Journal of Engineering Sciences, Assiut University, Vol. 38, No. 4, pp. 1013-1025, July 2010.

WORM DETECTION USING HONEYPOTS FOR WINDOWS ENVIRONMENT

***Mansour Ali H Alqubati*[1], *Yousef B Mahdy*[2] and *Hosny M. Ibrahim*[3]**

[1]*Student from Yemen NIAS*
[2]*Vice Dean of Faculty of Computers & Information*
[3]*Dean of Faculty of Computers & Information*

(Received June 9, 2010 Accepted June 24, 2010).

Recent cybersecurity incidents suggest that internet worms can spread so fast that in-time human-mediated reaction is not possible, and therefore initial response to cyberattacks has to be automated. In this paper we present a system for detecting known and unknown worms using honeypots. The proposed system detects worms by monitoring connection activity and watching for patterns of traffic that are expressions of some of the essential characteristics of worm behavior. The implementation is a signature-based detection as a first tier and an anomaly-based as a second tier in the detection process. At a network's gateway, the proposed system runs a vantage point from which all traffic into and out of the network is visible.

The system employs a honeypot to capture traffic, after discarding whitelisted patterns; as it automatically generates worm signatures which are matched with the signatures of the known worms stored in original database. When a signature is matched, the system reports it by issuing an alert that also includes the IP addresses involved in the transaction. Otherwise, the system monitors the changes in the performance of CPU, RAM and changes in files in the gateway which are considered as indicators to the presence of worms.

The proposed system was evaluated using a dataset collected from internet for several days, and potentially showed good results for detecting and collecting information about worms from local network. It was noticed that the performance was increased up to 23% more than other systems that uses honeypots.

KEYWORDS: *honeypot, worm, network security*

I. INTRODUCTION

The internet is a ceaseless evolving, dynamically increasing scale-free network in topology structure [1]. It encounters increasing threats that are caused by worms. Recent cyberattacks outbreaks have shown that internet worms are able to infect thousands of internet computers in less than one hour.

The internet worm is a self-replicating computer program, that sends copies of itself to other computer connected to that network and it may do so without any user intervention [2]. In recent years, several worm outbreaks on the Internet have caused major computer troubles worldwide [3]. Since the first Morris worm arose in 1988, security threat posted by worms has steadily increased. Recent worms, such as, “Code Red”, “Nimda” and “Slammer” [4, 5], emerge at high speeds and cause great economic losses to our society. It is likely that new worms will appear and able to propagate even faster than Slammer causing larger economic damage than Code Red. Therefore there will be a need for an automated mechanism to efficiently detect and prevent these worms and threats quickly enough to prevent any global outbreak.

Honeypot is a security technology that provides a good way to alter the situation of security; a honeypot is "*an information system resource whose value lies in unauthorized or illicit use of that resource*” [6, 7, 8]. There are two types of computer honeypots: low interaction and high interaction honeypots [9].

Low interaction honeypots are mainly used to detect the hackers and deceive them by emulating the operating system services and port services on the host operating system; while the *high interaction honeypots,* make the hackers interact more by using the real operating system services rather than the emulated services [10]. The internet security and defense are the focus of the present work.

This work presents a local worm detection and reacting system by analyzing the characteristics of traffic generated by the TCP-based worm. The proposed system can detect and reaction with known worms using the signature-based and detection model. Also, the system detects and reacts with the unknown worms using the anomaly detection model.

The implementation of the proposed system is a signature-based detection as a first tier for detecting the known worms, and an anomaly-based detection as a second tier for detecting the unknown worms. In the detection process, the proposed system runs at a network’s gateway a vantage point from which all traffic into and out of the network is visible.

The system employs honeypot to capture traffic (after discarding white-listed patterns) it automatically generates worm signatures which are matched with the signatures of the known worms stored in a database. Whenever a signature is matched, the system reports it by issuing an alert that also includes the IP addresses involved in the transaction. Otherwise, the system monitors the changes in the performance of CPU, RAM and changes in files in the gateway which are considered as indicators to the presence of worms.

The rest of the paper is organized as follows: It begins by reviewing some previous work related to the current work in section two. Section three presents the architecture and implementation of the proposed system, while section four presents the evaluation of the proposed system using realistic network traffic. Finally, section five summarizes and concludes the paper.

II. RELATED WORK

Recently, researchers have paid attention to the necessity of monitoring the internet for malicious activities, and have been interested in the idea of honeypots to capture and analyze worms’ behavior from different perspectives. They proposed many methods to apply this idea. For example the honeynet project using a network of high-interaction honeypots over a DSL connection, managed to capture various worms in action [4]. Such study produced detailed descriptions of worms’ behavior by analyzing network traffic and the honeypot state.

A study presented a method to capture the MSBlaster worm and launch a counter offensive against it by using a low-interaction virtual honeypot [11].

For automatically detecting and disabling worms' outbreaks at the network level, a study [12] employed the random IP scanning technique that worms use to discover new targets, by evaluating the system on an academic network and the results showed that it could detect and filter worm traffic within minutes without prior knowledge.

In [13], it is suggested that in addition to increasing the quantity of data used by alert systems, the quality can be improved as well. The study showed that if intrusion detection is like finding a needle in a haystack, then a honeypot is like a stack of needles. Honeypots are therefore used to create a highly accurate alert stream. Using logistic regression, researchers showed how a honeypot alert stream can detect worm outbreaks. Moreover, they defined three classes of events to capture memory, disk and network activities of worms.

The authors in [14] discussed the design and implementation of SweetBait, an automated protection system that employs low- and high-interaction honeypots to recognize and capture suspicious traffic: after discarding whitelisted patterns, it automatically generates worm signatures, providing a low response time. The signatures may be immediately distributed to network intrusion detection and prevention systems. At the same time, the signatures are continuously refined in order to increase accuracy and lower false identification rates. By monitoring signature activity and predicting ascending or descending trends in worm virulence, researchers are able to sort signatures. As a result, the set of signatures to be monitored or filtered is managed in such a way that new and very active worms are always included in the set. Meanwhile, the size of the set is bounded. Researchers also demonstrated how information can be distributed and deployed without any human intervention, minimizing reaction time to zero-day worms.

A worm propagation model based on two-factor model in the network which distributed honeynet has been deployed by [15]. Using worm propagation trend and the impact of honeypot on worm spread, researchers performed a simulation experiment to test their proposed method and they found that distributed honeynet was of great significance in worm warning and restraining in large-scale networks. In addition, the authors gave a correspondingly control strategy based on the mechanism of worm information-sharing and immunization. The honeypot host could divide network into many parts for its data control policy under distributed honeynet, preventing worm from spreading in large-scale networks and ensuring network security effectively.

A further study about the existing worm propagation models was done in [16]. In this study, the project utilized honeypots to detect worms and conduct simulations using some of these worm propagation models. In addition, the authors concluded that it was difficult to produce realistic results prior to a worm outbreak in the light of the results obtained from the experiment.

III. SYSTEM ARCHITECTURE AND IMPLEMENTATION

In general, the proposed architecture of a worm monitoring and detecting system can be divided into two different categories: host-based intrusion detection system (IDS) and network based IDS. Conventionally, a host based IDS detect whether or not the host is under attack while network based IDS detects whether or not a network is under attack. . This work investigates how a network based approach performs in detecting host-based worm infections. At a network's gateway, the proposed system runs a vantage point from which all traffic into and out of the network is visible.

The intrusion detection system for detecting worms in network can be classified into general categories: signature-based and anomaly detection. Signature-based detection is based on defining malicious patterns that the system has to detect [17]. The signature-based detection suffers from the problem that it requires a signature for each worm to be known. In contrast, the anomaly detection differs by constructing a profile of normal behaviors or activities on the network (host), and then looking for activities that do not fit the normal profile. Since not all the abnormal activities in the network are suspicious, anomaly detection has the problem of raising false alarms when it encounters normal traffic that it has not seen before. However, anomaly detection has the important advantage that it can be used to detect new worms with unknown signature.

3.1 System Architecture

Architecturally, the system consists of three main parts: a honeypot, databases and watchers (Figure 1).

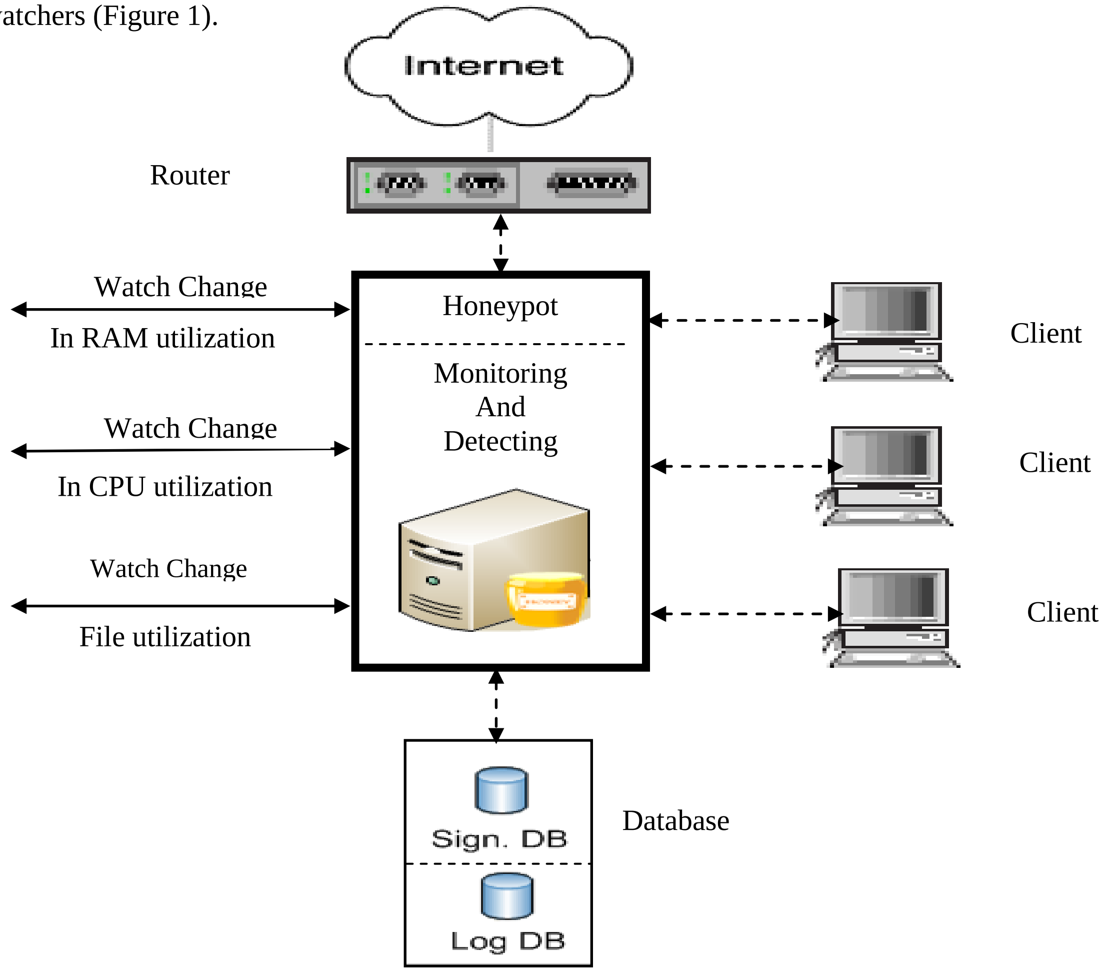

Figure 1: Architecture of the proposed system

3.1.1 Honeypot

Honeypot is the major part of the system through which all incoming and outgoing traffics must be directed for checking and monitoring them, also to detect abnormal behaviors like changes in RAM, CPU and files. The actual worm detection mechanism is based on trapping the worms inside the system. Using signature patterns stored in its database or any abnormal behavior when opening web page. The honeypot deludes the worm with a real system that is harmful, however, it already existing in the virtual honeypot that it is isolated from the real production system.

3.1.2 Databases

The proposed system is based on two types of databases: signature and log.

Signature Database

The signature database is used to store the signatures of known worms as shown in Fig. (2).

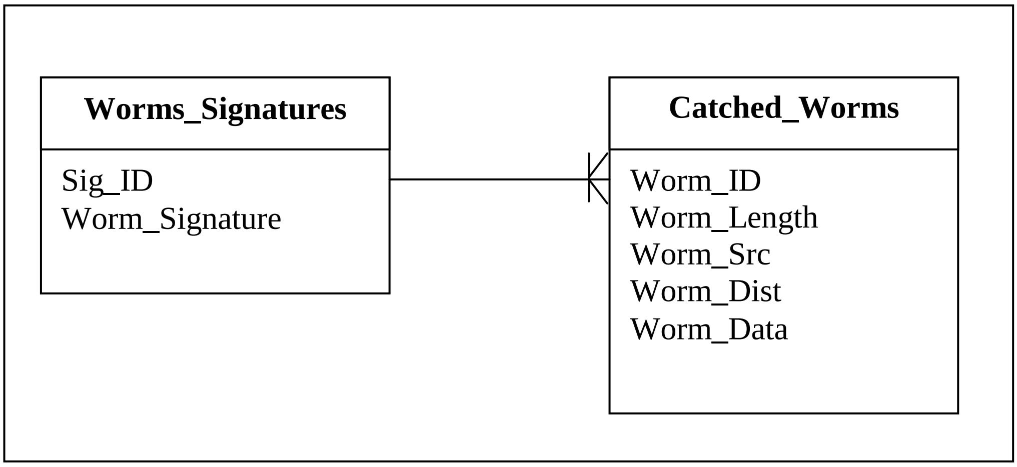

Figure 2: Structure of signature database

The above structure of the signature database defines the use of this database in the system, as the user enters the signatures to the signatures table of the database (updated manually by the user). The honeypot system uses the signature table to compare the signatures patterns to the captured packets. If the compared packets contain such stored signature, this will be stored in the Catched_Worms table as a worm. The worm signatures of known worms are collected from internet [19].

Log Database

The system can also store the detected worm information e.g. (packet length, worm source, worm destination and worm data) in a log file that can be checked easily and manually. In addition, log database store information about the web pages that are infected with unknown worms which are detected by the system. Also, the URL address of the page that contains the worm is stored.

3.1.3 Watchers

The system consists of three detectors. Processors watcher, RAM watcher and file watcher.

The processors watcher:

It monitors the changes in the processor performance.

The RAM watcher and file watcher:

It monitors the changes in RAM size, while the file watcher monitors the changes that happened to the files stored in the secondary storage, such as (deletes, modify access or rename the file name).

3.2 Implementation

The system was implemented with easy interface and configuration. The full object oriented model and design is used, as C# is selected as a programming language.

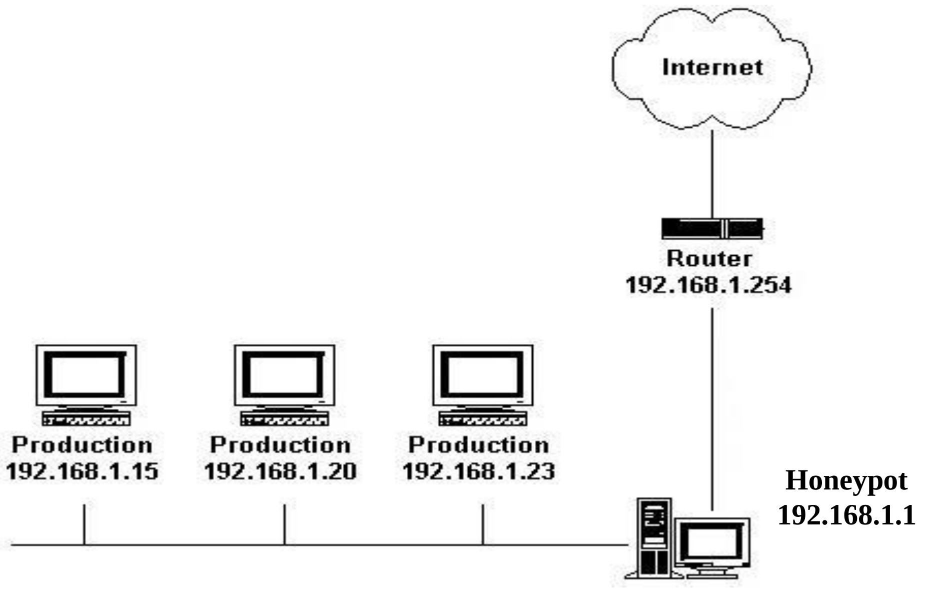

Figure 3: The proposed system deployment.

Figure (3) shows the deployment for the proposed system through the network, where the system is installed at VMware [18] under Windows platform. The proposed system is connected to the router from a channel and to the production system from the other channel so that the system detects the harmful worms before it arrives to the production system.

The system uses WinpCap library (windows packet capture) to capture networks packets that are incoming and outcoming to the system honeypot server. The suggested system detects both known worms and unknown worms. For known worms, the system matches the contents of packets with the signature that stored in database. For any worms detected, the user will be informed and this will be saved in log database. In contrast, when the signature is not found in the database (unknown worms to the system), the system will be depending on watchers like (CPU, RAM, and Files) that monitor the abnormal activities in CPU,RAM and files. To determine any abnormal behaviors in any one of these resources these will be recorded in log database as unknown worms and alerts will be given.

3.2.1 Detection Of Known Worms

The simplest form of signature-based detection is expression matching, which searches an event stream (log entries, network traffic, or like) for occurrences of specific patterns. This approach uses a longest common substring algorithm (LCS) [16] for worm's detection.

Using packet-oriented systems, the detection algorithm checks every packet and does the attack analysis on the basis of each packet, i.e., packets are not considered as a complete connection session, but individual packets are treated independently instead.

The packet matching process can either be deterministic, as the arriving packets are compared to packets stored in a database, or statistical, as the packet flow is statistically analyzed.

The process of detecting known worms depends on Winpcap library. It captures any network packet like TCP packets, UDP packets, Ethernet packets, etc. Additionally, it gets information about the packet like (packet length, packet IP source address and packet destination address). It is possible to store this information in buffer that can be sent to the honeypot for processing.

The honeypot of the system takes the packets from the buffer individually, and then compares the content of packets with signatures patterns that are stored in database. If there is any matching with signatures that are stored in database, the system will detect the worm, and then record its information.

3.2.3 Detection Of Unknown Worms

Previously, we discussed how the proposed system detects the known worms by signature matching. However, there is an alternative option that will be explained: how the suggested system works when the packets contain unknown worms? This can be achieved via watchers that monitor some resources of the machine to determine any abnormal activities that appear through sending and receiving packets or when running any other programs that may lead to abnormal behaviors. These watchers are a processor watcher, a RAM watcher and a File watcher.

1 - Processor Watcher

The processor watcher is used to monitor the changes in the performance of processor which appear as a result of any abnormal behavior activities, as it works as sniffer of the processor performance, i.e. it decreases the efficiency of processing.

Assuming that the user opens a web page; that may causes changes in the performance of the CPU, such as decreasing the speed of processing and consequently the usage of CPU increases more than 80%. This means that the page contains unknown worms and information about them will be stored in database.

2 - Ram Watcher

The RAM watcher is to monitor the changes in the RAM size. As mentioned above, the web pages that contain worms also can affect the RAM performance. Moreover, the user can observe abnormal change in the RAM performance which makes the machine hang. The system monitors the changes via RAM watcher if the memory usage increases more than 90%, which means that the web page contains unknown worms, and then can be saved in the system database.

3 - Files Watcher

The Files watcher is used to monitor the changes that happened to the files stored in the secondary storage. These changes can be described as deleting, modifying, accessing or renaming the file name. According to the technique of the honeypot system watcher as mentioned previously, the file watcher watches over the secondary storage for any hidden changes that the user cannot see or the changes that happen without his control. Internally, there are many worms designed for causing modifications in files or causing harm, especially changing the execution files; also other worms that try to set their files in the system environment. For that, all these types are considered unknown worms and proposed system will be able to detect all these activities due to the unknown worms, consequently gathering information about them in database.

IV. EXPERIMENT AND EVALUATION

The system is tested over the period of one week by using dataset collected from many web pages via the internet as listed in (Table.1). The packets are used to evaluate the known and unknown worms that the system tries to catch and collect information about them.

Table 2 and 3 show the number of worms that the system detects (known and unknown) worms. Figures 4 and 5 show the results that the system produced by simulation. Figure 4 displays the number of known worms that the system detected and we can note the following:

- Increase or decrease in the number of detected worms is not necessary, when the number of packets increased.
- The overall performance of the system is computed from:

$$\text{Performance of system} = \sum_{x=1}^{7}((?NP/?WD)/ND)*100 \qquad [1]$$

  Where NP is number of packets, WD is number of worm detected, and ND is number of days.
- The overall performance of the system is improved up to 23%.

In addition, Figure (5) shows the number of web pages that opened by users as well as the changes that appear when are requested. The system detects 13% infected pages from the pages that were opened, which is the ratio of the number of infected pages to the number of pages that are opened.

Finally, the results that were obtained by the suggested system is good, where the performance is increased up to 23%, as well as when comparing the produced results with previous work done in [16] whose performance was less than 17% as shown in Figure (6) where, the data shown are copied.

Table 1: Information about dataset

Day number	Number of packets	Number of pages
1	8282	65
2	6931	60
3	8302	35
4	5434	18
5	6921	45
6	5211	37
7	6133	50

Table 2: Number of known worms that svstem detected

Day number	Number of packets	Worms catched by system using signature
1	8282	2300
2	6931	1100
3	8302	2350
4	5434	1400
5	6921	1950
6	5211	1050
7	6133	1200

Table 3: Number of unknown worms that svstem detected

Day number	Number of pages	Number of detected pages
1	65	8
2	60	7
3	35	5
4	18	5
5	45	4
6	37	5
7	50	6

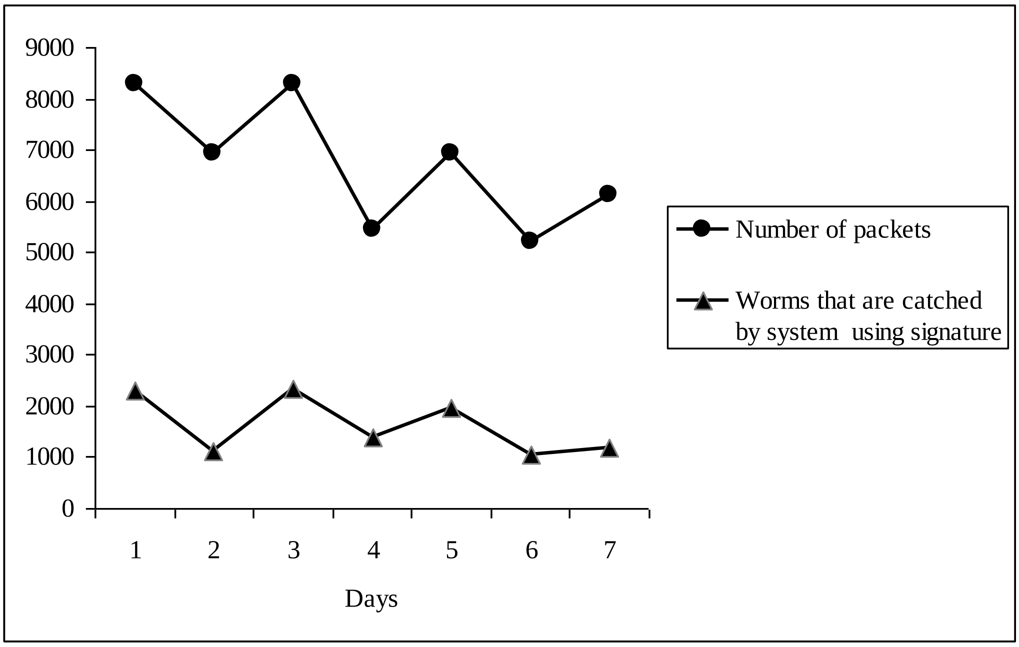

Figure (4):.Number of known worms that detected by system.

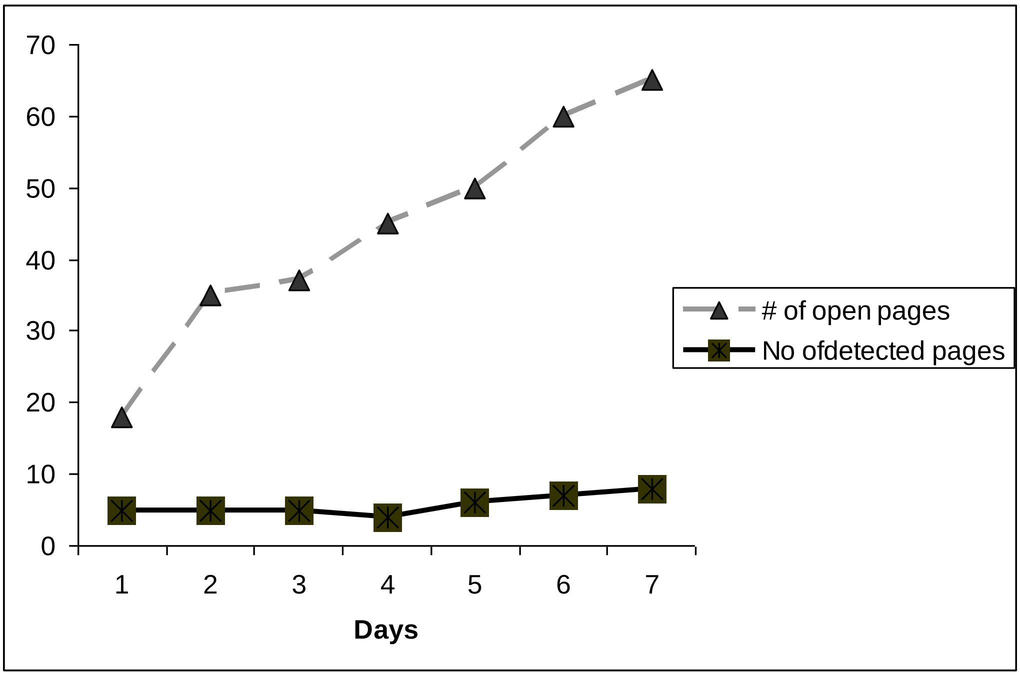

Figure (5): .Number of opened pages and the number of detected pages

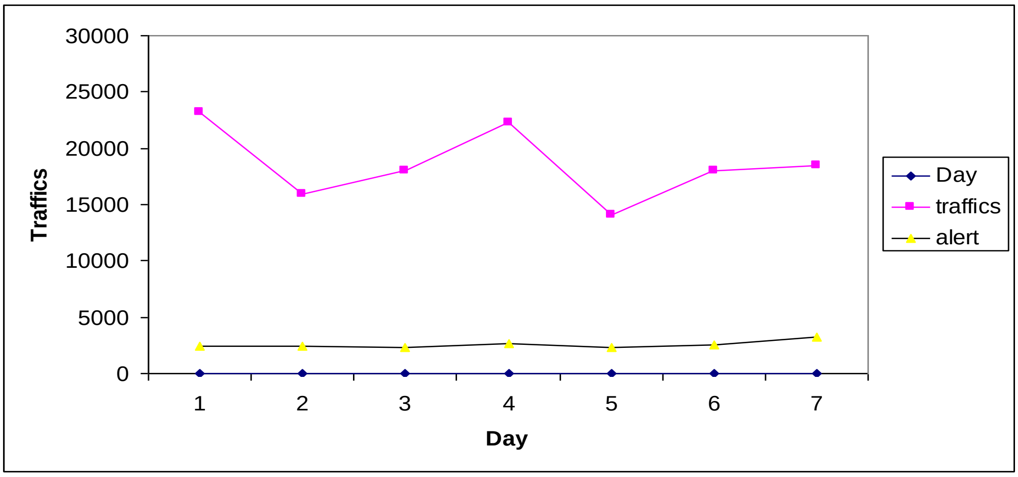

Figure (6): Results of system proposed in [16].

V. CONCLUSION

In this paper design and Implementation for a detection worm system using honeypots for windows environment. It's is major goal is to detect and collect information about worms, where it employs the honeypot techniques to achieve this task. The proposed system is evaluated using data sets collected from the internet [19].

Experiments show that suggested system is able to detect and collect information about worms. The performance of the over all system is found to increase up to 23%, which is good result according to diversity and high availabilities of worms in the internet.

Experiments prove that the results that obtained are not static, but it may be increase or decrease according to the following reasons from our perspective:

- The size and content of the data sets.
- The size of the network that may the system will be applied in it.

Finally, the results that obtained are good according to the reasons that mentioned above.

VI. REFERENCES

[1] Faloutsos. M, Faloutsos. P and Faloutsos. C, "On Power-Law Relationships of the Internet Topology," Computer Communications Review, 1999.

[2] Jose Nazario, "Defense And Detection Strategies Against Internet Worms, "Artech House, 2004.

[3] Su Fei, Lin Zhaowen and Ma Yan, "A Survey of Internet Worm Ropagation Models," IEEE, 2009.

[4] Honeynet Project, "Know Your Enemy: Statistics," http://www.securityfocus.com/infocus/1233, July 2000, access in 27/1/2010.

[5] Narisa Zhao and Xianfeng Zhang, "The Worm Propagation Model and Control Strategy Based on Distributed Honeynet," IEEE,2008

[6] Roger A. Grimes, “Honeypots for Windows,” Apress, 2005.

[7] Niels Provos, Thorsten Holz , “Virtual Honeypots: From Botnet Tracking to Intrusion Detection,” Addison Wesley Professional, 2007.

[8] Lance Spitzner,”Honeypots: Tracking Hackers,” Addison Wesley, 2003.

[9] Lucas Tamagna-Darr, Yin Pan, Bo Yuan And Charles Border, “Evaluating the Effectiveness of an Intrusion Prevention Honeypot Hybrid,” UMI, October, 2009.

[10] V.Maheswari And Dr. P. E. Sankaranarayanan, “Honeypots: Deployment and Data Forensic Analysis,” IEEE ,2007.

[11] Laurent Oudot, “Fighting Internet Worms With Honeypots,” http://www.securityfocus.com/infocus/1740, 2003.access in 27/1/2010.

[12] Georgios Portokalidis, “Zero Hour Worm Detection And Containment Using Honeypots,” Leiden University, November. 2004.

[13] David Dagon, Xinzhou Qin, Guofei Gu, Wenke Lee, Julian Grizzard, John Levine and Henry Owen, “HoneyStat: Local Worm Detection Using Honeypots,” Springer Berlin / Heidelberg, 2004.

[14] Georgios Portokalidis and Herbert Bos, “SweetBait: Zero-hour worm detection and containment using low- and high-interaction honeypots,” Elsevier B.V, 2006.

[15] Narisa Zhao and Xianfeng Zhang, “The Worm Propagation Model and Control Strategy Based on Distributed Honeynet,” IEEE, 2008.

[16] Dag Christoffersen and Bengt Jonny Mauland, “Worm Detection Using Honeypots,” IEEE, 2006.

[17] Al-Hammadi, Y. Leckie, C, “Anomaly detection for Internet worms,” IEEE, 2005.

[18] Samuel T. King, George W. Dunlap and Peter M. Chen, “Operating system support for virtual machines, “In Proceedings of the 2003 USENIX Technical Conference, 2003.

[19] http://www.ispace.edu.vn/forum/showthread.php?t=10617, access at 20-9-2009.

كشف ديدان الحاسب باستخدام مصائد الحاسب في بيئة نظام تشغيل النوافذ

إن التقدم التقني في مجال الشبكات لابد من ان يصاحبه تطور في مجال أمن المعلومات والذي اصبح ينمو بشكل متزايد مع ذلك التقدم خاصة في السنوات الاخيره . لذا اصبح أمن المعلومات مصدر قلق للمنظمات والافراد على حد سواء وذلك لزيادة الهجمات والتهديدات التي تتعرض لها أجهزة الحاسبات ومن تلك الهجمات البرامج الخبيثه (ديدان الحاسب) وهي تقوم باعمال تخريبية بأتلاف البرامج والملفات وتعطيل عمل الحاسب ومن خصائصها سرعة الانتشار الذاتي على الشبكة بدون تدخل المستخدم. فأصبح من الصعب على المستخدمين منعها وإزالتها يدوياً لذا من الضروري إيجاد وسائل آلية لهذا الغرض، وهذا أدى الى الاهتمام المتزايد في اكتشاف طرق أكثر جرأة في مجال حماية و أمن المعلومات لاستكمال الاساليب المستخدمه في حماية وامن المعلومات و إحدى هذه الطرق هي استخدام مصائد

الكمبيوتر(جرار العسل). وفي ضوء هذا، فإن الهدف من هذه المقالة العلمية هو تقديم نظام آلي لاكتشاف ديدان الحاسب سواء كانت المعروفة او المجهولة وذلك باستخدام مصائد الحاسب، ومصيدة الحاسب عبارة عن فخ يعمل على إبطاء المهاجم وتجعله يهدر وقته كما انها تجمع كافة المعلومات عن المهاجم والطرق التي استخدمها للهجوم حتى يتسنى لمراقب الشبكة إتخاذ الإجراء المناسب وهي لا تحتوي على اي مصادر ذات قيمة . النظام المقترح يكتشف ديدان الحاسب بمراقبة نشاطات الاتصال ويقوم بمطابقة الحزم مع البصمات المخزنة مسبقاً في قاعدة بيانات لبصمات الديدان المعروفة، بينما يتم كشف الديدان المجهولة بناءً على سلوكها لما تحدثه من تغيرات في ذاكرة الحاسب او معالجه او نظام ملفاته. وفي كلتا الحالتين يقوم النظام بتسجيل بيانات عن ديدان الحاسب التي تم اكتشافها في ملفات خاصة بتجميع البيانات لغرض تحليلها.

تم تنفيذ وتجريب النظام المقترح على شبكة محلية بتثبيت النظام المقترح في جهاز مضيف مرتبطة به مجموعة من الاجهزة الاخرى، وكانت نتائج التجارب لهذا النظام فاعلة حيث اظهر النظام المقترح تحسين بنسبة 23% مقارنةً بالانظمة المثيلة.