), and Hyun-Cheol Jeong\(^2\)

\(^1\)Dept. of Computer and Radio Communications Engineering, Korea University, Seoul, Korea
\(^2\)Korea Internet & Security Agency, Seoul, Korea

Abstract - Recently botnet malicious activities such as 7.7 DDoS attack and SPAM have wreaked havoc on the Internet-based information systems. Especially, botmasters have taken advantage of these botnets to cause various types of harms on networks. Accordingly, studies have ensued in search of various botnet detection technologies, without much success due to serious limitations such as the inability to set forth general and qualitative thresholds for detection. The traditional botnet-detecting methods contain thresholds arbitrarily defined based on personal experience and opinions of experts. The arbitrary nature has increased the possibility of false positives and false negatives. Thus, it is important to objectively define the thresholds for major malicious activities such as DDoS, spam and botnet propagation. Aware of the problem, a model is proposed herein, which, based on the training model of the Markov chain, is designed to estimate thresholds and, thereby, to detect malicious botnet activities. In addition, successfully created as a dynamically trained threshold, our model better detects malicious activities.

Keywords: botnet, DDoS, spam, propagation, Markov chain, threshold training.

1 Introduction

Basically, a botnet serves, for example, as a vehicle for sending spam mails, and for scanning ports. Once ports are detected, it extends the size of the group infected by it. Furthermore, to generate political and economic benefits, it is used for waging DDoS attacks, which flood victims' servers with unwanted information.

A 7.7 DDoS attack exemplifies a typical botnet attack. On July 7, 2009, for example, bots swarmed major Korean websites such as major portal sites, and the homepages of political parties, banks, and the Blue House. The attack was waged through the following four processing phases [1]:

Step 1: DDoS attack on websites in USA
Step 2: DDoS attack on websites in Korea
Step 3: Transmission of SPAM mails

Step 4: Self-destruction

In sum, botnets took on a variety of malicious forms, and, thereby, enabled diverse attack scenarios. In order to detect these malicious activities, most of the botnet detection systems (BDS) are equipped with detection basis such as threshold. However, most threshold-setting methods are based either on the qualitative factors such as experts’ opinions or on well-known empirical values. For this reason, botnet detection systems have not performed up to expectation, rendering it more difficult to measure the precise threshold of each activity.

Aware of the problem, a model is proposed herein, which, based on Markov chain (MC), is to estimate thresholds in the our botnet detection system. At first, factors are monitored in the traffic affected by botnets, such as frequency of sending packets, MX/SMTP query rate and multi-connection rate for a predefined duration of a time window. This log provides information necessary for creation of time series data. Based on the data, training is conducted, by means of the Markov assumption, for a predefined duration of a time window (i.e. 6 hours).

Then, thresholds are induced from the training information, which are in turn applied to a botnet detection system for different durations. Upon application, our proposed model showed a significant high detection rate of 91.8% and an average negative rate of 8%.

The rest of this paper is organized, as follows: Related works are described in Section 2, followed by description of the theoretical Markov chain; Section 4 presents our proposed model, along with a case study; and, Section 5 concludes the study.

2 Related Work

The following are the major works related to threshold estimation of malware:

H. Choi [2] designed a botnet detection algorithm, which was to analyze the DNS query similarity among botnet
groups. In order to analyze the similarity, it is important to set up a standard threshold. Despite the high detection rate, however, the threshold was not clearly defined. Choi used the empirical threshold, and acknowledged the qualitative limitation of his study in the "future work" section. The threshold random walk (TRW) [3] serves as the vehicle for estimating the density of botnet, and helps detect the botnet. As a powerful statistical tool, the TRW has been used to detect port-scanning [4] and spam-laundering attempts [5]. Al-Hammadi [6] observed the traffic logs of an ordinary network. Under the Al-Hammadi scheme, botnet detection is rendered, when an abnormal behavior is spotted and an arbitrarily defined threshold is reached. But, the algorithm has revealed fundamental problems such as false positives and false negatives in the dynamic network environment. Accordingly, focus was put on each individual malicious activity, and the time series data of each of them has been created. And then, the time series data was trained by the Markov chain and used for defining thresholds. Finally, we have modified our botnet detection system for comparison, based on our thresholds, of the detected events with real events.

3 Background

The Markov chain [7] concerns a sequence of random variables (e.g. $X_0, X_1, X_2$, etc.) and their Markov properties. Here, the current conditions of a system are independent of the system's past and the future conditions. Thus,

$$Pr(X_{n+1} = x \mid X_n = x_n, ..., X_1 = x_1, X_0 = x_0)$$

$X_0$ constitutes an element of the countable set $S$, which stands for a state space of the chain. Directed graphs are often used to represent Markov chains. The graphs are labeled as a percentage value that represents the possibility that an edge may migrate from one state to another. Figure 1 illustrates the conceptual mechanism.

![Figure 1. Overview of Markov Chain](image)

If the machine is in the state $y$ at the time $n$, then the current state alone affects the probability that it moves to the state $x$ at the time $n+1$, without any influence from the time $n$.

4 Threshold Estimation Model for Botnet Malicious Activities

Most botnet detection system include threshold measuring step for detection of botnet malicious activities. Also, almost system is depend on expert's experience and decide thresholds. Accordingly, it is important how to estimate the threshold without existing way.

The module of malicious activity detection (i.e. : bold line module) in Figure 2 is the threshold estimation process in the our botnet detection system.

![Figure 2. Overview of Botnet Detection System](image)

In this module, we adopted the concept of learning algorithm which is the Markov chain. From this algorithm, the threshold is trained and created the threshold for detection of botnet malicious activities such as DDoS, spam and botnet propagation.

4.1 Threshold Estimating Process

It is important to understand the conceptual map illustrated in Figure 3 in order to estimate the thresholds of malicious botnet activities such as the frequency of sending packets, MX/SMT query rate and multi-connection rate, for a particular duration of the time window.

![Figure 3. Conceptual Overview of Threshold-Estimating Model](image)

First of all, as shown in Figure 1, created is the time series data of each malicious activity, based on the botnet
traffic log covering 6 hours. Then, the Markov chain trains the data. In the process, the threshold is created and utilized to create the prediction time series data, which is compared with the time series data of the next time window (i.e. analysis of likelihood).

Finally, the process is repeatedly analyzed. In the following subsections, the analytical process is to be described in detail.

4.2 Threshold Training

In Figure 4, the information \((\alpha, \beta, \gamma)\) on a malicious activity was monitored from the detected botnet traffic log DB, and each time series log is extracted from it.

![Figure 4. Threshold Training](image)

From this log information, the status of each factor is defined, as shown in Table 1.

<table>
<thead>
<tr>
<th>Table 1. Definition of Each State</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Sending frequency: (S_{\text{sendFreq}} = {S_1, S_2, S_3, S_4})</td>
</tr>
<tr>
<td>• MX/SMTP query rate: (S_{\text{mxsmtpreq}} = {S_1, S_2, S_3, S_4})</td>
</tr>
<tr>
<td>• Multi-connection rate: (S_{\text{multi-conn}} = {S_1, S_2, S_3, S_4})</td>
</tr>
</tbody>
</table>

Also, the transition matrix \((T)\) represents the relations of each state to another. The following equation sums up the matrix that covers a 24-hour period consisting of four 6-hour training sessions:

\[
T = \begin{bmatrix}
S_1 & S_2 & S_3 & S_4 \\
S_1 & S_2 & S_3 & S_4 \\
S_1 & S_2 & S_3 & S_4 \\
S_1 & S_2 & S_3 & S_4 \\
\end{bmatrix}
\]

\[
T = \frac{\text{number of each state transition}}{\text{total number of transition}}
\]

(2)

Equation (3) sets forth the definition of the initial vector set \((x)\), based on the previous 6-hour data, as follows:

\[
\pi = \begin{bmatrix}
\frac{S_1}{S_1 + S_2 + S_3 + S_4} \\
\frac{S_2}{S_1 + S_2 + S_3 + S_4} \\
\frac{S_3}{S_1 + S_2 + S_3 + S_4} \\
\frac{S_4}{S_1 + S_2 + S_3 + S_4} \\
\end{bmatrix}
\]

(3)

4.3 Likelihood Analysis

Using the estimated thresholds, the likelihood analysis is executed, as shown in Figure 5.

![Figure 5. Likelihood Analysis](image)

At first, the Markov chain (i.e. threshold training) leads to creation of the predicted time series data. Equation (4) works out the estimate value of the likelihood between the previous data on the time series and the next data.

\[
-2LL = -2 \sum_{i=1}^{n} Y_i \ln P_i + (1 - Y_i) \ln (1 - P_i)
\]

(4)

where \(Y_i = \text{observation value}\), \(P_i = \text{predicted value}\)

In the equation (4), the closer the likelihood value gets to 0, the higher the accuracy becomes.

4.4 Detection of Malicious Activity

The last stage concerns detection of malicious activities by means of estimated thresholds \((\alpha, \beta, \gamma)\), as shown in the following figure:

![Figure 6. Detection of Malicious Activities](image)
In Figure 6, when the input data (i.e. packet sending frequency, MX/SMTP query rate and multi-connection rate) is bigger than the estimated threshold, the malicious botnet activities are detected.

5 Case Study

For our experiments, we used network trace data collected at a dormitory of the Korea University between the 1st and the 31st October, 2009. A dormitory of the Korea University consists of 11 class-C networks with autonomous subnetworks. Most of the IP space is allocated, but many subnetworks have inactive IPs. We collected information about TCP SYN/ACK connection traffic for port-scanning by botnet. In this network traces, 3 types of botnet groups were infected, including typical IRC bot (Rbot, Sbot) and HTTP bot (Padbot).

The focus was not on the types or the versions of botnets; rather, it was on their malicious activities. Then, the infected botnet groups were monitored for every 24-hour period. In addition, all botnets were controlled with automatic script tools, and activated naturally. In addition, all botnets were controlled with automatic script tools, and activated naturally. Then, Equation (4) computed the maximum likelihood (ML) of the average trained threshold and the detection rate (DR).

- ML of DDoS Attack: 0.29 with (DR of 88%)
- ML of spam: 0.2 (DR of 92%)
- ML of propagation: 0.14 (DR of 96%)

Finally, the trained threshold was applied to create the prediction time series, and Figure 5, 6, and 7 show the types of the resulting malicious activities.

In Figure 7, 8, and 9, the blue line illustrates the chronological trend of the minimum threshold required for detection of actual malicious activities. In other words, the line indicates the values of the critical mass for a particular time zone. Based on the values, it was accurately determined whether a malicious act had occurred or not.

On the other hand, the red line illustrates the trend of the post-training values of the critical mass. The more similar behavioral pattern the red line shows, the better the detection results are. Table 2 sums up the test results.

<table>
<thead>
<tr>
<th></th>
<th>DDoS</th>
<th>SPAM</th>
<th>Propagation</th>
</tr>
</thead>
<tbody>
<tr>
<td>True Positive</td>
<td>95%</td>
<td>93%</td>
<td>95%</td>
</tr>
<tr>
<td>True Negative</td>
<td>67%</td>
<td>100%</td>
<td>50%</td>
</tr>
<tr>
<td>False Positive</td>
<td>50%</td>
<td>0%</td>
<td>50%</td>
</tr>
<tr>
<td>False Negative</td>
<td>5%</td>
<td>7%</td>
<td>5%</td>
</tr>
</tbody>
</table>

Moreover, as shown in Table 3, our model predicted, with more than a 90% accuracy, the thresholds of various activities in terms of precision, recall and accuracy. Thus, it is fair to say that they were well-trained.
Table 3. Precision, Recall and Accuracy

<table>
<thead>
<tr>
<th></th>
<th>DDoS</th>
<th>SPAM</th>
<th>Propagation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>95%</td>
<td>100%</td>
<td>95%</td>
</tr>
<tr>
<td>Recall</td>
<td>95%</td>
<td>93%</td>
<td>95%</td>
</tr>
<tr>
<td>Accuracy</td>
<td>91%</td>
<td>95%</td>
<td>92%</td>
</tr>
</tbody>
</table>

6 Conclusions

In this study, a Markov chain-based quantitative analysis model has been proposed, which differs from the traditional methods based on empirical thresholds.

Our model shows how to design a threshold training mechanism with the observed data on the detected botnet malicious activity log. Especially, by setting forth the estimated values of learning-based thresholds, the detection rate has dramatically improved. Judging from the case study, it is fair to say that our proposed method of estimating thresholds has succeeded in accurately detecting botnet malicious activities.

Finally, it is necessary to conduct further researches on the issues such as the matrix size and the possibility of applying our approach to various network environments. Furthermore, our model needs proving its general applicability through repeated execution of training mechanisms such as Monte-Carlo simulation, and needs to incorporate in it advanced training algorithms such as the hidden Markov model.

7 Acknowledgement

This study was funded by the IT R&D program of MKE/KEIT. (Project title: 2008-S-026-02, The Development of Active Detection and Response Technology against Botnet); and also funded by the R&BD Support Center of Seoul Development Institute and the South Korean government (Project title: WR080951, Establishment of Bell Labs in Seoul / Research of Broadband Convergent Networks and their Enabling Technologies).

8 References