Misuse Detection of New Malicious Emails

Dong-Her Shih *, Chun-Pin Chang and Hsiu-Sen Chiang
Department of Information Management,
National Yunlin University of Science and Technology,
123, Section 3, University Road, Touliu, Yunlin, Taiwan, R.O.C.

Abstract

A serious security threat today is malicious emails, arriving as email attachments. An email virus is an email that can infect other programs by modifying them to include a replication of it. When the infected emails are opened, the email virus spreads itself to others. Today's society has seen a dramatic increase in the use of emails. We present a novel approach to detect misuse emails by gathering and maintaining knowledge of the behavior of the malicious emails rather than anticipating attacks by unknown assailants. Our approach is based on building and maintaining a profile of the malicious emails through analyzing its static activity. Any new activity of the email is compared to the malicious profile to detect a potential misuse. Comparison results show that our proposed methods outperformed than anti-virus software.

Keywords—email virus detection, self-organizing maps, network security

1. INTRODUCTION

In recent years, the number of Internet users worldwide has continued to rise dramatically as the Internet expands. Within this growth, serious problems such as unauthorized intrusions, denial of service attacks, and computer viruses have arisen. In particular, a computer virus (hereafter, virus) is able to cause damage to a large number of systems because of its ability to propagate. As a result, the power of such attacks can now have a serious impact on an information society. Analysis of reported virus incidents during the five-year period [4] provides interesting insights for anti-virus research, as it reflects a period of rapid uptake in the application of the Internet and the use of e-mail for business purposes. Not surprisingly, there is a substantial increase in the number of incidents reported during the period.

Recently, a kind of virus that can infect executable files has appeared. This kind of virus spreads far more rapidly than before and can propagates by email, one means of information exchange among users. Since distributed systems usually contain objects with heterogeneous security requirements. Existing solutions for distributed access control do not provide the required flexibility and manageability [31]. Email viruses can damage companies, and email virus infection is the majority in infection sources. It is important to understand how serious viruses are and how destructive they can be. A virus which propagates by email and independently of user operation (file copying and email sending) after being infected is called an "email virus" and is distinguished from a conventional virus which propagates as a result of user operation. An email virus is also defined as executables attachment in an email that performs some functions**,** such as damaging a system, transmitting copies of itself by email to other addresses from their address book and inbox automatically. For example, ILOVEYOU is such kind of a case [6]. Besides, some email viruses do not require the email receiver to open the attachment for them to execute. A known vulnerability in Internet Explorer-based email clients (Microsoft Outlook and Microsoft Outlook Express) will automatically executes the file attachment.

Our focus in this paper is primarily on detection of the misuse of malicious emails. Misuse detection systems offer a cost-effective compromise to establishing and assuring a certain degree of security in a system [3]. Our research presents a framework, an email virus filter that can detect malicious Windows attachments by integrating with an email server. Analyzing the characteristic of the email virus, to picked out differentiate one email virus from another. Our goal is to design and build a scanner that accurately detects email virus before they have been entered into a host. The methods discussed in the paper are acting as a network email filter to catch malicious email virus before users receive them through their email.

2. METHOD FOR MISUSE DETECTION

We proposed and implemented an algorithm to build a malicious email profile and detect anomalies in email behavior. Comparing malicious mail's actions against incoming email profile can indicate a potential misuse of malicious email. The proposed process of detecting misuse in malicious email detection is a process that has two distinct stages, which will be described in below.

Step 1: Build the Profile: The first step in detecting misuse is to build a malicious email profile. During this step, profiles are built and stabilized for algorithm based on

all the collected email viruses. A general assumption is made that the collected email viruses are sufficiently large.

Step 2: Test Profile: In the second step, we test the profile to determine the level of misuse by generating a degree of warning. The user's new emails to the detection system are tested against the malicious email profile. A misuse warning is then computed by comparing the difference between the new emails and the malicious email profile based on proposed method. Next, we describe methods, SOM and SOM with k-medoids algorithms, proposed for detecting misuse of an email.

2.1. Self-Organizing Maps

A Self-Organizing Map [13, 14], or SOM, is a neural clustering technique. Having several units compete for the current object performs the SOM clustering. The unit whose weight vector is closest to the current object becomes the winning unit. The weight of the winning unit is adjusted as well as those of its neighbors. SOM assume that there is some topology among the input objects and the unit will take on this structure in space. The organization of these units is said to form a feature map. It is more sophisticated than k-medoids in terms of presentation; it not only clusters the data points into groups, but also presents the relationship between the clusters in a two-dimensional space. SOM is also capable of presenting the data points in one- or three-dimensional space. However, two-dimensional space is most commonly used due to the trade-off between information content and ease of visualization.

2.2. SOM with K-Medoids

In order to find out the boundaries from results of SOM, we applied partitioning method. The most famous and commonly used partitioning methods are k-means and k-medoids, and their variation. The k-means algorithm is sensitive to outliers since an object with an extremely large value may substantially distort the distribution of data. Instead of taking the mean value of the objects in a cluster as a reference point, the medoid (representative object) can be used, which is the most centrally located object in a cluster. Thus, the partitioning method can still be performed based on the principle of minimizing the sum of the dissimilarities (distances) between each object and its corresponding reference point. This forms the basis of the k-medoids method. For example, *PAM* (partitioning around Medoids) [11], built in Splus, starts from an initial set of medoids and iteratively replaces one of the medoids by one of the non-medoids if it improves the total distance of the resulting clustering. *PAM*, use real object to

represent the cluster, works effectively for small data sets, but does not scale well for large data sets.

3. EXPERIMENTAL FRAMEWORK AND RESULTS

In this section we try to investigate some exploration features of email viruses. Every email virus is executed and tested in our lab. By observing and analyzing the behaviors of email viruses that we collected, we find some features that will help users and corporations to prevent from threat of email viruses. The purpose of our work was to explore the possibility of a standard technique to compute accurate detectors for new (unseen) malicious emails. We try to investigate some exploration features of email viruses. Every email virus is executed and tested in our lab. By observing and analyzing the behaviors of email viruses that we collected, we find some features that will help users and corporations to prevent from threat of email viruses. We gathered a large set of emails from both public sources and our own email server. To begin with, the emails were separated into two classes: malicious and benign. There were no duplicate emails in the data set and every email in the set is labeled as either malicious or benign by the commercial virus scanner. The type of gathered malicious email viruses were shown in Table1, which consisted of IW (Internet worms), Trojans, Macros, Scripts, File-infectors, and so on. The contrasting benign data was also collected in order to perform relevance analysis. We extracted a profile from each email in our dataset first, and from the profiles we extracted features to be used in classifiers. Using different features, we trained a set of classifiers to distinguish between benign and malicious emails. Note that the features extracted were static properties of the email and did not require execution.

3.1. Data set

Our data set consisted of malicious and benign clean emails in our UNIX email server. The sampling email viruses were downloaded from various FTP sites and were audited by commercial virus scanner. All these email viruses were discovered from 1999 to 2003. The observed email virus and type of distribution were shown in Table 1. To standardize our data set, we used an updated Norton's virus scanner and labeled our emails as either malicious or benign emails.

Mail format descriptions involve many attributes and analytical characterization was performed. This procedure first removes irrelevant or weakly relevant attributes prior to performing generalization. Finally, from the email format, we extracted a set of features to compose a feature vector for each email as shown in Table 2.

Table 1 Samples of email viruses

Macro	MELISSA.A, GORUM.A
Executable File	ZAUSHKA.A-O, JERM.A, COBBES.A, MAGISTR.B, KAMIL.B, YOUGDOS.A
Trojan	PTWEAK.A, GIFT.B, HYBRIS.C, SIRCAM.A, TROODON.A, XTC.A, FEVER.A
Script	**JavaScript**：GERMINAL.A, EXITW.A, SEEKER.A6, ACTPA.A, EXCEPTION.GEN **VBScript**：REPAH.A, HARD.A, NEWLOVE.A, INFO.A, LIFELESS.A, NOONER.A, KALAMAR.A, CHICK.C, CHICK.B, CHICK.E, LOVELETTER, CHU.A, EDNAV.B, GOOFFY.A, HAPTIME.B, HEATH.A, HORTY.A, VIERKA.B, ARIC.A, GORUM.B, ZIRKO.A
Worm	NIMDA.A-O , ALIZ, PETIK.C, PET.TICK.Q, BADTRANS.B, GIZER.A, ENVIAR.B, GOKAR.A, RADIX.A, UPDATR.A, KLEZ.E, KLEZ.H, LASTWORD.A, LOHACK.A, MERKUR.A, MYLIFE.E, PLAGE.A, PROLIN.A, SOLVINA.B, SHOHO.GEN, DESOR.A, PET.TICK.Q, PETIK.E, ZHANGPO.A, ZOHER.A, SOBIG.A, YAHA.E, BUGBEAR.A, BLEBLA.C, GAGGLE.C

Table 2 Extracted feature from email format

	Feature	Content
X_1	Mail content type	Text/plain/html, other
X_2	Mail size	Total size of email
X_3	MIME Format	Yes, No
X_4	Attachment file no	Number of attachments
X_5	Attachment size	Total size of attachments
X_6	Attachment file type	exe, doc, scr, pif……
X_7	Script language	VBScript, JavaScript…..
X_8	Subject	Re, Fw, Fwd, …
X_9	Carbon Copy	CC, BCC, …
X_{10}	Recipient	Single- Recipient, Multi-

To extract features from data set, we wrote a feature extraction program. The feature extraction program extracts features from Microsoft Outlook email file. In addition, the information was obtained without executing the unknown email but by examining the static properties of the email.

3.2. Misuse detection with SOM and k-medoids

In order to detect unseen new email virus, we are going to incorporating the SOM network in misuse detection first. We build the network structure from training data and use the testing data to measure its performance. The training data and the testing data contain audit events. The training data consists of audit events during malicious activities in the email server. The testing data contains audit events arising from both normal and malicious activities. During training, the structure of SOM network is constructed based on malicious email data. During testing, we compute the accuracy of

SOM network on the evidence of audit events of the testing data. We have used a sample of audit data contains normal activities and malicious activities.

First of all, we use Kohonen's Self-Organizing Map to organize benign email behavior into a two-dimensional map, according to emails' extracted features. Our input vectors consist of a set of malicious emails' features. The desired output is a two-dimensional map of N nodes (in this case, a 9X9 map of 81 nodes). The SOM algorithm has two parameters that change through iterations: the variance of the neighborhood function $\Lambda(n)$ (radial symmetric Gaussian function) and the learning rate $\eta(n)$ (n=1, 2 …). The adaptation laws for these parameters are presented as:

$$\eta(n)=0.9(1-n/1000),\ \Lambda(n)= \Lambda(n-1)(1-0.01n)$$

These parameters adaptation laws lead to a fast convergence of the algorithm without the lost of quality of its output. Using these laws together with the selective update of neurons weighs, there is a reduction of the algorithm complexity through iterations. The selective update consists of a threshold for the neighborhood function that allows only neurons above the threshold to be updated. Since the $\Lambda(n)$ decreases fast with time, so does the number of neurons to be updated.

Figure 1 shows a typical SOM produced by our algorithm with 75 malicious emails in Table 1. The blank nodes contained no emails' mapping, while those labeled with "M" contained malicious emails in step 1 building the profile. The malicious emails are assume clustered in the border of 9X9 grid. Then, by using step 2, the tested profile was shown in Figure 2 with 319 malicious emails and 1701 benign emails. In Figure 2, the "B" nodes contained benign emails and a "+" in the grid means the unspecified node which contained both malicious and benign emails' mapping in the same node. The map contains 2020 emails' behaviors that composed of malicious and benign emails. The distance between nodes on the map indicates the similarity of the emails' behavior, measured according to the features. Similarity here is measured not by the similarity of the content, but by the similarity of behaviors. It is also difficult to quantitatively measure the effectiveness of the SOM. But, by proper choosing the maximum distance of benign training data and its winning node, the malicious emails can be detected. Assume that the distance between malicious data and its winning node is larger than benign. We can calculate the detection rate of audit emails in testing data. Figure 3 shows the results in ROC curve of SOM with 4X4, 9X9 and 15X15 grids. From Figure 3, we can observe that higher detection rare will incorporated a higher false positive rate in SOM. If we can obtain more training data, we may obtain a better result.

Figure 1 Result of the SOM.

It is difficult to find clustering boundaries clearly in SOM. Therefore, we apply the k-medoids clustering after the training of the SOM. Assuming that there are only two clusters in the trained SOM map (malicious or benign). Applying the algorithm (in Figure 4) with 9×9 grids, Figure 4 shows the result of k-medoids clustering for the input vectors in SOM with testing data labeled clearly with “M” and “B”. There are no “+” hexagon in Figure 4, since every node in the map is clearly identified with malicious or benign by using k-medoids clustering. Then, the accuracy of SOM with k-medoids clustering can be calculated and shown in Table 3 incorporated with anti-virus software. With a little false positive rate, our proposed method shows a high detection rate in the testing data.

Figure 2 Result of the testing data

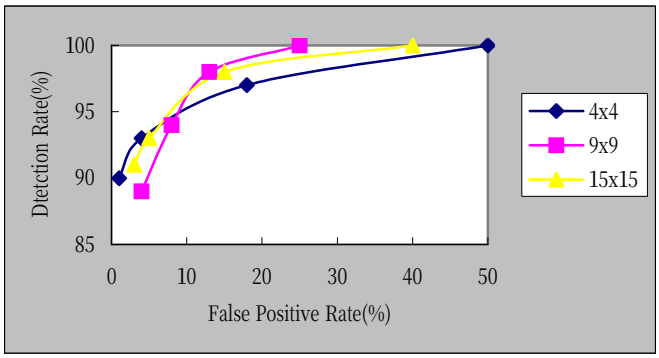

Figure 3 The ROC curves of the SOMs

Figure 4 Result of SOM with k-medoids

Table 3 Comparison results (%)

Profile Type	Detection Rate (%)	F P Rate(%)	Overall Accuracy(%)
SOM(k-medoids)	88.08	1.88	95.53
Pcc2002	41.66	0	88.96
Pcc2003	86.90	0	97.52
Norton2002	30.95	0	86.93
Norton2003	82.14	0	96.62

3.3. Discussion and detection results

A classification method may be compared and evaluated according to the following criteria:

Predictive accuracy: The higher the detection rate incorporated, the higher the false positive rate in our proposed method. A better way, to improve accuracy, is the introduction of the bagged classifier, which counts the votes in every other classifier and assigns the email to the class with the most votes.

Speed: Computation costs for generating a detection model in our proposed methods are almost the same as costs incorporated through other methods. It requires, however, much more effort to derive rules.

Scalability: There is a recursive form in the calculation of a SOM classifier that we did not demonstrate. It seems it takes the most ability to construct the detection model to efficiently function in large amounts of data.

Interpretability: The SOM with k-medoids method has a difficult of understanding in comparison to other methods.

In order to find out the capability of the proposed method in detecting unseen email viruses, we found some new email viruses over the course of our work and tested our methods to see whether the proposed method can unearth them or not. These email viruses were found in our email server after May 2003. Therefore, they are not contained in our data set. Table 4 shows the testing result of anti-virus software and our proposed methods. Note that anti-virus software 2002 is not updated in 2003 and anti-virus software 2003 is not updated until May 2003. Since they are signature-based, we found that there are many email viruses that the anti-virus software 2002 and 2003 could not detect. However, all these email viruses were detected after we update our anti-virus software in March 2004. As a result, our proposed methods outperformed than some available anti-virus software in the detection of new unseen email viruses.

Table 4 Testing results of new email viruses ("√"= detected)

Virus Profile	SOM k-medoids	SOM	Pcc 2003	Norton 2003	Pcc 2002	Norton 2002
NETSKY.C	√	-	-	-	-	-
NETSKY.D	√	-	-	-	-	-
MIMAIL.A	√	√		-	-	-
MYDOOM.A	√	√	-	-	-	-
PE_CIH.1003	√	√	√	√	√	√
WORM_YAHA.G	√	√	√	√	√	√
WORM_BAGLE.C	√	√	-		-	-
WORM_BAGLE.GEN-1	√	√	-		-	-
HTML_BAGLE.Q-1	√	-	-		-	-
WORM_BAGLE.J	√	√	-		-	-

4. MANAGERIAL ISSUES

The standard approach to protecting against malicious email viruses is to use a virus scanner. Because of the email virus has quick dissemination and polymorphisms characteristic, a traditional signature-based method may not be efficient to detect them. Eight to ten viruses are created every day and most cannot be accurately detected until

signatures have been generated for them [30]. During this time period, systems are vulnerable to attacks. Companies should be cautious of email virus threat, and develop their own security plan. We described some managerial issues in this section.

4.1. Myths about email viruses

It is surprising that many managers fear email virus threat, but lack a good understanding of risks and controls related to various security technologies. Following are some myths associated with email virus security.

●*Myth 1: "I did not open the email attachment, so my email is safe."* Individuals do not understand that receiving email would infect your computer even if you did not open the attachment. MELISSA and LOVELETTER are two examples of such cases.

●*Myth 2: "I use a wireless connection, so my email is safe."* It is true that when using a wireless connection, each individual is assigned a different IP address on each connection. This will make it harder for the hacker to find your computer and browse your contents. However, some Trojan email viruses, such as FEVER and TROODON, are independent of IP address. Therefore, there is no security guarantee in wiring environments. Individuals who stay connected for a long period of time are at risk of being "Trojaned".

●*Myth 3: "I use an antivirus software, so my email is safe."* Antivirus software can protect a computer from email viruses, but may not protect it from newly just-released viruses. In addition, antivirus software will offer no protection against hackers.

●*Myth 4: "I use a firewall, so my email is safe."* Firewalls do provide added security, but they will not provide protection against email viruses or an insecure computer.

Although understanding myths can correct our mistakes, managers and individuals must also familiar with the characteristic of email viruses and general email protection strategies.

4.2. Email virus protection strategies

Viruses are consistently ranked as one of the most frequent security threats in organization and virus protection has become a major business. Therefore, understanding some aspects of protection are needed. From the above section, we can learn some protection strategies.

1) Use anti-virus software and update regularly.

2) Use advanced anti-virus techniques.
3) Self-protection.
4) Rendering email unreadable.
5) Patch (Update) your operating system and application software regularly.
6) Caution necessary for infection through browsing a web page link in email.
7) Block risky attachments on email servers and gateways.

If all tools do not protect email properly, what is a manager to do? Since not one technique will provide complete protection, developing an email security plan is probably the best way to protect our emails privacy.

5. CONCLUSION AND FUTURE WORK

The contribution that we presented in this paper was a method for detecting different type of malicious emails included Macro and VBScript's attachments. We have presented a detection model that utilizes data mining methods to organize email virus in a domain to detect. Clearly the proposed method has generated clear clusters. The result of this system is very meaningful and can be easily incorporated with an email server to assist detection of malicious emails. Noted that the features extracted were static properties of the email and did not require executed. Operating from an email server, the proposed method could automatically filter the email each host receives. All of this could be done without the server's users having to scan attachments themselves or having to download updates for their virus scanners. Furthermore, its evaluation of an attachment is based solely on the behavior of the email and not the contents of the attachment itself. That added the ability to detect both the set of known malicious emails and a set of previously unseen, but similar malicious emails. As a result, companies must take even more precautions to guard against the introduction of email viruses into their systems. This paper also stated how companies can protect their emails from the intrusion of email virus and how to develop their own email virus security plan.

Virus Scanners are updated about every month. 240–300 new malicious executables are created in that time (8–10) a day [30]. During this time period, systems are vulnerable to attacks. Our method may catch those new malicious emails without the need for an update. Stopping the malicious viruses from replicating on a network level would be very advantageous. Certainly the prototype has been a useful tool for internal purposes, and it is likely to be a useful approach to assist other organizations in better understanding the interests of malicious email virus. One of the most important

areas of future work for this application is the development of more efficient algorithms. The current methods require a machine with a significant amount of memory to generate, and employ the classifiers. Another potential future work of proposed method is to make it into a stand-alone virus scanner and to port the algorithms to different operating systems. Finally, our future research will be investigating the scalability of the system, so that it can be incorporated with other detection models.

REFERENCES

1. David R. Anderson, Dennis J. Sweeney and Thomas A. Williams, Statistics for Business and Economics, 1990, West, San Francisco, pp81~82.

2. L. Bridwell, ICSA Labs' Eighth Annual Virus Prevalence Survey 2002, 2002. Available at http://www.icsalabs.com/2002avpsurvey/index.shtml

3. Christina Yip Chung, Michael Gertz, and Karl Levitt, Demids: A misuse detection system for database systems, In Third International IFIP TC-11 WG11.5 Working Conference on Integrity and Internal Control in Information Systems (1999), 159–178, Kluwer Academic Publishers.

4. Coulthard and T.A. Vuori, Computer viruses: a quantitative analysis, "Logistics Information Management" , 2002, vol.15, no.5/6, pp400-409.

5. R. Crawford, P. Kerchen, K. Levitt, R. Olsson, M. Archer and M. Casillas, "Automated Assistance for Detecting Malicious Code", Proceedings of the6th International Computer Virus and Security Conference, 1993.

6. L. Garber, "Melissa Virus Creates a New Type of Threat", Computer, 1999, Vol 32 No 6, pp.16 –19.

7. G. E. Gorman and B. J. Corbitt, "Core competencies in information management education", New Library World, 2002, Vol 103 No 11, pp.436-445.

References are upon request.

Acknowledgement: The author would like to give thanks to the National Science Council of Taiwan for grant NSC- 92-2218-E-224-015 to part of this research.