# A Behavior Approach to Instant Messaging Worm Detection

W. Guo, L. Wang
School of Computer Science
Shenyang Aerospace University
Shenyang, China

H.X. Zhou
School of Information Science and Technology
Liao Ning University
Shenyang, China

***Abstract*—In this paper, we present a behavior approach to detect Instant Messaging (IM) worm attacks. We extract characteristics of IM worm behaviors by analyzing the mechanism of IM worm propagation and define the corresponding characteristic functions the values of which can distinguish IM worm behaviors from normal user behaviors. Our approach starts to work through two stages. First stage, the training stage, we learn the means and deviations of characteristic functions from a profile. Second stage, the detection stage, simplified Mahalanobis distance is utilized to calculate the similarity of new data against the pre-computed profile. To make the detection mechanism insensitive to site and access pattern, a non-parametric Cumulative Sum (CUSUM) method is applied to this measure and generates an alert when the distance of the new input exceeds the allowable distance the algorithm set. As a result, IM worms can be detected in a fully automatic and very efficient fashion.The evaluation results show that the detection mechanism has short detection latency and high detection accuracy.**

***Keywords- instant messaging worms; simplified mahalanobis distance; non-parametric cumulative sum (CUSUM) method***

## I. INTRODUCTION

Instant Messaging (IM) services are very popular as an instant way of communication for tens of millions of Internet users over the Internet. Popular systems such as MSN Messenger (Windows Messenger in Windows XP) [1], Yahoo! Messenger (YIM) [2], AOL Instant Messenger (AIM) [3], and ICQ [4] have changed the way we communicate with friends, acquaintances, and business colleagues. However, multiple vulnerabilities have been discovered and have yet to be discovered in Instant Messaging clients[5]. As a result, they pose great security challenges.

IM worms are different from the regular scanning and email worms [5].Although researchers have exerted large efforts to understand and contain the propagation of scanning worms and email-worms[6-9],these researches are not quite suitable for IM worms due to the different infection mechanism. M. Williamson et al apply Virus Throttling to Instant Messaging worms that spread over Instant Messaging service to slow the spread of worms. It works by preventing an infected machine infecting many others. But there are many limitations [10]. It may delay the valid messages and be too restrictive for IM users allowing only one new contact/day and so on.

This paper presents a behavior approach to detect Instant Messaging worms. We first compute during a training stage the means and deviations of the characteristic functions the values of which are diverse from normal user behaviors to IM worm behaviors. Then, simplified Mahalanobis distance is utilized during the detection phase to calculate the similarity of new data against the pre-computed profile. To make our approach insensitive to site and access pattern, non-parametric CUSUM algorithm is applied to this measure and generates an alert when the distance of the new input exceeds the allowable distance the algorithm set. We demonstrate the effectiveness of the method on the data collected from a university IM server.

## II. PROPOSED APPROACH

### *A. Characteristic Formulation*

We find some characteristics which can distinguish IM worm behaviors from normal behaviors. After the load of worm code, IM worm may send a malicious URL in a text message to different users. So we can deduce that the proportion of URL sent will increase. We define function Count($x$) as the number of different users one user communicates with using the same value $x$. For example, if one user sends www.google.com to four different friends in contact list, then Count(www.google.com) is equal to four. To describe this characteristic, we define characteristic function of *URL()* as (1).

$$URL\,() = \begin{cases} \underset{\forall URL \in U}{Max} \{Count(URL)\}, U \neq \Phi \\ 0, U = \Phi \end{cases} \tag{1}$$

where $U$ is the set of URL a user sends.

Another common infection characteristic is that victims send files with same size and same content. Actually, these files are IM worms. Of course human’s behavior can also cause Instant Messaging communication with the same characteristic. Considering users do not do this all the time, we still use this characteristic as one of the worm’s behavior characteristics. To describe this characteristic, we define characteristic function of file transfer requests as (2).

$$FileReq() = \begin{cases} \underset{\forall a \in A}{Max} \{Count(a)\}, A \neq \Phi \\ 0, A = \Phi \end{cases} \tag{2}$$

where $A$ is the set of file size a user sends.

We also consider the number of friends one user communicates with during a certain period. When users use IM software, they can choose which friend or several friends in the contact list to communicate. However, a worm tries to spread as fast as possible, so it may communicate with a large number of friends in the contact list, which deviates form the normal user behavior. Since an IP address can represent a friend in the contact list, we define characteristic function *IPAder()* to describe this characteristic as (3).

$$IPAddr\ () = Number\ of\ distinct\ IP\ address \tag{3}$$

### B. Simplified Mahalanobis Distance

Mahalanobis distance is most commonly used as a multivariate outlier statistic. The metric essentially addresses whether the (unknown) new sample would be considered an outlier relative to the previously data profiles. Here we compute the distance between the characteristic distributions of the newly observed traffic against the profile learned in the training phrase. The higher the distance scores, the more likely this traffic is anomalous.

The Mahalanobis distance is defined as:

$$D(x,\bar{y}) = (x,\bar{y})^T C^{-1}(x,\bar{y}) \tag{4}$$

where $x$ and $y$ are two characteristic vectors, and each element of the vector is a variable. $x$ is the characteristic vector of the new observation, and $y$ is the averaged characteristic vector computed from the training examples. And $C^{-1}$ is the inverse covariance matrix as $C_{ij}=Cov(y_i\ ,y_j)$. $y_i\ ,y_j$ are the *i*th and *j*th elements of the training vector.

The Mahalanobis distance has the advantage of utilizing means and variances for each variable, and the correlations and covariance between measures. Instead of simply computing the distance from means, it weights each variable by its standard deviation and covariance, so the computed value gives a statistical measure of how well the new example matches (or is consistent with ) the training samples.

In our problem, we use the assumption that the characteristics are statistically independent. Although the assumption does not typically hold, the Mahalanobis distance does provide a useful measure of the deviation of a current traffic profile from the baseline. Thus, the covariance matrix *C* becomes diagonal and the elements along the diagonal are just the variance of each characteristic. As a result, we derive the simplified Mahalanobis distance:

$$d(x,\bar{y}) = \sum_{i=0}^{n-1} \frac{(x_i - \bar{y}_i)^2}{\sigma_i^2} \tag{5}$$

where the variance is replaced by the standard deviation. In our problem, *n* is set to three (since we have chosen three characteristics).

When contacting with friends by IM systems, users may not always use it due to the busy study or work. As a result, the values of characteristic functions may be below the corresponding mean, but that doesn't mean that it is anomalous. So such a deviation should not contribute to Mahalanobis distance. Consequently, we use (6) to compute simplified Mahalanobis distance.

$$d(x,\bar{y}) = \sum_{i=0}^{n-1} \frac{((x_i - \bar{y}_i)^+)^2}{\sigma_i^2} \tag{6}$$

where x+ is equal to x if x > 0 and 0 otherwise.

In order to make the detection approach insensitive to site and access pattern, a non-parametric Cumulative Sum (CUSUM) method is applied to this measure. We will discuss it in section 2.3.

### C. Non-parametric CUSUM Algorithm

Although we can set a threshold to detect IM worms, the non-parametric CUSUM method is applied to make our approach more general. This method enjoys all the virtues of sequential and non-parametric test, and the computation load is very light.

Let $\{X_n, n=1,2,3......\}$ be the simplified Mahalanobis distance sequence observed during the detection stage. One of the assumptions for the non-parametric CUSUM algorithm is that the mean of the random sequence is negative during normal conditions, and becomes positive when a change occurs. In general, $E(X_n) = \alpha$ .Thus, without loss of any statistical feature, $\{X_n\}$ is transformed into another random sequence $\{Z_n\}$, i.e. $Z_n = X_n - \beta$, where $\beta > \alpha$. Parameter β is a constant value for a given network condition, and it helps to produce a random sequence $\{Z_n\}$ with a negative mean so that all the negative values of $\{Z_n\}$ will not accumulate according to time. Therefore, we define $\{Z_n\}$ in our worm detection approach as follows:

$$\mathrm{Z_n} = X_n - \beta \tag{7}$$

Suppose that the increase in the mean of $\{Z_n\}$ during the worm attack can be lower-bounded by $\theta$ . Our change detection is based on the observation of $\theta >> \alpha$. The recursive version of non-parametric CUSUM algorithm is shown as follows:

$$\begin{aligned} y_n &= (y_{n-1} + Z_n)^+, \\ y_0 &= 0, \end{aligned} \tag{8}$$

where x+ is equal to x if x > 0 and 0 otherwise.

And there is another definition of the non-parametric CUSUM algorithm which is illustrated as follows:

$$y_n = S_n - \min_{1 \le k \le n} S_k \tag{9}$$

where $S_k = \sum_{i=1}^{k} Z_i$ , with $S_0 = 0$ at the beginning, and $y_n$ is our test statistic.

In order to reduce the overhead for online implementation, we use the recursive version. Let $d_N(.)$ be the decision at time *n*: '0' for normal operation (homogeneity) and '1' for attack (a change occurs). Here *N* represents the worm detection threshold:

$$d_N(y_n) = \begin{cases} 0, y_n \le N; \\ 1, y_n > N. \end{cases} \tag{10}$$

In other words, $d_N(y_n)=1(y_n>N)$, where $d_N(.)$ is the indicator function. The effect of introducing β is to offset the possible positive mean in $\{X_n\}$ so that the test statistic $y_n$ will be reset to zero frequently and will not accumulate with time. In our problem, $\beta$ and $N$ is set to three and twelve.

## III. Experiment Results

Our experiment is based on simulation. Below, we describe the simulation environment and proceed to present the results of the experiments. We collected a dataset of 521 users of a University IM server (the IM service is only available within the campus) and split the dataset into two chronological parts for training and detecting worms respectively. In particular, the first 80% of the trace was used as training data and the rest 20% was used for detecting IM worms. This simulates the process of learning from the historical data and applying the proposed algorithm to current and future data. Moreover, we simulated IM worm infection by sending files or URLs in text messages which are two major vectors for IM worm propagation [5] to the online friends of the contact list every five minutes. Then, a mixture of both normal and IM worm traffic was used in the proposed algorithm. And the worm sets were inserted at random places in the test data.

### A. Normal Traffic

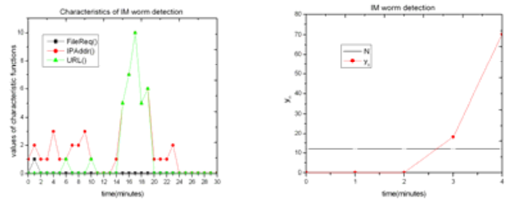
Due to the busy work or hard study, users may not communicate with friends in the contact list all the time. So there is no message communicated through IM service sometimes, especially after midnight. The results are shown in Table 1.

TABLE I. LEARNING RESULTS

| ***characteristic*** | $\mu$ | $\sigma^2$ |
|---|---|---|
| ***URL ()*** | 1.333312 | 0.420157 |
| ***FileReq()*** | 1.271003 | 0.236540 |
| ***IPAddr ()*** | 2.600212 | 0.737141 |

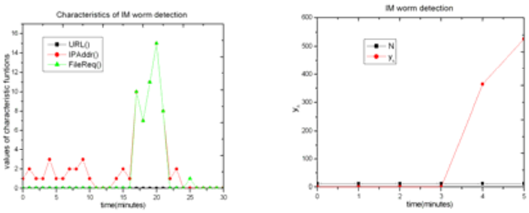
We observed that there are few file transfer requests and URLs in text messages when normal users use IM service. In most of cases, users communicate with each other by the text messages. From the results, we also observed that the means of URL() and FileReq() are 1.333312 and 1.271003, and the corresponding variances are 0.420157 and 0.236540. That means that although users send URLs in text messages or file transfer requests, they usually send the same URL or file only to one or two different friends. Although the mean and variance of IPAddr() is 2.600212 and 0.73714, it is still much smaller than the values of IPAddr() after adding IM worm traffic. We will show worm detection results in next section.

### B. Worm Detection



(a) Characteristic of IM det. (b) IM worm detection results.

FIGURE I. DETECTING IM WORMS WHICH PROPAGATE BY SENDING URLS IN TEXT MESSAGES



(a) Characteristic of IM detection. (b) IM worm detection results.

FIGURE II. DETECTING IM WORMS WHICH PROPAGATE BY SENDING FILES

Figure 1 shows the simulation of IM worms which propagate by sending URLs in text messages. (a) shows the change of characteristic functions presented in section 3.1. We observed that the values of *URL* () are not larger than one and the values of *IPAddr()* change within the range of zero to three when there is no IM worm traffic. However, the values of *URL* () and *IPAddr()* change abruptly to the summit of ten simultaneously after the introduction of IM worms. And there is no change on the values of FileReq(). As a result, IM worms can be detected in one sample since the introduction of IM worms as (b) shown.

Figure 2 shows the simulation of IM worms which propagate by sending files. (a) shows that the values of *FileReq()* are not larger than one and the values of *IPAddr()* change from one to three without adding IM worm traffic. However, the values of *FileReq()* and *IPAddr()* differ from the normal values after the introduction of IM worms. They change beyond seven simultaneously and the summits of them are fifteen. The values of *FileReq()* are zero all the time. Consequently, (b) shows that this approach can detect IM worm in one sample after the introduction of IM worms.

## IV. Conclusion

This paper presents a behavior approach to detect Instant Messaging worms. We first compute during a training stage the means and deviations of the characteristic functions the values of which are diverse from normal user behaviors to IM worm behaviors. Then, simplified Mahalanobis distance is utilized during the detection phase to calculate the similarity of new data against the pre-computed profile. To make our approach insensitive to site and access pattern, non-parametric CUSUM algorithm is applied to this measure and generates an alert when the distance of the new input exceeds the allowable distance the algorithm set. We demonstrate that the effectiveness of the approach on the data collected from a university IM server.

## ACKNOWLEDGEMENT

This work is supported by the national natural science foundation of China under

Grant Nos. 61300233 and Grant Nos. 61402298, Shenyang Science and Technology Foundation No. F13-316-1-35,the PhD Start-up Fund of National Science Foundation of Liaoning Province No. 20131086, and the transportation technology project of  Liaoning Province under Grant Nos.201502.

## REFERENCES

[1] Microsoft. MSN Messenger. http://messenger.msn.com/Yahoo! Inc. Yahoo! Messenger.http://messenger.yahoo.com/

[2] America Online, Inc. AOL Instant Messenger. http://www.aim.com/

[3] ICQ Inc. ICQ Pro 2003b. http://www.icq.com/

[4] Mannan M, van Oorschot PC. On Instant Messaging worms, analysis and countermeasures. In: Proc. of the ACM CCS Workshop on Rapid Malcode (WORM 2005). Fairfax, 2005.

[5] Wei Yu, Xun Wang,Prasad Calyam, Dong Xuan,Wei Zhao,Modeling and Detection of Camouflaging Worm, IEEE TRANSACTIONS ON DEPENDABLE AND SECURE COMPUTING,,2011,8(4):377-390.

[6] J. Jung, S. E. Schechter, and A. W. Berger. Fast detection of scanning worm infections. In Proceedings of RAID'2004, September 2004.

[7] Yong Tang,Bin Xiao,Xi cheng Lu, Signature Tree Generation for Polymorphic Worms, IEEE TRANSACTIONS ON COMPUTERS,2011, 60(4), 565-579.

[8] Lanjia Wang, Zhichun Li, Yan Chen, Zhi (Judy) Fu, and Xing Li，Thwarting Zero-Day Polymorphic Worms With Network-Level Length-Based Signature Generation ， IEEE/ACM Transactions on Networking，2009，17(5):1-14.

[9] M. Williamson and A. Parry. Virus throttling for Instant Messaging. In Virus Bulletin Conference, vb2004, Sept. 2004.