International Journal of Information, Control and Computer Sciences
ISSN: 2517-9942
Vol:7, No:8, 2013

An Efficient Framework to Build Up Malware Dataset

Madihah Mohd Saudi and Zul Hilmi Abdullah

Abstract**—**This research paper presents a framework on how to build up malware dataset. Many researchers took longer time to clean the dataset from any noise or to transform the dataset into a format that can be used straight away for testing. Therefore, this research is proposing a framework to help researchers to speed up the malware dataset cleaning processes which later can be used for testing. It is believed, an efficient malware dataset cleaning processes, can improved the quality of the data, thus help to improve the accuracy and the efficiency of the subsequent analysis. Apart from that, an in-depth understanding of the malware taxonomy is also important prior and during the dataset cleaning processes. A new Trojan classification has been proposed to complement this framework. This experiment has been conducted in a controlled lab environment and using the dataset from Vx Heavens dataset. This framework is built based on the integration of static and dynamic analyses, incident response method and knowledge database discovery (KDD) processes. This framework can be used as the basis guideline for malware researchers in building malware dataset.

Keywords**—**Dataset, knowledge database discovery (KDD), malware, static and dynamic analyses.

I. Introduction

NOWADAYS we are overwhelmed with lots of noise, missing and inconsistent dataset that need to be clean up prior conducting an analysis. Data cleaning process is needed to clean the dataset by filling in missing values, smoothing noise data, identifying or removing outliers, and resolving inconsistencies. Furthermore, a raw data is rarely can be used directly for learning or mining algorithms. This is the urge the formation of this research. An efficient framework to clean up dataset is introduced to ease the dataset cleaning task.

For this research the scope of the malware dataset is the Trojan dataset. A Trojan is also known as a Remote Administration Tool (RAT). It is a piece of software made for monitoring a system with malicious intention for examples stealing sensitive information such as username and password, credit card number and file deletion [8]. An example of a Trojan is called as a Flame Trojan. On May 2012, this Trojan has infected thousands of computers all over the world and it has been described as one of the most complex threats ever discovered. It has the capabilities to take screenshots secretly, record audio and sends this information to its creator via an encrypted channel. It caused chaos, loss of money and productivity and to certain extent tarnish organization's reputation. Therefore, due to the Trojan bad implications and lack of huge clean dataset of Trojan freely available for further analysis, this is the urge where this research comes in.

Apart from that, the motivations for this research are:

i. The hardness of the researcher or malware analyst to clean up dataset for the subsequent data mining analysis.

Standard procedures to clean up the dataset need to be carried out prior to the analysis part. This is to ensure the results or the outputs produced in the subsequent mining analysis will have a better accuracy and lower false positive rates. The dataset cleanup process is one of the most time consuming and the level of difficulty to clean up the dataset increases if the size of the dataset is large [11], [21].

ii. To get a clean dataset, it consumes a lot of time to process it.

There are various techniques can be used to clean up dataset but which one is the easiest and less time consuming? Many researchers gave up to do the cleanup up dataset since it is time consuming and requiring many man power to do it [21]. To clean up the dataset, the researcher needs to test each sample one by one.

Based on the above motivations, this research paper aims are to produce a framework to build up Trojan dataset and to provide a clean dataset based on the framework proposed. This clean dataset is transformed into the format which can be used for the subsequent analysis for data mining algorithm. Prior the dataset transformation, a new Trojan classification has been produced as part of the data transformation process.

In this research, Trojan dataset was downloaded from Vx Heavens website. By the time this research has completed, the link to this dataset has been taken out. But still the dataset can be referred from many other sources such as by [11], [25]-[28]. The scope of this research is on Windows platform only. There are thousands of Trojans that targeting this platform. Besides that, this research also focused on pre-processing stage, which is preparing a clean Trojan dataset to be used in subsequent analysis such as data mining analysis.

This paper is organized as follows. Section II presents the related works with cleaning dataset. Section III explains the methodology used in this paper which consists of static and dynamic analyses and the architecture of the controlled lab environment. Section IV presents the research finding which consists of a new Trojan classification, the new dataset transformation and machine learning algorithm results and Section V is a closing remarks and summaries the future work of this research paper.

Madihah Mohd Saudi and Zulhilmi Abdullah are with the Faculty Science and Technology (FST), Universiti Sains Islam Malaysia (USIM) (e-mail: madihah@usim.edu.my, zulhilmi.a@usim.edu.my).

II. Related Works

Cases where data is used for data mining directly without any kind of pre-processing are so rare. Data pre-processing seems to constitute an obligatory step, though this step at the same time is time consuming [4].There are a number of data pre-processing techniques. Data cleaning is part of processes involve for data pre-processing, where it can be applied to remove noise and correct inconsistencies in the data. Moreover, data integration merges data from multiple sources into a coherent data store, such as a data warehouse or a data cube. Data transformation, such as normalization, may be applied. For example, normalization may improve the accuracy and efficiency of mining algorithms involving distance measurements.

Malware can be classified based on characteristics including the way of infection and the target of the attack as well as concealment techniques implement by related malware to evade detection [23]. Sophisticated malware use complicated techniques to make security tools difficult to detect. Some of the techniques used by malware to protect them are obfuscation technique, encryption, polymorphism and packing [24].

Graziano et al. [5] included cleaning and normalization processes of the dataset in their malware analysis system as one of the important steps in order to gain a more accurate result. They implemented related processes in large scale environments in order to minimize the false positive. While Barreno et al. [2] stated that it may be difficult to get clean dataset especially in the initial training of malicious data analysis. Same goes with the malware dataset and other form dataset for information security research. Therefore, a proper technique to improve quality of dataset especially in malware analysis is very important.

Data reduction can reduce the data size by aggregating, eliminating redundant features or clustering [6]. After downloading the dataset, the process was started by transforming the Trojan's raw data into an appropriate format. The steps involved in this phase included data cleansing to remove any noise, duplication or outlier and data transformation. Under this process, the static and dynamic analyses were implemented using the incident response standard operating procedures (SOP). SOP is a step by step process and the detail information that the researcher should conduct. Table I below shows the comparison of the research related to the dataset clean up.

III. Methodology

The methods used in this research follow the standard operating procedures (SOP) as shown in Fig. 1.

TABLE I
Comparison Table of Related Work to the Dataset Clean Up

Title and Authors	Method	Domain
PAPER 1: MadihahMohd Saudi, A New Model for Worms Detection and Response[11].	Static and dynamic analyses. • Used feature selection, static analysis, dynamic analysis and data cleaning and transformation. • The five main features of the worm algorithm are being extracted into semiformat structure comprising five different of subareas which are the payload, infection, activation, operating algorithm and propagation to capture the worm characteristic. • Lastly, it is transformed into nominal data with five numeric values.	Worms on Windows platform
PAPER 2: Thomas Stibor, A Study Of Detecting Computer Viruses In Real-Infected Files in the n-gram Representation with Machine Learning Methods [17].	The hexdump. • Step: The hexdump is "cut" into substrings of length n 2 N, denoted as n-grams. The collection is transformed into a vector of dimension5 d = 16n	DOS executable files on DOS and some Windows platforms.
PAPER 3: Luai Al Shalabi, ZyadShaaban and Basel Kasasbeh, Data Mining: A Preprocessing Engine [1].	Normalization: min-max normalization, z-score normalization and normalization by decimal scaling. • HSV data set was normalized using the three methods of normalization. • Used two training data sets for each normalization method. Decision tree methodology for data mining and knowledge discovered was used to test the six training data sets that were designed earlier.	HSV data set from UCI repository.
PAPER 4: RahmatWidiaSembiring&JasniMohamadZain, The Design of Pre-Processing Multidimensional Data Based on Component Analysis [15].	RapidMiner is used for data pre-processing using FastICA algorithm. • Conducted two tests of classification, namely the implementation of pre-processing, FastICA and clustering, and compared the results with no pre-processing.	Wisconsin breast cancer datasets, lung cancer datasets and prostate cancer datasets.
PAPER 5: Thomas Zimmermann and Peter Weißgerber, Preprocessing CVS Data for Fine-Grained Analysis [22].	Functions sections approach • The extraction calls the CVS *log* command in the root directory of the project to be extracted. • In mapping of changes to fine-grained entities, each revision is decomposed into its building blocks, and then a diff between the two revisions r1 and r2 is calculated. • The result is used to create the set. • Next, each line is mapped to its enclosing function. • In data cleaning, they filter out transactions of size greater N in the analysis	CVS archives

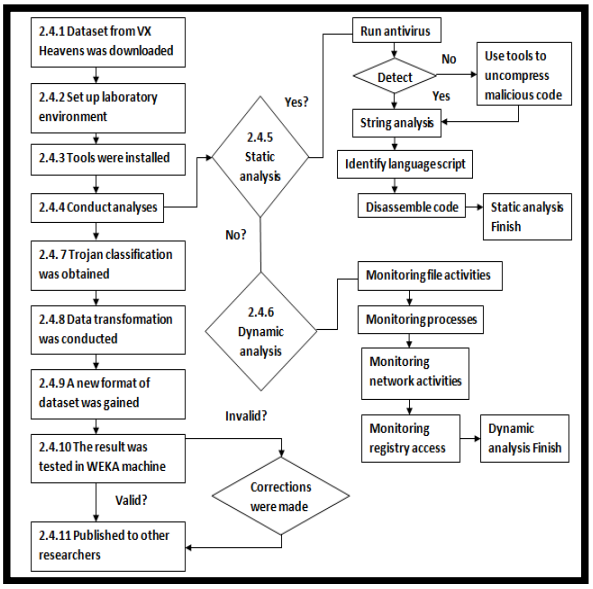
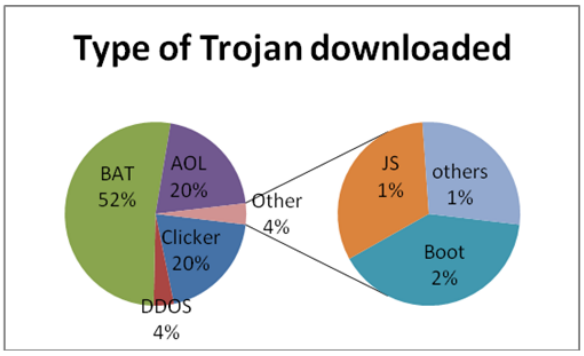


Fig. 1 A framework to build the Trojan dataset

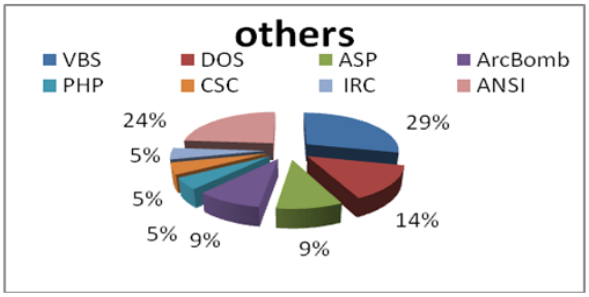
A. Dataset

The first step was, downloading the dataset from VX Heavens website. All Trojan and variants were downloaded to be tested. However, only the Trojan from Windows platform was chosen. For the domain of this research, Windows platform was chosen due to more attacks and vulnerabilities exploited in Windows platform discovered [7]. In addition, the amounts of Trojans that attack on other platforms are fewer than Windows. From a survey on the Internet provided by http://www.esecurityplanet.com (2012), in comparing with Linux platform, Windows has the reputation for being the worse due to there have been more accounts of Windows being under the attack of worms, viruses and Trojans. Basically the problems are because Windows is poorly coded, so Windows is a bigger target compared to Linux. Because of a lot of attacks, hence there are a lot of Trojan appeared from Windows. Fig. 2 shows the different types of Trojan downloaded. The dataset in this research consists of different types of Trojans source from VX Heavens. From 1,982 samples of Trojan downloaded from VX Heavens, the categories are: Clicker, DDOS, BAT, AOL, Boot, ASP, ANSI, ArcBomb, CSC, IRC, JS, PHP, VBS, and DOS.

There are several reasons why this research chose to gather data from the VX Heavens source; firstly, many studies have used this data for their testing, for example from Dai et al. [3] and Stibor [17]. The second reason is because the variants are more important than the quantity of the datasets, since these already represent different types of Trojan in VX Heavens and the third is due to the scope of this research, which only focuses on Windows Trojan. Other researchers that used the dataset from VX Heaven are from Nataraj et al. [12], Shafiq et al. [16], Dai et al. [3], and Mohd Saudi et al. [10].



(a) The whole trojan dataset



(b) Details for others trojan dataset

Fig. 2 Different type of trojan dataset (a) The whole trojan dataset (b) Details for others trojan dataset

B. Research Environment

The researcher set up the controlled laboratory with 2 computers which are installed with VMWare. The lab is built up in a controlled lab environment, separated from the production network. Fig. 3 shows the architecture of the lab.

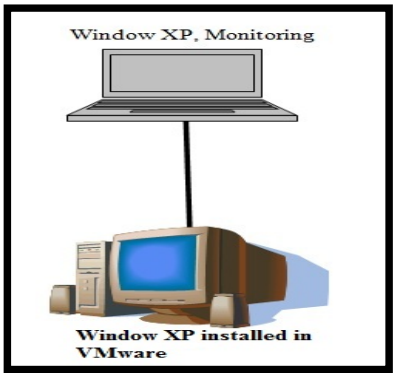


Fig. 3 Lab architecture

The following in Table II is the lists of the tools used in this lab. Almost 80% of the tools used are an open source software or free basis.

TABLE II
TOOLS AND THEIR FUNCTIONALITIES

Functions	Tools	Purpose of Action
Scan tool	AVG antivirus	To prepare the scan tool to detect various forms of malicious code including those with newer signatures
String research tool	String.exe (from Sysinternal)	To display and extract suspicious sets of ASCII characters included in a file
Unpack tool	Proc dump 4.01 Unpack tool UPX tool	To decompress and unpack the Trojan code
Virtual PC	VMWare Work Station	To allow multiple operating systems to run on a single computer.
TCP view	TCPView	TCPView is a Windows program that will show detailed listings of all TCP and UDP endpoints on system, including the local and remote addresses and state of TCP connections.
Disassembler /Debug Tool	OllyDbg	To perform detailed code analysis.
Process Monitoring	Prcview v 3.7.3.1 Process Explorer(from Sysinternal)	To identify the resources used by all running processes, including DLLs and registry keys. Process explorer provides a wealth of useful information regarding how the Trojan was impacting upon the victim computer

IV. FINDINGS

The followings are the findings of this research. A new Trojan classification is produced to ease the data transformation process. The output of the new clean dataset is transformed into nominal dataset which can be used for the subsequent analysis using the data mining algorithm [9]. WEKA an open source software is the software used to do the mining analysis.

A. New Trojan Classification

The researcher produced a new Trojan classification after reviewing previous Trojan classification from different researches [13], [14], [18]-[20].The new Trojan classification was used to classify each of Trojan based on the characteristics as shown in Fig. 4. A study by Abuzaid et al. [29] used infection, activation, payload and operating algorithm as the main characteristics for their Trojan classification. But the result of the accuracy rate of the classification has not been discussed yet. In contrast with this paper, the accuracy rate of the Trojan classification is detailed explained under section *C.*

B. Data Transformation

The Trojan classification was noted with certain number representation to be transformed into nominal data. The researcher identified the nominal data for each of Trojan. Fig. 5 shows the data transformation details. After conducting data transformation for each of Trojan, the researcher came out with a clean Trojan file that was compatible in WEKA machine. Fig. 6 shows the new format of dataset.

The new file is an arff format. The arff file was tested as an input in WEKA machine to validate. Fig. 7 shows the arff file was uploaded into WEKA machine.

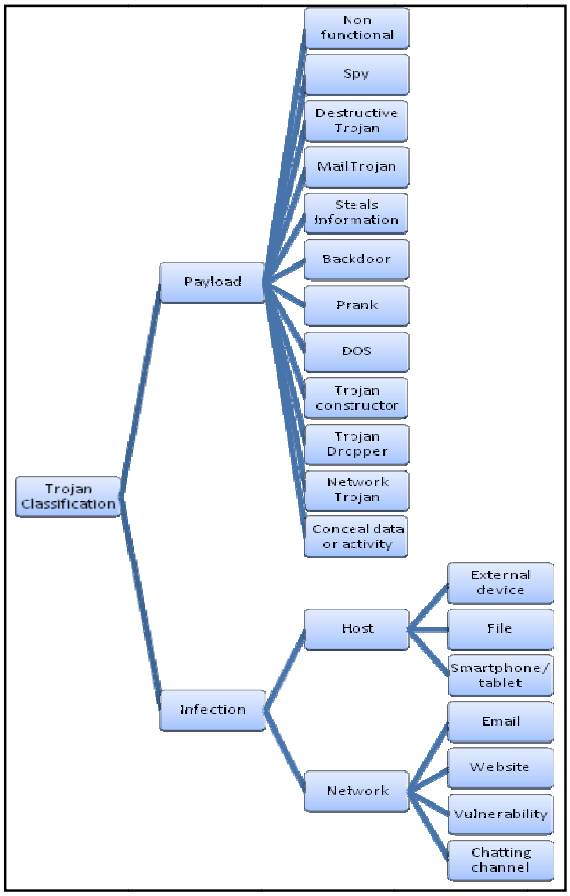


Fig. 4 A new trojan classification

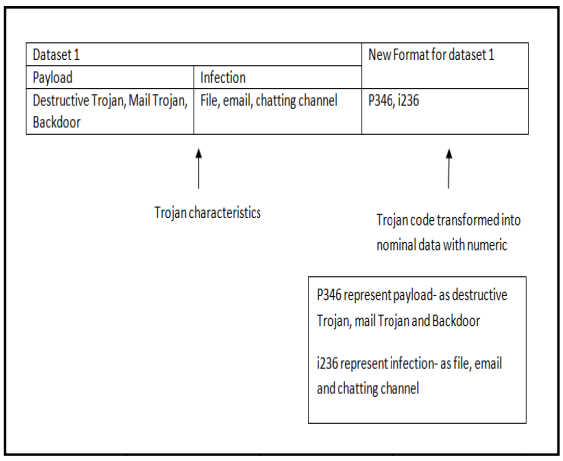


Fig. 5 Data Transformation (using certain number representation)

```
@relation Trojan_characters

@attribute Instance_number numeric
@attribute Payload {p1,p2,p3,p4,p5,p6,p7,p8,p9,p20,p30,p40}
@attribute Infection {i1,i2,i7,i3,i4,i5,i6}


@data
p5 , i2
p3 , i2
p5 , i25
p5 , i2
p4 , i36
p4 , i2
p346 , i36
p4 , i36
p7 , i2
p30 , i24
p30 , i24
```

Fig. 6 Arff format of Trojan dataset

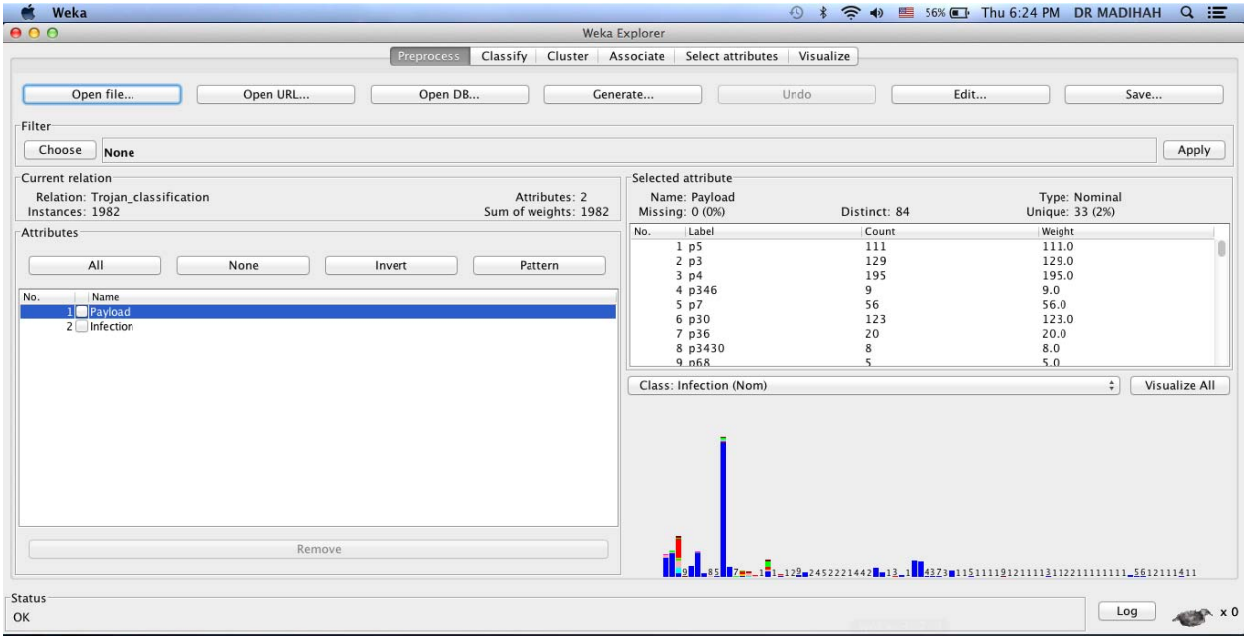


Fig. 7 Arff file was tested in WEKA machine

The purpose of this figure is to show how the data can be mined in WEKA software. Later, the data can be analyzed using different machine learning algorithm inside the WEKA.

C. Machine Learning Algorithm Results

This section presents the finding results of the dataset built from the framework proposed in this research paper. The proposed framework is called as EFMaD which stands for Efficient Framework to build Malware Dataset. Sequential Minimal Optimization (SMO) algorithm is chosen to classify the dataset and the result as displays in Table II. Another similar work but with different malware classification by Mohd Saudi *et al* [10] is compared with. This work used the same source of the dataset. True positive rate (TPR) and false positive rate (FPR) are used during the experiment. The experiment was conducted using the WEKA software.

TABLE II
MACHINE LEARNING ALGORITHM RESULTS

Classifier	EFMaD Results (%)		Comparison work (%)	
	TPR	FPR	TPR	FPR
SMO	98.8	1.3	98.1	0.2

TPR represents True Positive Rate, FPR represents False Positive Rate.

Table I shows the result of TPR for EFMaD is 0.7% higher compared with the existing work [10], which indicates a good result. Though the FPR of EFMaD is 1.1% higher than compared work, there is always room of improvement to lower the FPR in future.

V. CONCLUSIONS AND FUTURE WORKS

As a conclusion, this research managed to provide a new clean Trojan dataset to be published to help other researchers in

data mining research. The Trojan dataset was presented in nominal data which was compatible to be used directly in WEKA machine learning algorithm for data mining process. Furthermore, based on the experiment conducted using WEKA software, the TPR of the classified data which is 98.8% is produced. This result can be used as a reference and comparison by other researchers with the same interests.

For future work, different machine learning algorithms will be tested to the dataset produced from this research. Apart from that, the dataset produced in this research paper will be uploaded in the website so other researcher with the same interest can use the dataset and framework introduced. This paper is part of a larger project to build up an automated malware clean up model. Ongoing research will include other malware classification and the development of software to automate the malware dataset cleanup.

ACKNOWLEDGMENT

The authors would like to express their gratitude to UniversitiSains Islam Malaysia (USIM) for the support and facilities provided. This research paper is supported under USIM's grant [PPP/FST/SKTS/30/12812].

REFERENCES

[1] Al Shalabi, Luai., Syaaban, Zyad., & Kasasbeh, Basel. (2006). Data Mining: A Preprocessing Engine. Applied Science University, Amman, Jordan (Electronic version). (Accessed 25 March 2013).

[2] Barreno, M., Bartlett, P. L., Chi, F. J., Joseph, A. D., Nelson, B., Rubinstein, B. I., ... & Tygar, J. D. (2008, October). Open problems in the security of learning. In *Proceedings of the 1st ACM workshop on Workshop on AISec*(pp. 19-26). ACM.

[3] Dai, Jianyong., Guha, Ratan., & Lee, Joohan. (2009). Efficient Virus Detection Using Dynamic Instruction Sequences(Electronic version). (Accessed 29 March 2013).URL:http://www.academypublisher.com /jcp/vol04/no05/jcp0405405414.pdf

[4] Engels, Robert., Theusinger, Christiane. (1998). Using a Data Metric for Preprocessing Advice for Data Mining Applications(Electronic version).(Accessed 27 March 2013).URL:http://www.esis.no/people/ robert.engels/papers/engels_theusinger_ECAI98.pdf

[5] Graziano, M., Leita, C., & Balzarotti, D. (2012, December). Towards network containment in malware analysis systems. In *Proceedings of the 28th Annual Computer Security Applications Conference* (pp. 339-348). ACM.

[6] Han, J., Kamber, M.(2000). Data Preprocessing (Electronic version). (Accessed 28 March 2013).URL:http://www.cse.iitm.ac.in/~cs672/ Lectures/Data_Preprocessing.pdf.

[7] Is Linux really more secure than Windows? (2011) [online] Available from:http://www.esecurityplanet.com/trends/article.php/3933491/Is-Lin ux-Really-More-Secure-than-Windows.htm (accessed 29 March 2013).

[8] Mangarae, Aelphaeis.(2006) Trojan White Paper [Igniteds.NET], Available from: http://igniteds. (Accessed 29 March 2013).

[9] Mertz, C.J. and Murphy, P.M. (1996). UCI Repository of machine learning databases. University of California (Electronic version). Available from: http://www.ics.uci.edu/~mlearn/MLRepository.htm (Accessed 29 March 2013).

[10] Mohd Saudi, M., Cullen, A.J. and Woodward, M. (2011), Efficient StakcertKdd Processes In Worm Detection, World Academy Of Science, Engineering And Technology Journal, Issue 55, pp. 453-457.

[11] Mohd Saudi, Madihah. (2011). A New Model for Worms Detection And Response (Electronic version). (Accessed 25 March 2012).

[12] Nataraj, Lakshmanan., Yegneswaran, Vinod., Porras, Phillip., & Zhang, Jian. (2011). A Comparative Assessment of Malware Classification using Binary Texture Analysis and Dynamic Analysis(Electronic version). (Accessed 26 March 2013).URL: http://vision.ece.ucsb.edu/publications/aisec17-nataraj.pdf.

[13] Plusquellic, Jim.,(2008). Taxonomy of Trojans for IC Trust. (Electronic version). (Accessed 13 May 2012). URL: http:// www.ece.unm.edu/~jimp/HOST/papers/Trojan_taxonomy.pdf.

[14] Rajendran, Jeyavijayan., (2011). Toward a Comprehensive and Systematic Classification of Hardware Trojans. (Electronic version). (Accesses 29 March 2013).

[15] Sembiring, Rahmat Widia., & Mohamad Zain, Jasni. (2012). The Design of Pre-Processing Multidimensional Data Based on Component Analysis, Faculty of Computer System and Software Engineering, Universiti Malaysia Pahang (Electronic version). (Accessed 29 March 2013).URL: http://umpir.ump.edu.my/1204/1/new1-20110414.pdf.

[16] Shafiq, M. Zubair.,Khayam, Syed Ali., & Farooq, Muddassar. (2008). Embedded Malware Detection using Markov n-grams(Electronic version). (Accessed 29 March 2013).URL: http://nexginrc.org/ nexginrcAdmin/PublicationsFiles/dimva08-zubair.pdf.

[17] Stibor, Thomas. (2010). A Study Of Detecting Computer Viruses In Real-Infected Files in the n-gram Representation with Machine Learning Methods (Electronic version). (Accessed 29 March 2013). URL:http://www.sec.in.tum.de/assets/staff/stibor/iea.aie.final.extended.p df

[18] Tehranipoor, Mohammad.,Koushanfar, Farinaz, (2010). A Survey of Hardware Trojan Taxonomy and Detection. . (Electronic version). (Accessed 29 March 2013).URL: http://www.computer.org/ portal/web/computingnow/0910/theme/designandtest3

[19] Trojan Horse (2012) [online] Available from: www.webopedia.com /TERM/T/Trojan_horse.html (Accessed 29 March 2013).

[20] Wang,Xiaoxiao.,Salmani, Hassan., Tehranipoor, Mohammad., and Plusquellic, Jim. (2008). Hardware Trojan Detection and Isolation Using Current Integration and Localized Current Analysis (Electronic version).(Accessed on 29 March 2013) .URL:http://www.ece.unm.edu /~jimp/pubs/DFT08_FINAL.pdf

[21] Witten, I. H., & Frank, E. (2005). *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann.

[22] Zimmermann, Thomas., & Weißgerber, Peter. (2004). Preprocessing CVS Data for Fine-Grained Analysis (Electronic version). (Accessed 26 March 2013).URL: http://msr.uwaterloo.ca/slides/Zimmermann.pdf

[23] Rad, B. B., Masrom M., Ibrahim, S. (2011). Evaluation of Computer Virus Concealment and Antivirus Concealment and Anti-Virus Techniques: A Short Surver.*International Journal of Computer Science Issues*, 8(1).

[24] Gharibi, W., Mirza, Abdulrahman. Software Vulnerabilities, Banking Threats, Botnets and Malware Self-Protection Technologies.

[25] Schultz, M. G., Eskin, E., Zadok, E. and Stolfo, S. J. (2001). Data Mining Methods for Detection of New Malicious Executables. *In Proceedings of the 2001 IEEE Symposium on Security and Privacy*, IEEE Computer Society, pp 38, (Accessed 26 March 2013)

[26] Henchiri, O. and Japkowicz, N. (2006). A Feature Selection and Evaluation Scheme for Computer Virus Detection. *Proceedings of the Sixth International Conference on Data Mining, 2006. ICDM '06.* Hong Kong: IEEE Xplore, pp. 891 - 895. Available from: http://doi.ieeecomputersociety.org/10.1109/ICDM.2006.4(Accessed 26 March 2013)

[27] Moskovitch, R., Y. Elovici and Rokach,L. (2008a). Detection of unknown computer worms based on behavioral classification of the host. *Computational Statistics & Data Analysis* 52(9). pp.4544-4566.

[28] Khan, H., Mirza, F. and Khayam, S. A. (2010). Determining malicious executable distinguishing attributes and low-complexity detection. *Journal in Computer Virology*. 7(2), pp. 95-105

[29] Abuzaid, AM, Mohd Saudi, M. M Taib, B. & Abdullah, ZH. (2013) An Efficient Trojan Horse Classification (ETC*), IJCSI International Journal of Computer Science Issues*, Vol. 10, Issue 2, No 3, March 2013, pp.96-104.