Paper: ASAT-16-130-CE

16th International Conference on
AEROSPACE SCIENCES & AVIATION TECHNOLOGY,
ASAT - 16 – May 26 - 28, 2015, E-Mail: asat@mtc.edu.eg
Military Technical College, Kobry Elkobbah, Cairo, Egypt
Tel : +(202) 24025292 – 24036138, Fax: +(202) 22621908

Developing Anti Spyware System using Design Patterns

Mohamed Adel Sheta*, Mohamed Zaki†, Kamel Abd El Salam El Hadad ‡ and H. Aboelseoud M.§

Abstract: Spyware is considered a great threat to confidentiality since it can cause loss of control over private data for computer users. This kind of threat might select some data and send it to another third party without the consent of the user. Spyware detection has been presented traditionally by three approaches signature based detection, behavior based detection and specification based detection. These approaches were successful in detecting known spyware but have proven failure in detecting unknown spyware. In this paper we introduce a framework for anti spyware system that combine data mining and design patterns in detecting and classifying spyware. The proposed anti spyware system can be reusable as it can modify itself for any new or modified spyware.

Keywords: Spyware, Data mining, Design patterns.

1. Introduction

Malicious codes (malware) which were designed by hackers include multiple attacking methods such as data intercept and denial of service (DoS) attacking. These malicious codes spread using the weak spot of certain program, and zero day attack is expected. Malware contains virus, worm, Trojan horse and spyware [1].

First, virus is a contagious computer program that can copy itself and infect another computer. It spreads from one computer to another in some form of executable code when its host is taken to the target computer; for example when a user sends it over a network or the internet or carried it on a removable medium [2]. Second, worm is a self replicating malware computer program. It spreads through a computer network by sending copies of itself to other computers on the network and it may do so without the interference of any user. Unlike a virus, it does not need to be attached to any program. Worms almost always cause high network traffic, even if only by consuming bandwidth, whereas viruses almost always damage or modify files on an infected computer [3].

* Ph.D. Student, Department of Computer Engineering, Military Technical College, Egypt, mohshe2007@hotmail.com.
† Prof. of Computer and System Engineering, Al-Azhar University, Egypt, azhar@eun.eg.
‡ Dr., Department of Computer Engineering, Military Technical College, Egypt, kamel_elhadad@hotmail.com.
§ Dr., Department of Computer Engineering, Military Technical College, Egypt, h_aboelsoud@yahoo.com.

Third, Trojan horse is a malware that seems to perform a desirable function for the user before run or install but instead facilitates illegal access of the user's computer system. "It is a harmful piece of software that looks legitimate. Users are typically tricked into loading and executing it on their systems", as Cisco describes. The term is originated from the Trojan horse story in Greek mythology [4]. Finally, according to the Department of Computer Science and Engineering at the University of Washington, spyware is defined as "software that gathers information about use of a computer, usually without the knowledge of the owner of the computer, and relays the information across the internet to a third party location". Another definition of spyware is given as "any software that monitors user behavior, or gathers information about the user without adequate notice, consent, or control from the user" [5].

Unlike viruses, spyware is usually installed with the user's approval, since it provides some useful functionality either on its own or by another software application. That's why spyware extends beyond the boundaries of what is considered legal and illegal software and thus falls in a grey zone. The installed spyware may be capable of capturing keystrokes, taking screenshots, saving authentication credentials, storing personal email addresses and web form data, and thus may obtain behavioral and personal information about users [5]. It may also communicate system configuration including hardware and software, system accounts, location information, and information about other aspects of the system to a third party. Spyware may, e.g., show characteristics like nonstop appearances of advertisement pop-ups, open a website or force the user to open a website which has not been visited before, install browser toolbars without seeking acceptance from the user, change search results, make unexpected changes in the browser, display error messages, and the occurrence of network traffic without any request from the user.

In this paper, a spyware system with data mining and design patterns will be introduced for detecting and classifying the new or modified spyware types. This proposed approach can be reusable as it can modify itself for any new or modified spyware. This paper is organized as follows; related work and design patterns are discussed in section 2 and section 3, respectively. The proposed approach and the system architecture are explained in section 4 and section 5, respectively. The conclusion and future work are discussed in section 6.

2. Related work

Spyware detection techniques are used to detect the spyware and prevent the infection of the computer system. They can be categorized into signature based detection, behavior based detection, specification based detection, and data mining based detection [3] that will be discussed in the next sections.

2.1 Signature based detection

Signature based detection detects spyware by comparing the spyware signature to the database. These signatures are created by examining the disassembled binary code of spyware. Disassembled code is analyzed and features are extracted. These features are used to construct the signature of a certain spyware family [2]. The main advantages of this technique is that it can accurately detect known spyware and using less amount of resources to detect the spyware because it mainly focus on signature of attack. The main disadvantage is that it is unable to detect the new unknown spyware as there is no available signature for this new type of spyware [6].

2.2 Behavior based detection

The function of behavior based detection is to analyze the behavior of known or unknown spyware. It usually occurs in two phases: training phase and detection phase. During training phase the behavior of the system in the ideal state is observed and machine learning technique is used to create a profile of such normal behavior. The detection phase is the comparison of the current system behavior after attack to the normal behavior and differences are flagged as potential attacks [7]. The main advantage of this technique is that it is able to detect known as well as new unknown instances of spyware because it focuses on the behavior of system to detect unknown attack. The main disadvantage of this technique is that it constantly needs to update the data describing the system behavior. It needs more resources like CPU time, memory and disk space as well as level of false positive is high [8].

2.3 Specification based detection

Specification based detection is derivative of behavior based detection that tries to overcome the high false positive rate associated with it. Monitoring programs are involved in executions and detecting deviation of their behavior from the specification, rather than detecting the occurrence of specific attack patterns [9]. The main advantage of this technique is that it can detect known as well as new unknown instances of spyware and level of false positive is low. The main disadvantages of this technique are the level of false negative is high, not as effective as behavior based detection in detecting new attacks and development of detailed specification is time consuming.

2.4 Data mining based detection

Data mining has been the main focus of many spyware researchers for detecting the new unknown spyware. They have added data mining as a fourth proposed spyware detection technique [3]. Data mining helps in analyzing the data with automated statistical analysis techniques, by identifying meaningful patterns or correlations. The results from this analysis can be summarized into useful information and can be used for prediction. Machine learning algorithms are used for detecting patterns or relations in data which are further used to develop a classifier. The current anti spyware systems suffer from some drawbacks such as; the need for updating data describing the system behavior to detect new unknown or modified spywares, and the high level of false positive & false negative. Data mining is capable of detecting new unknown or modified spyware with high detection rate compared to signature based, behavior based, and specification based detection methods [5].

Raja K. *et al.* [5] detected spyware by using data mining method with a byte sequence mining approach. The experiment applied on low amount of data set contained 137 files. Out of these, 18 files were spyware and 119 files were benign. Feature sets generated by Common Feature Based Extraction (CFBE) selection method produced better results than feature sets generated by Frequency Based Feature Extraction (FBFE). The overall detection accuracy (ACC) was 90.5% and the Area under Receiver Operating Characteristic Curve (AUC) score of 0.83.

Raja K. *et al.* [10] detected adware by using data mining method with an opcode sequence extraction mining approach. The experiment applied on low amount of data set contained 600 files. Out of these, 300 files were spyware and 300 files were benign and AUC was equal to 0.949. The Malware File Percentage (MFP) is equal to 50% and it needed to be <=15% of the total population in order to yield a high prediction performance [11].

Raja K. and Niklas L. [12] detected scareware by using data mining method with a variable length instruction sequence mining approach. The experiment applied on low amount of data set contained 800 files. Out of these, 550 files were scareware and 250 files were benign .The results were AUC equal to 0.972 and low false negative rate of 2.3%. MFP is equal to 68.8%.

Zongqu Z. *et al.* [13] detected malware by using data mining method based on the control flow of software mining approach. The experiment applied on higher amount of data set contained 9398 files. Out of these, 4828 files were malware and 4570 files were benign. The results were ACC equal to 97%, AUC score of 0.993 and low false positive rate equal to 3.2%. MFP is equal to 51.4%.

2.5 Comparison

This section summarizes in Table 1 a comparative study between the related works.

Table 1. Comparison of related works

Study	Feature Extraction	Feature Selection	Classifiers
Detection of Spyware by Mining Executable Files [5].	Byte sequence n-grams	CFBE and FBFE	ZeroR, Naive Bayes, SMO, J48, JRip Random Forest
Accurate Adware Detection using Opcode Sequence Extraction [10].	Opcode n-grams	TF-IDF, CPD	ZeroR, Naive Bayes, SMO, IBk,J48,JRip
Detecting Scareware by Mining Variable Length Instruction Sequences [12].	Opcode n-grams	TF-IDF, CPD	JRip, SMO, DT, IBk, NB, Random Forest
Malware Detection Method Based on The Control-flow Construct Feature of Software [13].	Opcode	VSM	J48, Bagging, Random Forest

3. Design patterns

Software design patterns are solutions to problems that arise regularly during software design. They are meant to serve as readily applicable, time saving strategies for software development. The structured documentation that accompanies a properly defined pattern allows developers to quickly identify and apply patterns to a given problem. Design patterns are a general reusable solution to a commonly occurring problem within a given context in software design. A design pattern is not a finished design that can be transformed directly into code. It is a description or template for how to solve a problem that can be used in many different situations.

The idea of a design pattern was developed by Christopher Alexander in his work on reusable strategies for architecting space and structure. Each pattern describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice [14]. Design patterns are classified into three main categories depending on their functions; creational patterns, structural patterns and behavioral patterns. Creational patterns encapsulate knowledge about which concrete classes the system uses and hide how instances of these classes are created. Structural patterns are concerned with how classes and object are composed to form larger structures. Behavioral patterns are concerned with algorithms and the assignment of responsibilities between objects and use inheritance to distribute behavior between classes. In Table 2, all types of design patterns related to their categories are summarized and it discussed with more details in [14].

One challenge for the current anti spyware systems is the need of changing the structure of old anti spy programs. Strategy design pattern is capable of defining a family of algorithms, encapsulating each one, and making them interchangeable [15]. Thus, the system using this design pattern will be able to specify the strategy of classifying the spyware as a main class to its subclasses for each spyware type. It's also being reusable for a new spyware family that can be built and considered as a new subclass of spyware without changing the main program structure. Also, Factory design pattern provides an interface for creating an object, but it leaves choice of object's concrete type to a subclass [15]. Therefore, when a file is classified as new unknown or modified spyware the system will create a new object according to its type so the system will update itself.

Table 2. Types of design patterns

Creational	Structural	Behavioral
Factory	Adapter	Interpreter
Abstract Factory	Bridge	Template Method
Builder	Composite	Chain of Responsibility
Prototype	Decorator	Command
Singleton	Facade	Iterator
	Flyweight	Mediator
	Proxy	Memento
		Observer
		State
		Strategy
		Visitor

4. The proposed anti spyware approach

In the proposed approach data mining will be combined with design patterns method in an integrated anti spyware system. Data mining is able to detect new unknown or modified data, moreover the reusable advantage of the design patterns [14]. So that, in the proposed system there is no need for changing the structure of the current anti spyware programs in case of a new spyware type is existed.

4.1 The framework

The proposed anti spyware framework will be consists of three layers as shown in Fig. 1. First, the lowest layer is the selected design patterns which will be suitable in the case of anti spyware containing the strategy and factory design patterns. Second, the middle layer is the security patterns that include the traditional methods for spyware detection as feature extraction (i.e., byte sequence n-grams and opcode n-grams) and selection (i.e., FBFE and VSM). Finally, the highest layer is the new anti spyware patterns for each type of spyware which combined data mining and design patterns for detecting new unknown as well as modified spyware.

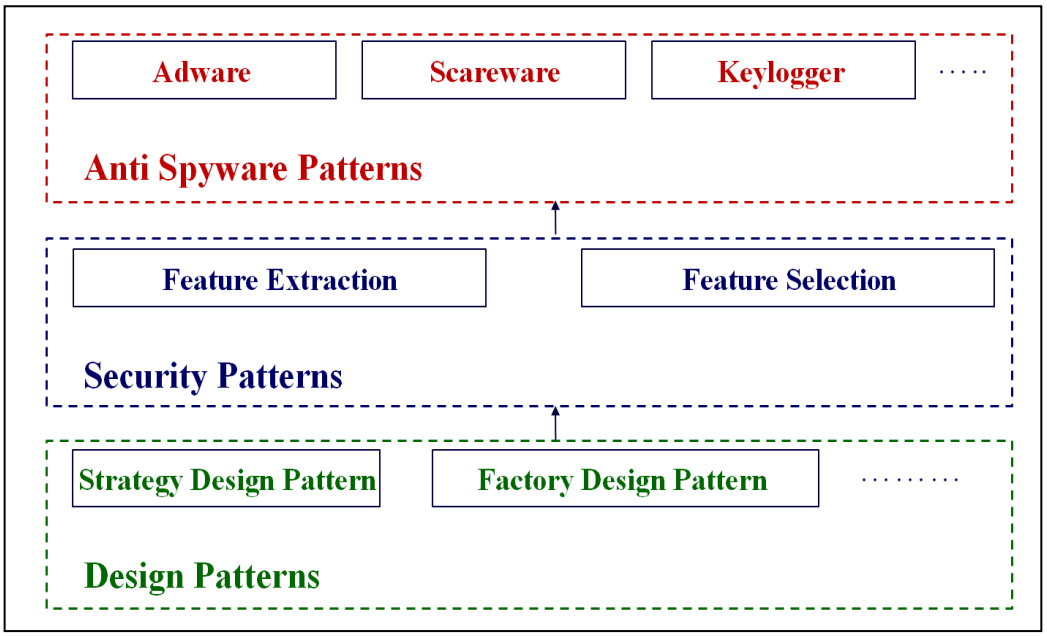
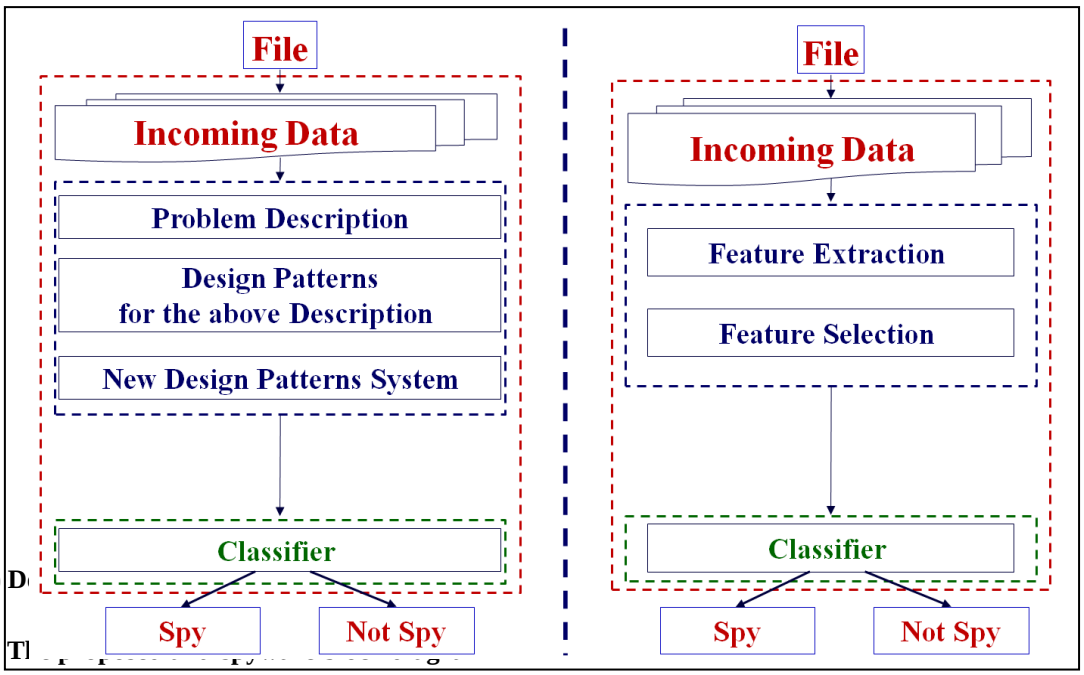


Fig. 1. The proposed anti spyware framework

4.2 The block diagram

In the proposed anti spyware block diagram we used the design patterns approach starting with the problem description then choosing the corresponding design patterns to this problem and after that building a new design patterns system to train the classifier for spyware detection as shown in Fig. 2(a). In the traditional, anti spyware block diagram started with feature extraction then feature selection before training the classifier as shown in Fig. 2(b). Note that the feature extraction and selection in the traditional anti spyware block diagram will be included in the stage of choosing the corresponding design patterns of the proposed approach.



(a) D

Fig. 2. Th

5. The system architecture

In the proposed anti spyware system architecture the data mining is combined with the design patterns techniques as shown in Fig. 3.

5.1 Classifier training offline

Fig. 3 (a) shows the classifier training / testing phase that is done in offline state.

5.1.1 Data set mining

Our data set consists of benign and spyware binaries of different families. The benign files were collected from Download.com [16], which certifies the files to be free from spyware. The spyware files were downloaded from the links provided by SpywareGuide.com [17], which hosts information about different types of spyware and other types of malicious software. First this data set is represented to extract and select the features of spyware. Then these extracted and selected features are used to produce a new training data set.

5.1.2 Classifier learning

The major part of the training data set is used for training the classifier to classify the input as a benign or one of spyware family or type. The remaining part of the training set will be used to verify or test the classifier performance.

5.2 Classifier execution online

Fig. 3 (b) shows the classification phase that is done in online state. After the validation steps of the classifier is done in the testing step in Fig. 3 (a), the classifier is used to classify the input file after representation of the data in the format that is predefined to the classifier before.

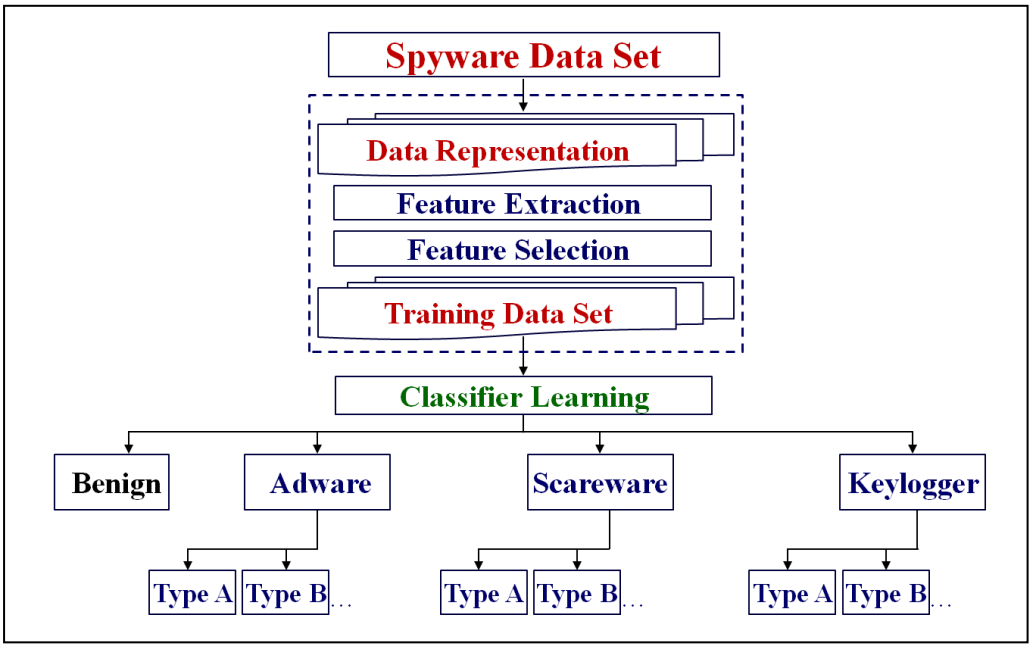
5.2.1 Factory design pattern

First, if the input data file is known to the classifier then the classifier will detect its type from the predefined spyware types known to the system. Second, if the input data file is a modified spyware type from a spyware family then the classifier will detect this type of the input data and the factory design pattern will classify this modified spyware type to its related predefined spyware family.

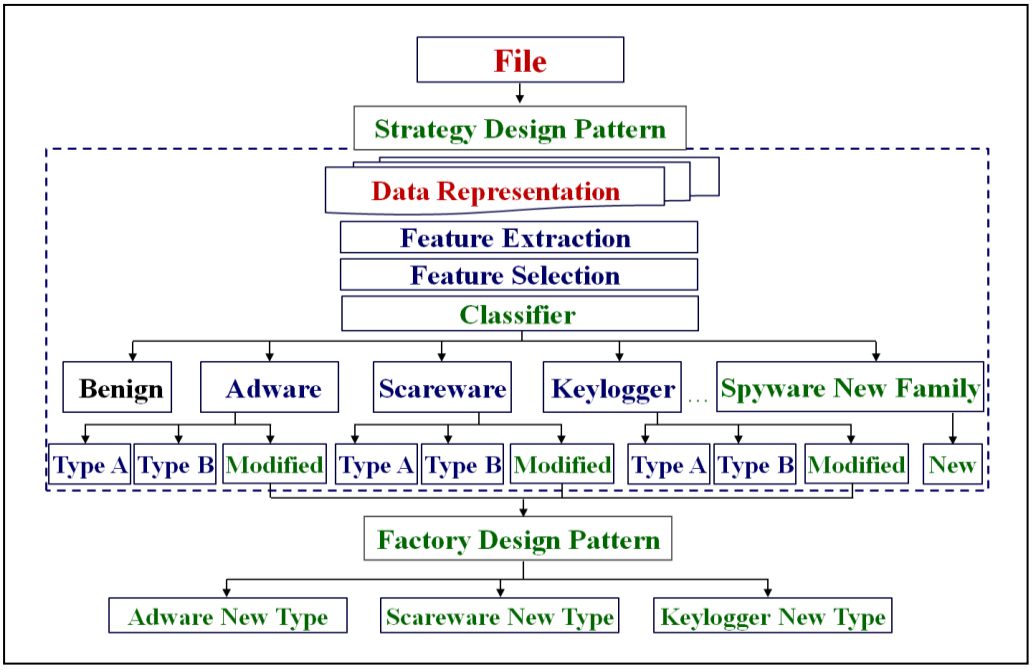
Finally, if the input data file is a new spyware type then the classifier will detect this type of the input data as unknown spyware type then the factory design pattern will classify this new spyware type as a new spyware family.

5.2.2 Strategy design pattern

This new family will be inherited in the design of the whole anti spyware system using the strategy design pattern. The modified or new type of the spyware will be included in the system spyware database to be reused for the next detection of spyware. The proposed solution can modify itself in case of new or modified spyware to be detected.



(a) Classifier training / testing phase (offline)



(b) Classification phase (online)

Fig. 3. The proposed anti spyware system architecture

6. Conclusion and future work

In this paper we proposed anti spyware system based on data mining combined with design patterns for detecting and classifying spywares. The classification of new unknown

or modified spyware will create a new object according to its type so the system will update itself. The proposed anti spyware system is reusable in which it can modify itself in case of new spyware so that new unknown as well as modified spyware can be detected. The proposed anti spyware system based on data mining combined with design patterns will be implemented as a future work to proof this new idea and to compare the experimental results with the previous anti spyware methods based on data mining listed in the related work.

7. References

[1] G. Padmavathi, and S. Divya "A Survey on Various Security Threats and Classification of Malware Attacks, Vulnerabilities and Detection Techniques", *The International Journal of Computer Science & Applications (TIJCSA)*, Vol. 2, pp. 66-72, India, 2013.

[2] Donghwi Lee, Won Hyung Park, and Kuinam J Kim "A Study on Analysis of Malicious Codes Similarity Using N-Gram and Vector Space Model", *IEEE International Conference on information and applications (ICISA)*, pp. 1-4, Republic of Korea, 2011.

[3] Jyoti Landage, and Wankhade "Malware and Malware Detection Techniques: A Survey", *International Journal of Engineering Research & Technology (IJERT)*, Vol. 2, pp. 61-68, India, 2013.

[4] Mohamad Fadli Zolkipli, and Aman Jantan "A Framework for Malware Detection Using Combination Technique and Signature Generation", *IEEE International Conference on Computer Research and Development*, pp. 61-68, Malaysia, 2010.

[5] Raja Khurram Shazhad, Syed Imran Haider, and Niklas Lavesson "Detection of Spyware by Mining Executable Files", *IEEE International Conference on Availability, Reliability and Security (ARES)*, pp. 295-302, Sweden, 2010.

[6] Kai Huang, Yanfang Ye, and Qinshan Jiang "ISMCS: An Intelligent Instruction Sequence based Malware Categorization System", *IEEE International Conference of Anti-counterfeiting, Security, and Identification in Communication*, pp. 509-501, China, 2010.

[7] Mohammad Wazid, Avita Katal, R.H. Goudar, D.P. Singh , and Asit Tyagi "A Framework for Detection and Prevention of Novel Keylogger Spyware Attacks", *IEEE International Conference on Intelligent Systems and Control (ISCO)*, pp. 433-438, India, 2012.

[8] Karan Sapra, Benafsh Husain, Richard Brooks, and Melissa Smith "Circumventing Keyloggers and Screendumps", *IEEE International Conference on Malicious and Unwanted Software*, pp. 103-105, USA, 2013.

[9] Raihana Md Saidi, Siti Arpah Ahmad, Noorhayati Mohamed Noor, and Rozita Yunos "Windows Registry Analysis for Forensic Investigation", *IEEE International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE)*, pp. 132-136, Malaysia, 2013.

[10] Raja Khurram Shahzad, Niklas Lavesson, and Henric Johnson "Accurate Adware Detection using Opcode Sequence Extraction", *IEEE International Conference on Availability, Reliability and Security (ARES)*, pp. 189-195, Czech Republic, 2011.

[11] Asaf Shabtai, Robert Moskovitch, Clint Feher, Shlomi Dolev, and Yuval Elovici "Detecting unknown malicious code by applying classification techniques on opcode patterns", *Springer on Security Informatics*, Germany, 2012.

[12] Raja Khurram Shahzad and Niklas Lavesson "Detecting scareware by mining variable length instruction sequences", *IEEE International Conference on Information Security South Africa (ISSA)*, pp. 1-8, South Africa, 2011.
[13] Zongqu Zhao, Junfeng Wang, and Jinrong Bai1 "Malware detection method based on the control-flow construct feature of software", *International Journal of The Institution of Engineering and Technology (IET) on Information Security*, Vol. 8, pp. 18-24, England, 2013.
[14] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, *Design Patterns: Elements of Reusable Object-Oriented Software*, Boston, Massachusetts, Addison-Wesley Longman Publishing Co., Inc., USA, 1995.
[15] Eric Freeman, Elisabeth Freeman, Bert Bates, and Kathy Sierra, *Head First Design Patterns*, O'Reilly Publishing Co., Inc., USA, 2008.
[16] Download, http://Download.com, accessed 2015-02-10.
[17] Spyware Guide, http://SpywareGuide.com, accessed 2015-02-10.