A NARRATIVE WEB BASED MALWARE PROPHECY ACCESSION USING K-NEAREST NEIGHBOUR CLASSIFICATION AND CLIENT HONEYPOT DATA

Vishal Prajapati*, Anurag Chandna[1], Sonu Kumar[2]

[*] M. Tech. Scholar, Computer Science & Engineering, Uttarakhand Technical University, Dehradun, India
[1]Asst. Professor, Computer Science & Engineering, Roorkee College of Engineering, Roorkee, India,
[2]Asst. Professor, Computer Science & Engineering, Roorkee College of Engineering, Roorkee, India

ABSTRACT

Malware is major security threat on the Internet now-a-days. Many anti-virus companies are receiving a large number of malware samples every day. Malware is a type of malicious software which is used to disrupt computer operation, gather sensitive information, or gain access to private computer systems. Malware is defined by its malicious intent, acting against the requirements of the computer user, and does not include software that causes unintentional harm due to some deficiency. The term badware is sometimes used for malware, and applied to both true (malicious) malware and unintentionally harmful software. Adwares, Bots, Spam, Spyware, Backdoors, Trojan horse, Rootkit and Ransomeware are the types of malware.

In this thesis honeypot system, efficient malware classification approaches, malware behaviour, malware analysis and predefined malware categories are studied. Further, to classify the malware into various classes, open source web technologies PHP and Mysql are also studied. Many sandboxes are also studied, for thesis purpose cuckoo sandbox is used for the static and dynamic analysis of the malware applications. It simply means that one can throw any suspicious file at it and in a matter of seconds cuckoo will provide us back some detailed results outlining what such file did when executed inside an isolated environment. I developed a supervised machine learning based web GUI for scanning and classifying the malware application into different classes.

KEYWORDS: Ransomeware, Sensitive Information, Adwares, Bots, Spam, Spyware, Backdoors, Trojan horse

INTRODUCTION

A cyber-attack system is any kind of offensive system used by individuals or large organizations that targets infrastructures, computer networks and information systems, and other devices in various malicious fields. The threats or attacks usually originate from an unknown source that either steals, modifies, or completely destroys particular target by hacking into a vulnerable part of the system.

Cyber-attacks have become increasingly sophisticated and dangerous and a preferred method of attacks against large entities, by attackers. Cyber-war or cyber-terrorism is synonymous with cyber attacks and exploits three main factors for terrorising people that further impairs tourism, development and smooth functioning of governments and other infrastructure in a country or large business corporations. These factors are fear about security of lives; large scale economic loss that causes negative publicity about a corporation or government and vulnerability of government systems and infrastructure that raises questions about the integrity and authenticity of data that is published in them.

Role of malware in cyber attacks:

In all these different techniques from the early times to now, one component of the attack architecture necessarily needs to be detected and mitigated from spreading itself. These are the malicious executable files or the malware, that do the bulk of the intrusive activities on a system and that spreads itself across the hosts in a network. Malicious software (malware) is defined as software performing actions intended by an attacker, mostly with malicious intentions of stealing information, identity or other resources in the computing systems. The malware

exhibit different sort of malicious behaviour on the target systems and it is essential to prevent their activity and further proliferation in the network using different methods.

The internet represents a vital resource identifying a collaborative process between organisations and individuals, which can communicate and perform business processes. The internet arose from a research environment, so it was not designed to be a very secure environment. Therefore, various internet users, such as agencies and organizations have been attacked or probed by intruders, with resultant losses to productivity and reputation. In some cases, organizations have decided to disable their internet access and have invested significant resources for improving the security of their internet connection. Internet connectivity offers enormous benefits in terms of increased access to information, but using the internet can become a dangerous experience for those with low levels of security. Major threats of using the internet are problems with TCP/IP services, the complexity of host configuration and vulnerabilities introduced in the software development process.

Generally, web attacks are divided into two forms

I. Server Side Atteck.
II. Clent Side Atteck

Honeypots Description

The Concept of Honeypots though not the term itself, was first explained by Clifford Stoll's book titled The Cuckoo's Egg (1999) and Bill Cheswick's (Evening with Berferd, 2001). The Cuckoo's Egg (1999) is a story in which the author patiently tracks down a hacker after monitoring his activities for months. An Evening with Berferd (2001) is a chronicle of hacker's activities and how he is lured and tracked down. Lantz Spitzner defines a Honeypot as "a security resource whose value lies in a bring probed, attack or compromised".

The objective of this process is to then improve the security of the operating system and network, as well as gather information about the attacker's motives and behavior. By contrast, instead of waiting for attackers passively, an active Honeypot will go and search for the attackers. The Honeypots can be used:

1. By the network administrator, to learn how the attackers are getting into the networked computers, by monitoring attacker methods and exploits when they compromise the Honeypot system.
2. To collect known and unknown malicious codes for analysis by security professionals and companies to release patches or learn new attack methods used.
3. As an easy target for the attackers to compromise. Therefore, it will attract the attacker's attention while keeping their eyes away from network servers, which are harder targets; meanwhile the network administrator is informed of the attack and able to defend it.

The log files of Honeypot traffic have a high value for the security team and owner of the network because they reveal the traffic of the attacker with a low false positive, compared to a firewall or an Intrusion Detection System (IDS).

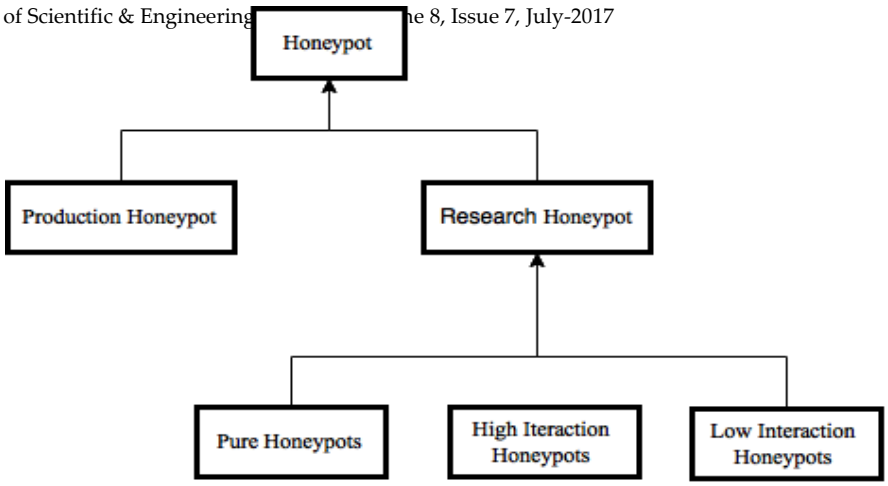


Fig. 1.2: Type of Honeypot.

The above figure 1.1 shows the general type of honeypots. Apart from these honeypots, other some specific honeypots may be there like malware honeypot etc.

Table 1.1: General form of honeypots

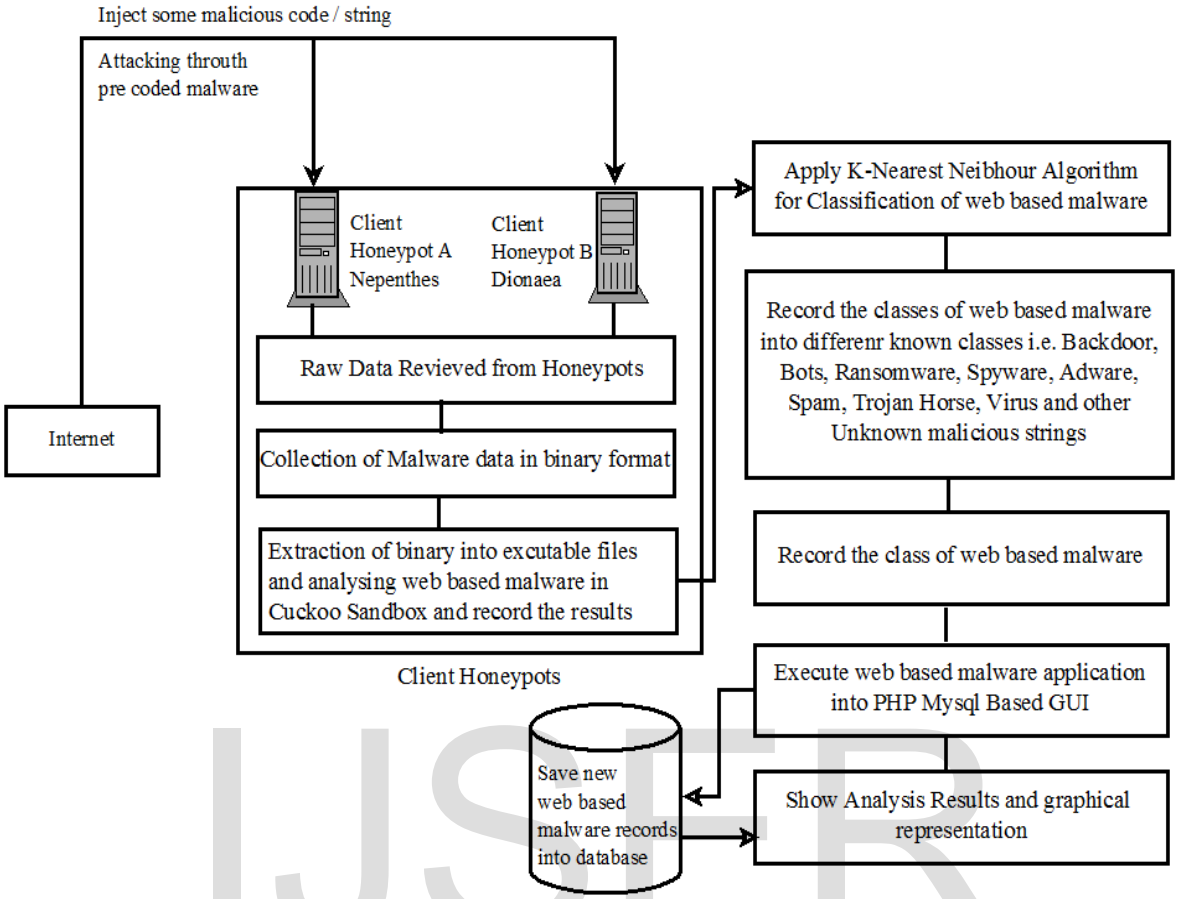
	Low Interaction	Medium Interaction	High Interaction
Access	Low: No Access	Controlled Access	Full Access
Real OS	No	No	Yes
Risk Level	Low Risk	Medium Risk	High Risk
Data Collection	Limited: Only connection attempts	Varied: Depending on the intruder skills	Extensive: All Available Data
Setup & Configuration	Easy	Easy/Medium	Hard
Maintenance	Easy	Medium	Hard

Problem Statement

The problem statement for the thesis is to classify the malware collected from honeypots into predefined categories using k-nearest neighbour approach.

Methodology implemented:

The following diagram shows the flow of work done in brief. The different steps with detailed work are given below with *K-nearest* neighbor algorithm:



K-nearest neighbour approach for classification : *K-nearest* neighbor algorithm is a method for classifying objects based on closest training examples in the feature space. *K-nearest* neighbor algorithm is among the simplest of all machine learning algorithms. Training process for this algorithm only consists of storing feature vectors and labels of the training data. In the classification process, the unlabelled query point is simply assigned to the label of its *k* nearest neighbors. Typically the object is classified based on the labels of its *k* nearest neighbors by majority vote.

If *k* =1, the object is simply classified as the class of the object nearest to it. When there are only two classes, *k* must be an odd integer. However, there can still be ties when *k* is an odd integer when performing multiclass classification. After we convert each string pattern image to a vector of fixed-length with real numbers, we used the most common distance function for KNN which is Euclidean distance:

$$d(x,y) = ||x - y|| = \sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

where *x* and *y* are histograms in shows visualizes the process of KNN classification.

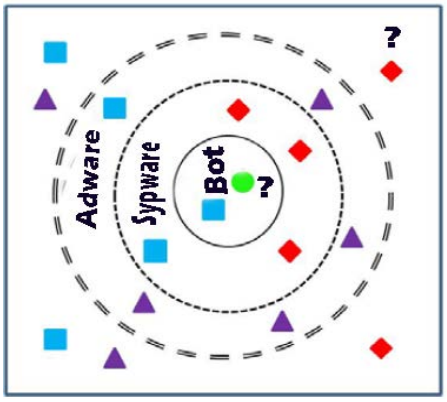


Figure 3.2: A KNN Approach for malware classification

In the Figure 3.2, three malware classes are taken i.e Adware, Spyware and Bot. The triangle, circle, rectangle and diamonds are the string pattern, which is to be classified into various categories.

Result and Discussions:

The observation obtained from the different scanning processes of the web based GUI. This section deals with the look and feel of the web based GUI also

The application has been developed using open source tools (php and mysql), it has 3 modules:

1. **File scan:** With this module one can upload the malicious file into the file system.
2. **Root directory scan:** This module is developed for root directory (in which application is installed).
3. **Directory scan:** It will scan the whole directory selected by the user.

Results of root directory scanning process. It is clear that after scanning process a "**spyware**" is found with the pattern number in **directory.php** file, below this, a small **malware description** is given, which is showing the behavior of the malware. the infected line of code with the line number is given, which indicates that the string pattern (which is defined as malicious) is matched with the given line of code (LOC).

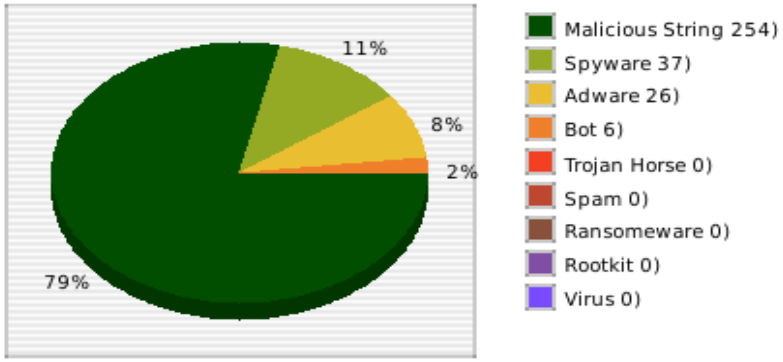


Figure 4.4: Pie chart for root directory scan

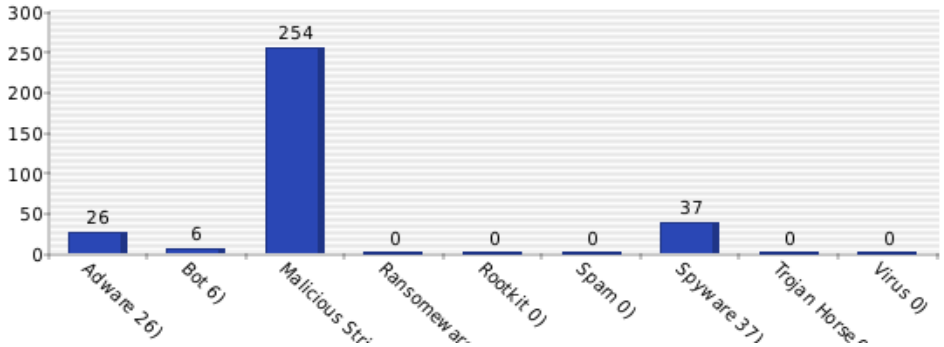


Figure 4.5: Bar graph for root directory scan

In above figures (Figure 4.4 and Figure 4.5), the graphs generated dynamically for root directory (in which the application is installed) scan using php& mysql.

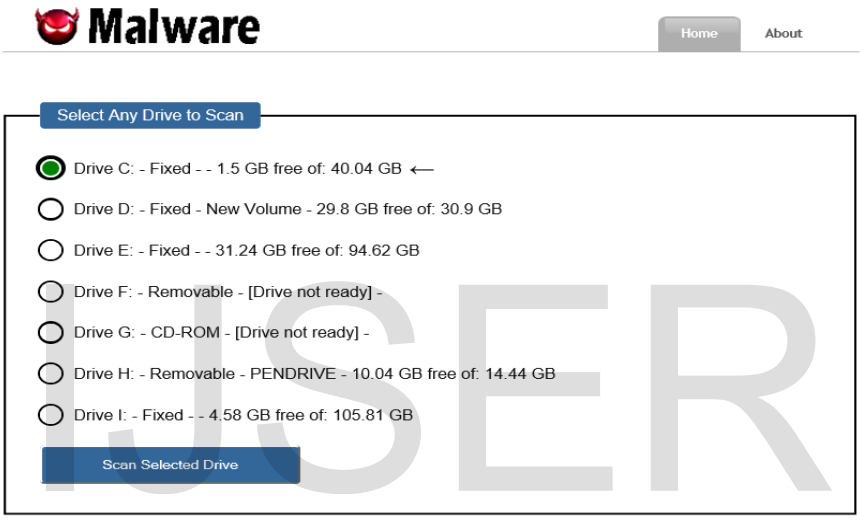


Figure 4.6: Disk drive scan module

In the Figure 4.6, the application automatically detects the disk drives installed on the computer system in windows operating systems and asks which drive to be scan. Only one drive can be scanned at time. In the scanning process it will scan the whole directory and read each line of code in the application and match the string patterns. If there is a matched, it will give the class of the malware as a result.

CONCLUSION

In this works, After analyzing process we got many string patterns for different malware application. These string patterns were used to classify the malware data into various categories by applying the K-Nearest Neighbour algorithm.

On the basis of results obtained, following conclusions are made:

1. *New Threats Found:* After deployment of the honeypot data a large number of malwares captured daily in binary file format, those binary files contains packed applications. After extraction of the binary file, we got the actual executable and supporting files.

2. *Category of malware predicted:* After string pattern were obtained we developed a php mysql web application based on supervised learning. Which is capable to classify malware data into different categories: Adware, Bot, Backdoor, Spyware, Spam, Trojan horse, Rootkit, Virus and other malicious strings. In the final experiment, we took total 60 samples from the both honeypots. In the 60 samples we got total 323 parts of malicious code were found, in which 79% malicious strings, 11%Adware, 8% Spywares and 2% Bots were detected.

Future Scope

1. In this work we have taken only 60 malware samples to classify the malware data, but in future more number of samples can be taken for more string patterns because more string patterns can give the better results.
2. The application is designed using php and mysql which is web based, so one can host this application to his/her website and prevent from the attacks.
3. Currently, the web based application is able to classify the malware class, but not able to remove the vulnerability, in order to enhance the capability of the application, in future one can make the application strong enough to be able to remove the complete part of code from the portable executable or malware samples

REFERENCES

[1] Abbott, R. P.; Chin, J. S.; Donnelley, J. E.; Konigsford, W. L.; Tokubo, S. and Webb, D. A. 1976. Security analysis and enhancements of computer operating systems. *Technical report, DTIC Document*.

[2] Alosefer, Y. and Rana, O.F. 2011. Predicting Client-side Attacks via Behaviour Analysis using Honeypot Data. *In: 7th International Conference on Next Generation Web Services Practices (NWeSP)*. pp. 31-36

[3] Amer, S. and Hamilton Jr, J. 2010.Intrusion Detection Systems (IDS) Taxonomy-A Short Review.*Defense Cyber Security*, 13(2).

[4] Bailey, M.; Oberheide, J.; Andersen, J.; Mao, Z., Jahanian, F. and Nazario, J. 2007. Automated classification and analysis of internet malware.*In: Proceedings of the 10th international conference on recent advances in intrusion detection (RAID'07)*. pp. 178-197.

[5] Bolzoni, D.; Etalle, S. and Hartel, P. 2006. POSEIDON: a2-tieranomaly-based network intrusion detection system. *In: Information Assurance, 2006. IWIA 2006. Fourth IEEE International Workshop on*, pp:10–156

[6] Brauckhoff, D.; Wagner, A. and May, M. 2008. FLAME: a flow-level anomaly modeling engine. *In: Proceedings of the Conference on Cyber Security Experimentation and Test, pp: 1–6.*

[7] Bridges, S. M. and Vaughn, R. B. 2000. Intrusion detection via fuzzy data mining. *In: 12th Annual Canadian Information Technology Security Symposium*, pp: 109–122.

[8] Hoglund, G. and McGraw, G. 2004.*Exploiting Software: How to break code*. Pearson Education India.

[9] Howard, J. D. 1997. An analysis of security incidents on the Internet Technical report, DTIC Document.

[10] Linda, O.; Vollmer, T. and Manic, M.2009. Neural Network based Intrusion Detection System for critical infrastructures Neural Networks. *International Joint Conference on IJCNN*. pp.1827-1834.

[11] Mezghani, D.; Boujelbene, S.; Ellouze, N. 2010. Evaluation of SVM Kernels and Conventional Machine Learning Algorithms for Speaker Identification. *In: International Journal of Hybrid Information Technology*, 3(3).

[12] Yanfang Ye, Tao Li, Qingshan Jiang, and Youyu Wang. 2010. CIMDS: Adapting Post processing Techniques of Associative Classification for Malware Detection. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions* 40(3): 298 – 307.