Research Article Open Access

Modeling Web Security Analysis Attacks with CySeMoL Tool

Abbas A. Abdulhameed[1*], Razi J. Al-Azawi[2], Basil Al-Mahdawi[3]

1 Department of Computer Science, College of Science, Mustansiriyah University, Baghdad, IRAQ
2 Department of laser and Optoelectronics Engineering, University of technology, Baghdad, IRAQ
3 Department of Mechatronics, Engineering Middle Technical University, Baghdad, IRAQ

*Correspondent contact: abbasabdulazeez@uomustansiriyah.edu.iq

ArticleInfo

Received
05/05/2020

Accepted
07/06/2020

Published
20/08/2020

ABSTRACT

The utilize of the web has made humans and companies powerless to exterior assaults. Indeed, cyber problems essentially influence information frameworks with distinctive types of malicious attacks such as spyware, virus, social engineering, etc. The Internet e-mail service, in particular, has become one of the most dependable methods of communication among people, institutions, and companies. The development of digital signatures to e-mail services has raised the e-mail security, which led to replacing the standard mailing of registered letters. Unfortunately, the process of sending and receiving e-mails has created a negative impact means on security and privacy from cybercriminals by diffusing spam and malware. As a result, e-mail hosts are constantly under attack by malicious programs that are often attached to e-mails. In this paper, the simulation model and prototype of an email traffic monitor developed and tested in order to prove the ability of our proposed method for detecting new viruses. This paper states the success possibility of this new method based on the simulation results. The results of the analysis suggest that the Cyber Security Modeling Language (CySeMoL) model has a good performance of operating system vulnerability prediction. At last, some useful suggestions in the context of the CySeMoL model are presented.

KEYWORDS: Virus detection; CySeMoL; Traffic monitor; Web security; Simulations; Cybersecurity.

الخلاصة

أدى استخدام الأشخاص والشركات والمؤسسات إلى الويب إلى جعل الأجهزة قادرة على شن هجمات خارجية. في الواقع ، تؤثر المشاكل السيبرانية بشكل أساسي على أطر المعلومات ذات أنواع مميزة من الهجمات الخبيثة مثل برامج التجسس والفيروسات وما إلى ذلك. أصبحت خدمة البريد الإلكتروني عبر الإنترنت ، على وجه الخصوص ، واحدة من أكثر طرق التواصل الموثوقة بين الأشخاص والمنظمات والشركات. أدى تطوير التوقيعات الرقمية لخدمات البريد الإلكتروني إلى رفع مستوى أمان البريد الإلكتروني ، مما أدى إلى استبدال البريد القياسي للرسائل المسجلة. لسوء الحظ ، أثرت عملية إرسال واستلام رسائل البريد الإلكتروني سلبًا على أمان وخصوصية المجرمين الإلكترونيين من خلال نشر البريد العشوائي والبرامج الضارة. ونتيجة لذلك ، يتعرض مضيف البريد الإلكتروني للهجوم باستمرار من قبل البرامج الضارة التي غالبًا ما يتم إرفاقها برسائل البريد الإلكتروني. في هذا البحث ، تم تطوير واختبار نموذج المحاكاة والنموذج الأولي لمراقب حركة مرور البريد الإلكتروني لإثبات قدرة طريقتنا المقترحة للكشف عن الفيروسات الجديدة. توضح هذه الورقة إمكانية نجاح هذه الطريقة الجديدة بناءً على نتائج المحاكاة. تشير نتائج التحليل إلى أن نموذج CySeMoL لديه أداء جيد للتنبؤ بهشاشة نظام التشغيل. أخيرًا ، يتم تقديم بعض الاقتراحات المفيدة في سياق قالب CySeMoL.

INTRODUCTION

The electronic information infrastructure currently performs an inescapable characteristic in interactions with mankind. With developing dependence on on-line mailing resources, there is a collimated growing number of vulnerabilities and attacks on this service. As a result, growing demand for security solutions [1, 2]. Despite the important role of email in the everyday life of a typical Internet user, email is not sufficiently well secured [3, 4]. The rapid proliferation of the Internet communication medium has led to the emergence of widespread abusive behaviors by certain individuals. According to several surveys, unsolicited marketing messages and spams form more than half of the total daily message traffic [5].

A computer virus is a specific type of malware designed to replicate and spread by modifying other computers programs, and inserting its own code. When this replication succeeds, the affected programs are referred to as “infected” with a computer virus. The sole purpose of any virus is to replicate and spread itself, and damage to the host system is often a consequence of infection.

The e-mail viruses in general is a specific type of malware designed to replicate itself and make a large volume of e-mail traffic over time to overwhelm e-mail servers and client network resources, leading to a serious disruption in the use of e-mail services. While most of the related studies classify all self-propagating malicious code as viral and do not differentiate the methods of infection (email, network exploit, physical media, etc.), mass-mailing worms (those that spread via infected emails) are one of the more malignant variants of malicious code [6].

A large number of anti-virus product software, there are various methods developed by companies and organizations that detect and neutralize e-mail virus damage, but only four main methods have been widely used: scanners, heuristics, behavior blocks, and integrity checker. Are: scanners, integrity checkers, heuristic analysis, and behavior block.

Deficiencies in identifying and blocking email viruses may be very severe and have important implications. For explanations, the virus I love you, usually named "Love Beetle," is a crossbred of an email virus and the worm. Thus, the resulting commercial damage alone will be from five to billions of damages worldwide [7].

Reproduction of an e-mail virus generates a giant quantity of web site visitors and raises the load on the mail servers. Mail viruses act as a cyber-attack via pulling databases and mail servers [8]. Recently, if the vaccination was applied as demonstrated by research and analytical studies on specific computer nodes in the network, this can effectively analyze the number of infected computers and the infection rate [9].

The cost of damage can be significantly reduced by the damage caused by e-mail viruses. In the case of early detection of attacking email viruses through an antivirus program, identifying and hindering the spread of email viruses in the mail servers is the new approach presented in this paper. To reveal whether e-mail viruses can be detected through simulation and by monitoring e-mail messages that pass-through mail servers, a prototype network was built and operated to monitor e-mail traffic. In Section 2, we present the related work. Section 3; provides a background on the concepts used in this study. In Section 4, we present our overall approach for the verification of CySeMoL functional requirements and explains the extraction simulation environment to validate properties from the CySeMoL requirements. Section 5 illustrates our verification approach in a case study, which considers the techniques for detecting a virus’s system. Finally, in Section 6, we conclude and outline some ideas for future work.

RELATED WORK

Having effective methods for detecting traditional and new viruses is an essential step to counter virus sabotage. Software organizations and companies specialized in computer network protection It has been detected. Many of the innovative ideas developed and which have been used to detect possible menace of viruses. Recently, two of the new methods for detecting viruses have been described. The method; “data-mining Methods for Detecting New Malicious Executable Files,” this method describes a way of how artificial intelligence may detect viruses. Each of these learning algorithms are able of extract executables malicious and can generate the required rule sets for detecting the corresponding viruses [10].

The detection of a large number of known viruses is an indication of the success of the data mining approach [11]. The second method, Balzer, developed a sort of e-mail wrapper to detect the viruses that come within the e-mail attachments. This method focuses on the email attachment, where most viruses spread by email are sent as email attachments. The aggregation method provides runtime monitoring and authorization to ensure the safe implementation of the content and to prevent any suspension of malicious behavior.

Through the intermediation of interfaces that operations use to access and modify network resources, monitoring and authorization takes place [12]. Thus, by following specific rules, the violation process can be detected by the wrapper detection. Users are permitted to attend or attend infringing operations when the wrapper notifies users in case of violation of these rules.

The next section of this paper describes the methodology of the proposed virus detection method, which is used to monitor the e-mail traffic [13].

Several tools exist that focus on the security analysis of detection. Some tools such as ADVISE [14], CyberSage [15], and CySeMoL [16, 17] Centering on a probability analysis of an attacker reaching the attacker goal. These systems do not provide suggestions to improve the security of the system. ADVISE requires profound security knowledge, since the assessor needs to provide the attack tree as input. The effects of the evaluation are largely dependent on the quality of the provided It was attacking the tree. The output of CySeMoL is similar to ADVICE, but does not rely on an attack tree provided by the assessor. Alternatively, generic blocks are defined that allow the assessor to model the detection virus system.

This reduces reliance on the security setting of the assessor. An important drawback, even so, is that CySeMoL uses a fixed attacker model, decreasing the flexibility for the assessor to analyze the security with respect to different types of aggressors. In CyberSage, the assessor must provide the workflow. A series of attack steps to arrive at the attacker’s goal. This also calls for a significant security background and does not consider alternative attack paths. Other tools focus on compliance with standards.

BACKGROUND

In this section, we introduce the virus detection that will be used in our approach to describe the system and then use formulas in CySeMoL model simulation environments to specify the requirements of the software viruses, and worm creators highly prefer electronic messages.

The Safety of Email

Data confidentiality, authentication, integrity, non-repudiation, access control, and accessibility are the most important security criteria security services that should be carefully considered when creating software applications and systems. [9]. Within conventional e-mail systems, though, there is little allowance for these surveillance agencies, because the e-mail is exposed to both negative and positive attacks. The passive attack risk leads to the launch of message contents and site visitor’s analysis, whilst the energetic attack risk consists of amendment of message contents, Masquerade, Replay, and Denial-of-Service (DoS) [18] All of these mentioned threats may interfere with the traditional e-mail protocols as follows:

1. Knowledge exposure: Most emails are usually sent in transparent (not encrypted) form. In fact, people other than the actual clients may monitor the contents of electronic mail by utilizing our open changing software.
2. Traffic analysis: Some countries automatically display the e-mail messages of positive customers as the phase of their echelon project. This exercise is now not simply for counterterrorism reasons, however, there are different motives such as facilitating the war in opposition to industrial espionage and for carrying out political eavesdropping [8]. This is not, however, exclusively devoted to national governments, provided that there is an increasing market in supplying knowledge via e-mails to commercial and criminal products.
3. Message Modification: Email content may be modified during transport or storage. In this case, an attacker in the middle does not necessarily need to manage the gateway because an attacker located on the same local area network (LAN) can use an address resolution protocol (ARP) spoofing tool such as ettercap to intercept or modify all mail packets sent to or from the mail server, or the gateway.
4. Masquerade: may submit an email to another citizen or company's name.
5. Play previous message: Other recipients can receive previous messages. That can contribute to injury, misunderstanding, or risk to an employee or institution's image. This will result in further disruption if emails are used for other purposes, such as the flow of money, authentication or booking.
6. Spoofing: Anyone can insert incorrect messages into the content of another user's

profile. Using malware this can be performed from a local network or from an external network.

7. Denial of service: By overloading it with mailboxes this attack will disable the mail system. This can be provided using Trojan horses or malware in the details of emails sent to clients. By re-entering the wrong login codes, you will even block out a user profile.

The CySeMoL

Cyber Security Modeling Language (CySeMoL) [17] is a complex formalization for assessing cybersecurity on information system architectures. CySeMoL provides modeling capabilities in the Unified Modeling Language (UML) and OMG System Modeling Language (OMG SysML) [19] with the Bayesian attack graph realization of the object constraint language (OCL).

The output of CySeMoL is a heatmap that is based on a class diagram that indicates how difficult it is for an attacker to access the various architecture resources and provide them with a specific entry point or even multiple entry points. Because the attacker has the probability to be connected at any node in the infrastructure, the attack scenarios can be modeled according to a certain number of possible situations.

The CySeMoL is based on experiences obtained from specialists in the sector and observational research. This language includes numerous architectural elements from both domains of software and hardware. Among these elements, the computer, network equipment, operating systems, web servers, firewalls, network interfaces and much more. The attributes of the metaclass in this language describe the attack steps and countermeasures for these attack steps. The attack steps represent part of the attack scenarios that are performed to achieve attack intentions. Countermeasures are defined as countermeasures to block the achievement of attack intentions. While the present CySeMoL attack targets are in an attack, trees can be defined [20].

THE PROPOSED METHODOLOGY

CySeMoL Meta-Model

The CySeMoL classification is defined in terms of the underlying UML on which CySeMoL is based, and it is done using UML through the CySeMoL metamodel. A CySeMoL metamodel is structured by using modules that are derived from the base class “Attacker”. With the exception of the attacker, all concepts are described in the same way. First, some overall information about the concept is outlined. Second, all possible connections for the concept are described using both text and figure. Third, the attack steps and defenses of each concept are described in a table. Fourth, the attack steps and defenses corresponding to the concept are described in depth.

Each attack step and defense are also described in a predictable manner first, and overall information about it is presented. Then, the quantitative logic corresponding to it is described. One or more processes define the behavior of a module: as can be seen in Figure 1. All building blocks for the module are hierarchically derived from the base class.

DATA COLLECTION METHOD

Several input simulation parameters, where each one has specific characteristics with certain effects, in addition to the procedure followed, may have a signification impact on the simulation output results. In our work, we will depend on running software simulations of two types, namely, a control simulation and a virus-contaminated simulation. In the control simulation type, there will be no e-mail viruses in the following procedure. In the type of simulation infected with viruses, various types of viruses will be imposed on the network under consideration.

Definition of Model

Throughout this part, all views of the model are characterized and modeled using the CySeMoL tool. View consists of various categories of assets entities, including such firewalls, application servers, mobile tools, web browsers, password authentication mechanisms, etc. The relationship between the assets is explained and shown in Figure 2.

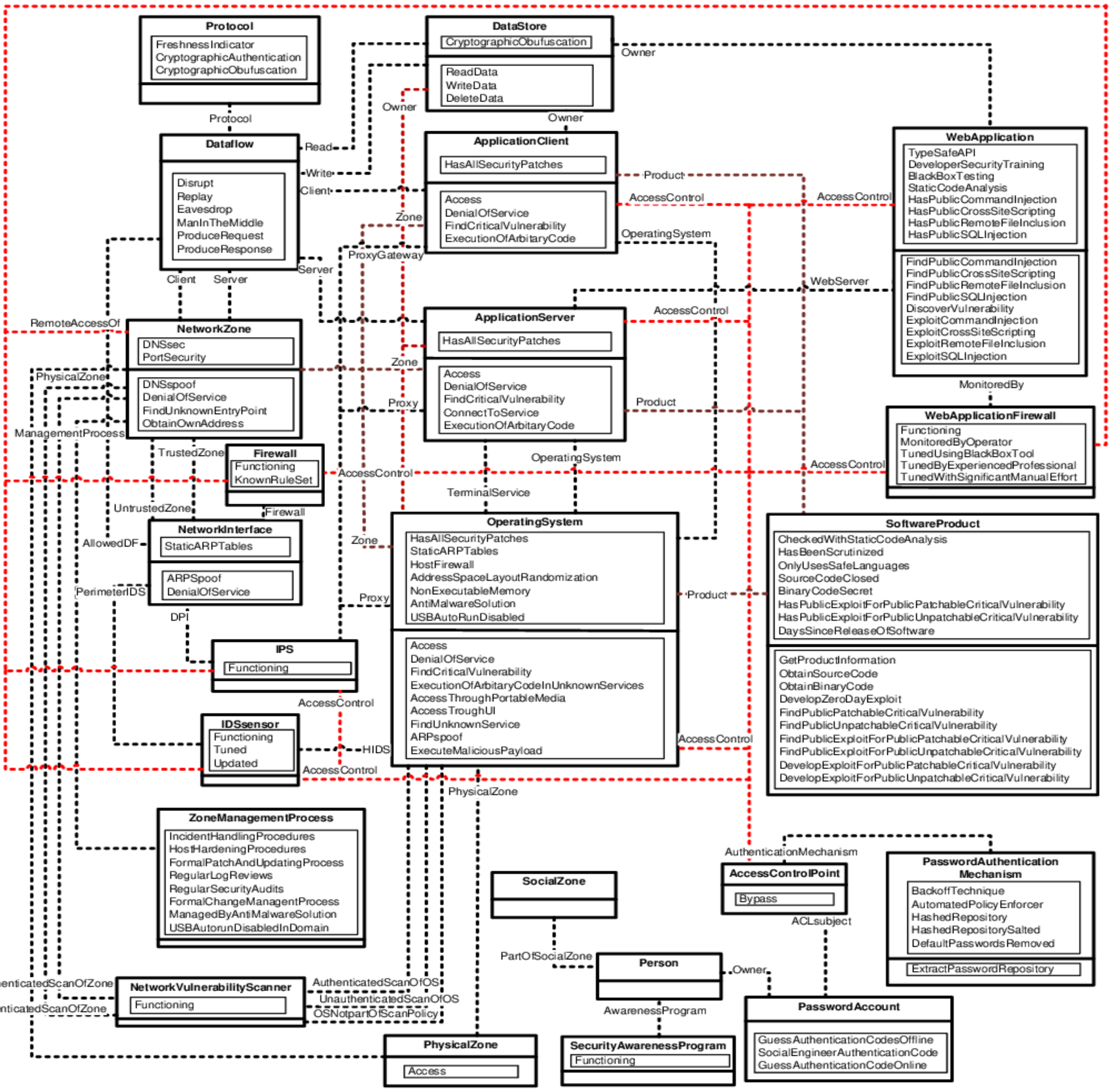

Figure 1. The Metamodel to CySeMoL.

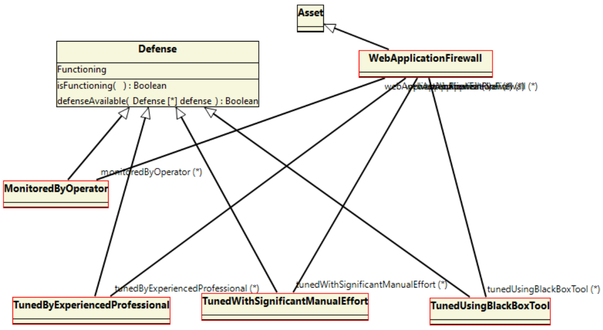
Asset
Defense
Functioning
isFunctioning( ) : Boolean
defenseAvailable(Defense [*] defense) : Boolean
WebApplicationFirewall
MonitoredByOperator
monitoredByOperator (*)
TunedByExperiencedProfessional
tunedByExperiencedProfessional (*)
TunedWithSignificantManualEffort
tunedWithSignificantManualEffort (*)
TunedUsingBlackBoxTool
tunedUsingBlackBoxTool (*)

Figure 2. Model View based Web Application Firewall.

Input Variables

These variables represent the number of nodes in the concerned network, which are in fact, are the number of computers of the network under consideration in this paper.

- Monitor Virus Scan Interval: This number gives us an indication of how often the e-mail traffic monitor will run the scan for viruses.
- Number of viruses present. This number shows how often viruses exist on the network and indirectly determines email activity.
- Monitor Range: The number of nodes (we mean computers here) that can be monitored by the mail traffic monitor. As the workload increases on the screen for a wide range of screens.

- Monitor Score: As far as e-mail virus activities are concerned, the sensitivity of e-mail traffic monitors should determine this number. The more sensitive you are to monitoring email virus activity, the lower the number.

Monitor Simulation

This control simulation is to be carried out while e-mail viruses are not present. Here we use the simulation to be an exercise on power. Every virus warning produced by the remote monitoring system will not be considered and, in the absence of this control experiment, will be viewed as a false alarm. Accordingly, the main goal of the control experiments is to determine whether the traffic monitor will give false virus alerts or not. Therefore, simulations must be conducted whit in two different virus environments:

- Unique Virus: The simulations will be carried out here in an area involving a specific virus.
- Multi-virus: The simulations here will be conducted in an environment with the presence of multiple types of e-mail viruses.
- The results of each simulation are stored in two files:
- Log.txt: This file records some of the email activities. It will record certain information of e-mails that have large attachments size.
- VirusAlert.txt: This file contains an alert about every virus that creates an email traffic monitor

Simulation Results

Multiple forms of simulations being implemented: representations of viruses, and simulations of function. These two styles of simulations were carried off the same preliminary setting but with the exception that there were no email viruses in the control simulation.

Control Simulations

Implemented Six observational simulations where each simulation began with fifty nodes. As clients (email users) forty-nine nodes are assigned, and the last node is a server (email server). Testing must create a warning system if a plant of the encrypted mails with a ranking greater than 28 can be successfully constructed. This is anticipated that there will be no virus warning when setting the current simulation. Each client has a different possibility of getting an email virus. It also has a random chance to compose and reply e-mail messages. The virus will change the behavior of the client when infected by a virus, according to the characteristics of the virus. As explained above, all control current simulations were carried with the same initial settings. Out of the seven simulations, one of these controller simulations generated a false alarm alert screen. In these control simulations, the false alert rate was approximately 15 %. The results of the simulation control are shown in Table 1. And Figure 3.

Table 1. Control simulation.

Configuration for Network				
Number Of Clints	Score	Domain Checking	Monitor Domain	Case
50	30	4	4	No virus

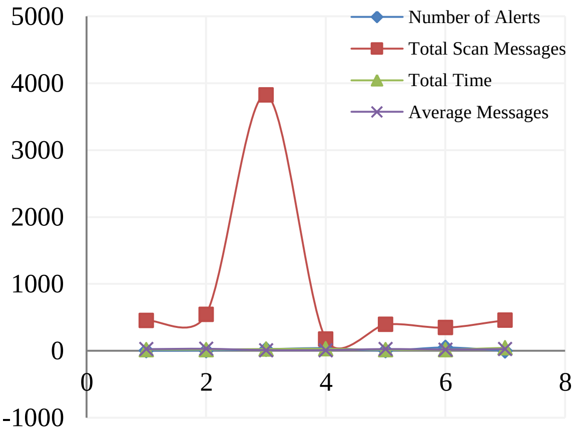

Figure 3. The Behavior of a Simulation.

Designed to simulate Viruses

Infected viruses brought models to have almost the same environment as control models but we have email viruses present. These simulations were carried in two main cases: single virus simulation case and multiple virus simulation case. Regardless of the number of e-mail viruses presented, all control simulations were carried with the same initial setup as was dependent in the control simulations. In this section, we present the results of these two cases. In the single virus simulation case, there was only one type of virus was provided in each simulation. In seven of these single virus simulations, the monitor produced five correct virus alerts. Hence, the detector was 60% percent effective in providing a true positive warning, although in these simulations it neglected to mention the existence of two viruses. The false-negative frequency was thus 33%. In the case of multiple virus simulations, every of the five held simulations included more than one form of the virus. They performed the first four tests of two viruses. The monitor developed five models for the hole eight true virus alerts; however, the actual positive rating for the test was 85%.

The negative test frequency was 20 % since two viruses in one system were not identified by the monitor. The fake-negative rate was 30%, and the actual concrete average was 85%. It was determined by the monitor being able to successfully detect two viruses while it failed to detect one virus, as shown in Table 2 and Figure 4.

Table 2. Multiple virus simulation.

Configuration for Network				
Number Of Clints	Score	Domain Checking	Monitor Domain	Case
50	30	4	4	4 viruses

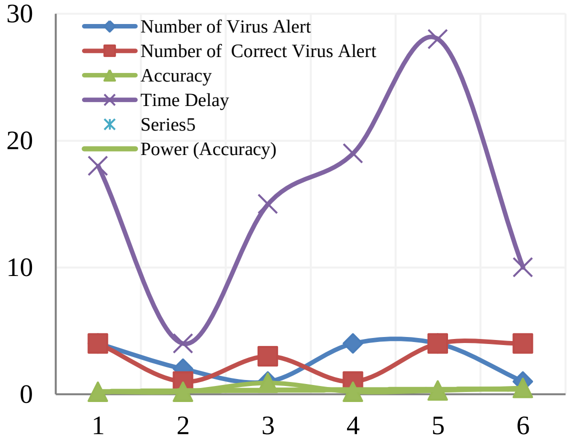

Figure 4. Simulation several viruses.

CONTROL SYSTEM PERFORMANCE

The effects of the previous section's held models revealed that the proposed device is capable of identifying email viruses to an appropriate degree of precision. And there are several limitations to this suggested strategy. It can initially give rise to some false virus alerts; secondly, it may struggle to identify one of the viruses. In the following sections, we evaluate the simulation results.

False Positive Alert Analysis

In the third control simulation trial, even though the absence of virus presence in one of the simulations of control, the monitor erred in giving the virus the alert. Nevertheless, the question arises as to why the monitor gives false virus alerts. In fact, in this simulation. the log data reveals the source of the faulty alert, as shown in Figure 5, Where the total of messages is typically very large per unit of time, reflecting email actions per unit time. In the third control simulation, the number of messages per unit was ten times higher than in the other six simulations. Therefore, false virus alerts are logically high due to the large amount of email activity. The hash decreases the risk of misclassifying multiple attachments with specific contents utilizing attachments.

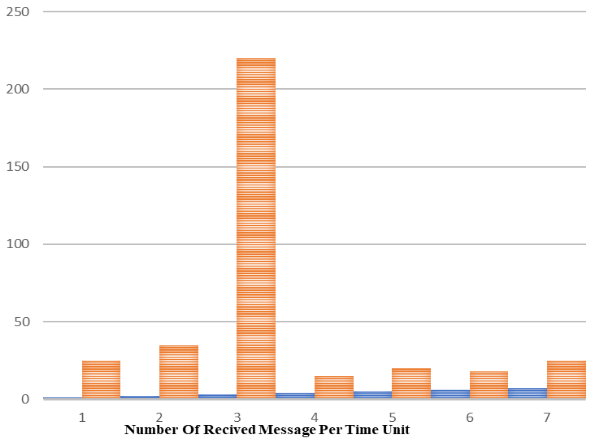

Figure 5. The False virus alert for email messages.

False Negative Alert Analysis

In the third trial simulation of the viruses in the infected system, the suggested surveillance approach failed to identify two viruses, and it was ineffective to capture one virus out of the three viruses in the fourth trial system. In fact, our traffic controller was able to identify several of the viruses in the simulations performed. Although the screen failed in detect a few of the viruses, it managed to detect most of them. The problem is that when there are few numbers of client threads in the region of the monitoring where there is virus infection, the tracking operation does not have a sufficient amount of infected tree nodes to establish the tree needed to cause a virus warning. The findings obtained from the tests carried out indicate that the suggested detection strategy is capable of detecting the email viruses. The accuracy of the proposed technique was nearly 75%. This percentage of effectiveness is agreed to take into account that the control algorithm has no specific knowledge about any web viruses. The email virus has successfully spread out in all 50 conducted simulations in Figure 6. In fact, the email virus has a small chance to die before it spreads out. The beginning of those initially infected users sends out virus copies to their neighbors. If all their neighbors decide not to open the email attachment for the first cycle, therefore, no virus email existed in the network after those neighbors finished checking their email for the first time.

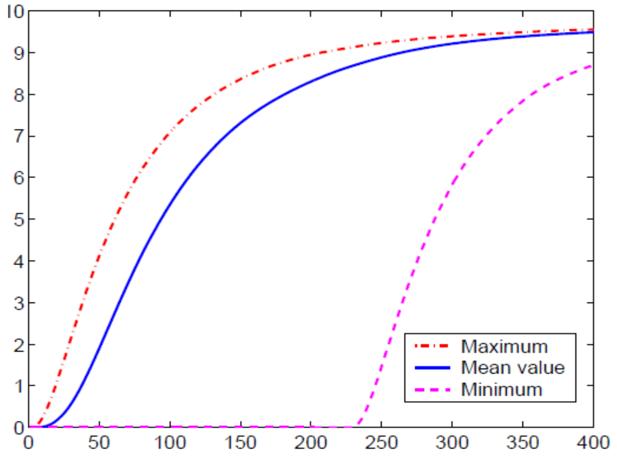

Figure 6. Random effect in the simulation.

CONCLUSIONS

In this paper, we presented a new approach for a new anti-virus technique that runs an antivirus program on the mail server to detect the e-mail viruses that arrive with the attachments by examining the network traffic activity based on the program called e-mail traffic monitor. The positive feature of this technique is to installing the antivirus software only at the mail server instead to install it on every client computer. This e-mail traffic activity monitor has the ability to minimize the anti-virus software cost because it is required to be installed only on the mail server. We presented a new email virus model by considering email user behaviors, such as the unusual frequently attempts to opening the email and the probability to open an email attachment. We explained why we believe email network can and should be modeled as we did use formulas in CySeMoL model simulation environments. The CySeMoL enables cyber security analyses of enterprise architectures without requiring any major cyber security knowledge of the modeler. The email traffic tracker would also be able to identify emerging-malware depending on their behavior. That we have revealed different effects of this experimental strategy from the model tests. We intend to explore the effects of self-control reviews on the potential to identify errors in a network structure in the future.

ACKNOWLEDGMENT

The Authors would like to express their acknowledgment to the Computer Science Department, College of Science, Mustansiriyah University for the encouragement and support to perform the current research.

REFERENCES

[1] D. J. McManus, C. Sankar, H. H. Carr, and F. N. Ford, “Intraorganizational versus interorganizational uses and benefits of electronic mail,” Information Resources Management Journal (IRMJ), vol. 15, no. 3, pp. 5–13, 2002.

[2] I. E. Korshunov, M. V. Lyadvinsky, S. M. Beloussov, and A. Sergeev, “System and method for restoration of MICROSOFT exchange server mail,” Nov. 5 2019, uS Patent 10,467,187.

[3] A. Boiko, V. Shendryk, and O. Boiko, “Information systems for supply chain management: uncertainties, risks and cyber security,” Procedia Computer Science, vol. 149, pp. 65–70, 2019.

[4] J. L. Ferrer-Gomilla, J. A. Onieva, M. Payeras, and J. Lopez, “Certified electronic mail: Properties revisited,” Computers & Security, vol. 29, no. 2, pp. 167– 179, 2010.

[5] A. Bhowmick and S. M. Hazarika, “Machine learning for E-mail spam filtering: review, techniques and trends,” arXiv preprint arXiv:1606.01042, 2016.

[6] A. Gonzalez-Torres, V. L. Byrd, and P. Parsons, “VKE: a Visual Analytics Tool for CyberSecurity Data,” in Proceedings of the International Conference on Security and Management (SAM). The Steering Committee of the World Congress in Computer Science, Computer, 2019, pp. 56–62.

[7] T. M. Chen and J.-M. Robert, “The evolution of viruses and worms,” Statistical methods in computer security, vol. 1, pp. 1–16, 2004.

[8] D. Moore, C. Shannon, D. J. Brown, G. M. Voelker, and S. Savage, “Inferring internet denial-of-service activity,” ACM Transactions on Computer Systems (TOCS), vol. 24, no. 2, pp. 115–139, 2006.

[9] C. Wang, J. C. Knight, and M. C. Elder, “On computer viral infection and the effect of

immunization," in Proceedings 16th Annual Computer Security Applications Conference (ACSAC'00). IEEE, 2000, pp. 246–256.

[10] S. J. Stolfo, E. Eskin, M. Bhattacharyya, and S. Herskop, "System and methods for detecting malicious email transmission," Jan. 17 2019, uS Patent App. 16/026,801.

[11] S. K. Sahay and A. Sharma, "A Survey on the Detection of Windows Desktops Malware," in Ambient Communications and Computer Systems. Springer, 2019, pp. 149–159.

[12] R. Bhargava, D. P. Reese et al., "System and method for passive threat detection using virtual memory inspection," Mar. 14 2017, uS Patent 9,594,881.

[13] J. Aizen, I. Rabinowitz, L. Kovacevich, M. Mccole, and L. Dauter, "Automated real estate transaction workflow management application extending and improving an existing email application," Dec. 27 2018, uS Patent App. 16/013,702.

[14] A. A. Akinola, "Quantitative evaluation of cyberattacks on a hypothetical school computer network," 2019.

[15] P. Marsh, "Knowledge swarming using mobile knowledge mentoring-the emergence of the ubiquitous cyber sage: knowledge transfer," Civil Engineering= Siviele Ingenieurswese, vol. 2016, no. v24i7, pp. 57– 63, 2016.

[16] T. Sommestad, M. Ekstedt, and H. Holm, "The cyber security modeling language: A tool for assessing the vulnerability of enterprise system architectures," IEEE Systems Journal, vol. 7, no. 3, pp. 363–373, 2012.

[17] H. Holm, T. Sommestad, M. Ekstedt, and L. Nordstrom, "CySeMoL: A tool for cyber security analysis ¨ of enterprises," in 22nd International Conference and Exhibition on Electricity Distribution (CIRED 2013). IET, 2013, pp. 1–4.

[18] K. Ahmad, S. Verma, N. Kumar, and J. Shekhar, "Classification of internet security attacks," in Proceeding of the 5th National Conference INDIACom2011Bharti Vidyapeeth's Institute of Computer Applications and Management, New Delhi ISSN, 2011, pp. 0973–7529.

[19] A. Abdulhameed, A. Hammad, H. Mountassir, and B. Tatibouet, "An approach to verify SysML functional requirements using Promela/SPIN," in 2015 12th International Symposium on Programming and Systems (ISPS). IEEE, 2015, pp. 1–9.

[20] M. Valja, M. Korman, R. Lagerstr ¨ om, U. Franke, and ¨ M. Ekstedt, "Automated architecture modeling for enterprise technology manageme using principles from data fusion: A security analysis case," in 2016 Portland international conference on management of engineering and technology (PICMET). IEEE, 2016, pp. 14–22.