Misuse Detection of Email Viruses base on SOM with k-medoids

Dong-Her Shih, Sheng-Fei Hsu, Hsiu-Sen Chiang, and Chun-Pin Chang

National Yunlin University of Science and Technology, Department of Information Management, 123, Section 3, University Road, Touliu, Yunlin, Taiwan
{shihdh,g9320803,g9023728,g9123728}@yuntech.edu.tw

Abstract. 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. We propose 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. 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.

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 [1] 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.

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. As shown in [2], the email vector again took an upward turn. Virus disasters and incidents have organizational ramifications beyond the money, resources, and effort required to recover from such incidents.

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.

© A. Gelbukh, R. Monroy. (Eds.)
Advances in Artificial Intelligence Theory
Research on Computing Science 16, 2005, pp. 139-148

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 Related Work

Detecting malicious executables is not a new problem in security. Protection from unknown viruses is, indisputably, the issue of the day in computer virology. Early methods used signatures to detect malicious programs. Current approaches are matching them to a set of known malicious programs. Experts were typically employed to analyze suspicious programs by hand. In an attempt to solve this problem, the anti-virus industry generates heuristic classifiers by hand [4]. Using their expertise, signatures were found that made a malicious executable example different from other malicious executables or benign programs. One example of this type of analysis was performed by Spafford [5]. He analyzed the Internet Worm and provided detailed notes on its spread over the Internet, the unique signatures in the worm's code, the method of the worm's attack, and a comprehensive description of system failure points. Malicious code is usually classified [6] into the following categories: Viruses, Worms, Trojan horses, Back doors, Spyware, Attack scripts, Java attack applets and Dangerous ActiveX. Combining two or more of these malicious code categories can lead to powerful attack tools. For example, a worm can contain a payload that installs a back door to allow remote access. When the worm replicates to a new system (via email or other means), the back door is installed on that system, thus providing an attacker with a quick and easy way to gain access to a large set of hosts. Once the back-door tool gains a large installed base, the attacker can use the compromised hosts to launch a coordinated attack, such as a distributed denial-of-service (DDoS) attack [7]. At IBM, Kephart and Arnold [8] developed a statistical method for automatically extracting malicious executable signatures. Their research was based on speech recognition algorithms and was shown to perform almost as good as a human expert at detecting known malicious executables. Their algorithm was eventually packaged with IBM's anti-virus software. Lo *et al.* [9] presented a method for filtering malicious code based on "tell-tale signs" for detecting malicious code. Similarly, filters for detecting properties of malicious executables have been proposed for UNIX systems [10] as well as semiautomatic methods for detecting malicious code [11]. As is well known, the main deficiency of commonly used anti-virus scanners is that these scanners are able to detect and delete only those malicious programs described in their anti-virus databases. The anti-virus community relies heavily on known byte-code signatures to detect malicious programs. The main disadvantage to this approach is that the scanner cannot detect a virus if the database doesn't contain its description and the user is unprotected from this new virus threatening during this time period.

Previous work done on detection can be divided into anomaly detection and misuse detection. Anomaly detection deals mainly with attack to the system from abuse. Misuse detection deals mainly with the attack to the system by an authorized user who is misusing his/her privileges. Misuse detection has generally been employed to complement the shortcomings of other prevention techniques [3]. Prior work on

misuse detection has been mainly focused on using logs and user profiles. Profile-based detection systems audit the deviation of user activities from normal user profiles. In the past, a user's command history has been reviewed based on the percentage of commands used over a specific period of time. The logs are then scanned and mined [12] for interesting temporal and sequential patterns about a user's activity [13]. A successful misuse detection system must overcome many challenges. First of all, a user's profile may change over time. To handle dynamic profiles, learning algorithms are required to track user behavior and adapt to a dynamically changing concept. Chung et al. in [3] describe their misuse detection system, DEMIDS, for database applications. DEMIDS uses the access information of the user to the database tables, columns, and other structures to build a user profile to track the behavior of the user.

In the specific area of malicious email detection, however, we are unaware of published work directly relating to user query profile learning and abnormal query behavior detection. We implement a misuse detection warning by comparing user's email behavior to malicious email, learned through clustering, thus creating a new dimension to profile-based misuse detection.

Our method is different from the previous researches because we analyzed the entire features of malicious email instead of only boot-sector viruses or only Win32 binaries. Our technique is similar to data mining techniques that have already been applied to Intrusion Detection Systems by Lee *et al.* [14]. We applied a similar framework to the problem of detecting new malicious email viruses but included macros and script those Schultz *et al.* [15] and other researchers didn't provide.

3 Methodology

A Self-Organizing Map [16], 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. SOM is also capable of presenting the data points in one- or three-dimensional space. 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) [17], 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. The k-medoids algorithm is more robust than k-means in the presence of outlier and noise because a medoid is less influenced by outliers or other extreme values than a mean. Therefore, we apply the k-medoids clustering to results of SOM. The procedure of the presented method is the training of the SOM and then applying the k-medoids clustering. The input vectors of the SOM are the benign email data and the input vectors of the k-medoids clustering are the prototype vectors of the SOM. The outline of the proposed method is shown in Figure 1.

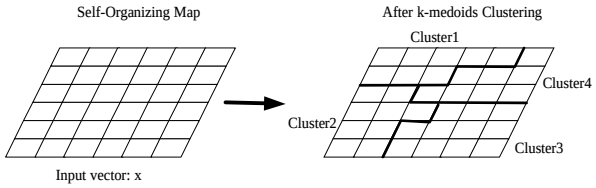

Fig.1. The outline of the proposed method

4 Experiment 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.

To evaluate the performance, we computed the false positive rate (FP rate) and the detection rate. The false positive rate is the number of benign emails that are mislabeled as malicious divided by the total number of benign emails. While the detection rate is computed as the number of malicious emails correctly classified divided by the total number of malicious emails tested. The overall accuracy of the algorithm is calculated as the number of emails the system classified correctly divided by the total number of emails tested.

4.1 Data set

We gathered a large set of emails from both public sources and our own 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. To standardize our data set, we used an updated Norton's virus scanner and labeled our emails as either malicious or benign emails. We statically extracted different features that represented different information contained within each email. These features were then used to generate detection models. The sample emails were

separated into two classes: malicious and benign. Our data set consisted of a total of 2,203 programs split into 344 malicious emails and 1,859 benign emails. 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 Table 1, 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. 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.

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:** EXCEPTION.GEN，GERMINAL.A，EXITW.A，SEEKER.A6，ACTPA.A **VBScript:** REPAH.A，HARD.A，NEWLOVE.A，INFO.A，CHU.A，GOOFFY.A，NOONER.A ， ARIC.A ， KALAMAR.A ， CHICK.C ， CHICK.B ， CHICK.E ，ZIRKO.A，LOVELETTER，EDNAV.B，HAPTIME.B，HEATH.A，HORTY.A，VIERKA.B，GORUM.B，LIFELESS.A
Worm	NIMDA.A-O ， GOKAR.A ， ALIZ ， PETIK.C ， PET.TICK.Q ， BADTRANS.B ，GIZER.A ， BUGBEAR.A ， ENVIAR.BR ， ADIX.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，BLEBLA.C，GAGGLE.C

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 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-

4.2 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. 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 9×9 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.

Neural Networks are also sensitive to the number of neurons in their hidden layers. Too few neurons can lead to underfitting. Too many neurons can contribute to overfitting, in which all training points are well fit, but the fitting is uncertain about testing point. We adjust some parameters (neurons and attractive radius) through some experiments to make SOM have better performance. Using False positive rate and detection rate, to evaluation the performances of the SOM in various experiment designs status. We can calculate the detection rate of audit emails in testing data and gain the best performance, when use a 9×9 map of 81 nodes and the scale 5 of the attractive radius. The over accuracy rate of the different scale radiuses are presented in Table 3. Figure 3 shows the results in ROC curve of SOM with 4×4, 9×9 and 15×15 grids. 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.

Table 3. The over accuracy rate(%) of the different radiuses in the same neurons

	Attractive radius		
Map	Scale 3	Scale 5	Scale 7
9×9	98.28%	98.28%	98.28%
4×4	98.64%	98.64%	98.28%
15×15	98.23%	98.37%	98.18%

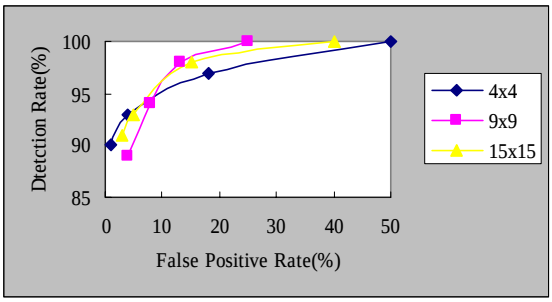

Fig. 2. The ROC curves of the SOMs

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 with 9×9 grids, Figure 3 shows the result of k-medoids clustering for the input vectors in SOM with testing data labeled clearly with "M" and "B".

Fig. 3. Result of SOM with k-medoids

Current most virus scanners technology use signature-based detector that detect new viruses [18]. The classic signature-based detection algorithm relies on signatures (unique telltale strings) of known malicious executables to generate detection models. Signature-based methods create a unique tag for each malicious program so that future examples of it can be correctly classified with a small error rate. These methods do not generalize well to detect new malicious binaries because they are created to give a false positive rate as close to zero as possible. Whenever a detection method generalizes to new instances, the tradeoff is for a higher false positive rate. Then, the

accuracy of SOM with k-medoids clustering can be calculated and shown in Table 4 incorporated with anti-virus software. With a little false positive rate, our proposed method shows a high detection rate in the testing data.

Table 4. 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

One of the primary problems faced by the virus community is to devise methods for detecting new virus that have not yet been analyzed. Eight to ten viruses are created every day and most cannot be accurately detected until signatures have been generated for them [4]. During this time period, systems protected by signature-based algorithms are vulnerable to attacks. In order to find out the capability of the proposed method in detecting unseen email viruses, we found some new email viruses and tested our methods to see whether the proposed method can unearth them or not. These email viruses were found after May 2003. Therefore, they are not contained in our data set. Table 5 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 October 2004. As a result, our proposed methods outperformed than some available anti-virus software in the detection of new unseen email viruses.

Table 5. Testing results of new email viruses (*"√"= detected*)

Virus Profile	SOM with k-medoids	SOM	Pcc2003	Norton2003	Pcc2002	Norton2002
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	√	√	-	-	-	-

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. 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. 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.

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.

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.

References

1. Coulthard, A., Vuori, T.A.: "Computer viruses: a quantitative analysis", Logistics Information Management, vol.15, no.5/6, (2002) 400–409
2. Bridwell, L.: ICSA Labs' Eighth Annual Virus Prevalence Survey 2002 (2002). Available at http://www.icsalabs.com/2002avpsurvey/index.shtml
3. Chung, C. Y., Gertz, M., and Levitt, K.: 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. Gryaznov, D.: Scanners of the Year 2000: Heuristics, Proceedings of the 5th International Virus Bulletin (1999)
5. Spafford, E. H.: The Internet worm program: an analysis, Tech. Report CSD–TR–823, Department of Computer Science, Purdue University (1988)
6. McGraw, G., Morrisett, G.: "Attacking malicious code: Report to the infosec research council", IEEE software, vol.17, no.5, (2000) 33–41
7. Michie, D., Spiegelhalter, J., Taylor, D. C. C.: Machine learning of rules and trees. In Machine Learning, Neural and Statistical Classification, Ellis Horwood (1994)
8. Kephart , J. O., Arnold, W. C.: Automatic Extraction of Computer Virus Signatures, 4th Virus Bulletin International Conference (1994) 178–184

9. Lo, R.W., Levitt, K.N., Olsson, R.A.: MCF: a Malicious Code Filter, Computers & Security, vol.14, no.6, (1995) 541–566
10. Kerchen, P., Lo, R., Crossley, J., Elkinbard, G., Ollson, R.:" Static Analysis Virus Detection Tools for UNIX Systems", Proceedings of the 13th National Computer Security Conference (1990) 350–365
11. Crawford, R., Kerchen, P., Levitt, K., Olsson, R., Archer, M., Casillas, M.: Automated Assistance for Detecting Malicious Code, Proceedings of the6th International Computer Virus and Security Conference (1993)
12. Ling, C.X., Gao, J., Zhamg, H., Qian, W., Zhang, H.: Improving encarta search engine performance by mining user logs, International Journal of Pattern Recognition and Artificial Intelligence (IJPRAI) 16 (2002) no. 8.
13. Sleznyov, A., Mazhelis, O.: Learning temporal patterns for anomaly intrusion detection, Symposium on Applied Computing (2000) 209–213
14. Lee, W., Stolfo, S., Mok, K.: A Data Mining Framework for Building Intrusion Detection Models, IEEE Symposium on Security and Privacy (1999)
15. Schultz, M. G., Eskin, E., Hershkop, S., Stolfo, S. J.: "MET: An Experimental System for Malicious Email Tracking", in Proceedings of the 2002 New Security Paradigms Workshop, NSPW-2002, Virginia Beach, VA: September 23rd - 26th (2002)
16. Kohonen, T.: The Self-organization maps, Proc. IEEE, vol.78, no.9, (1990) 1480–1481
17. Kaufman, L., Rousseeuw, P. J.: Finding Groups in Data: an Introduction to Cluster Analysis, John Wiley & Sons (1990)
18. White, Steve R., Swimmer, Morton, Pring, Edward J., Arnold, William C., Chess, David M., and Morar, John F.,: Anatomy of a Commercial-Grade Immune System, BM Research White Paper (1999). Available at: http://www.av.ibm.com/ScientificPapers/White/Anatomy/anatomy.html.