Malware Identification Embedded into Malicious Websites Using Client Honeypot Based on Hybrid Detection

Supinder Kaur[1], Harpreet Kaur[2]
[1]Research Scholar, [2]Assistant Professor
[1, 2]Department of Computer Science and Engineering, Sant Baba Bhag Singh Insitute of Engineering and Technology, Punjab Technology University, Jalandhar (Punjab) INDIA
[1]supinder@engineer.com, [1]er.harpreetarora@gmail.com

ABSTRACT:- Attackers are using the malicious websites to launch the cyber attacks against the client's user applications which is one of the growing threat to the internet community. The growing threat to spread the internet attacks through malicious websites led to new technologies to defend against these kind of attacks and client honeypots is one of the these technologies. Client honeypots crawl the internet websites to find and to identify the malicious web servers. The technology called Active H oneypots (Honeyclients) is designed to act as security devices in search of malicious servers that attack client side applications. The main objective of Active Honeypot is to actively browse the cyber space to identify and enlist the malicious websites. This Paper presents the usefulness of client honey pots in terms of collection of malwares which are embedded into malicious websites, the comparative analysis of state based detection of malwares with signature based detection mechanisms, then the design and implementation of the system by incorporating the hybrid model of detection known as signature & state based malware infection detection. Both the detection mechanism is running in parallel mode and most of software tools used in this implementation is open source.

Keywords: **– Network Security, Client H oneypot, SNORT, Honeypots,**

I. INTRODUCTION

A potential malicious website refers to a web page which contains the malicious content which can directly exploit the client side applications to launch the cyber attacks on user's computer. These kinds of attacks occurs on the client side system through web service, therefore they are so called the web-based client side attacks. In the Internet services, the client application normally sends requests and receives responses from servers. With the help of the client honey pot technology, we can actively visit the malicious website to collect all the attack traces which exploit the client side applications such as Mozilla, internet explorer etc. When a vulnerable client application interacts with a malicious server, the malicious server can respond to the client's request with malicious codes to exploit and compromise the client system. In web service, when a user uses a web browser to surf an URL, a malicious server can send to the web browser a web page with malicious code to exploit the client-side system. The result can be an installation of malwares in the client system without the user's consent and disclosure of user's information. The user's computer is often "owned" by attacker and can take part in generating SPAM and Distributed Denial of Service (DDOS) attacks [two stage classification models].

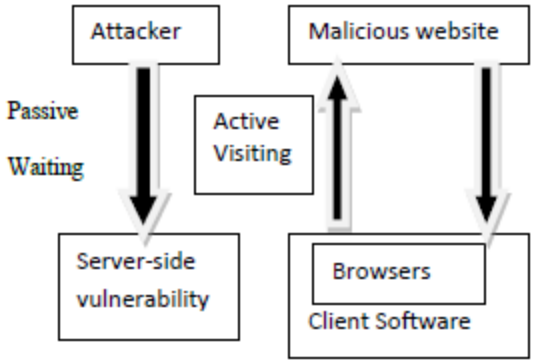


Figure 1: Honeypot versus Client Honeypot

In figure 1, while common server side honeypots expose the server side vulnerabilities and passively await attack from attackers whereas in case of client honeypot, there is actively visitation of the websites which exploit the client side application to launch the attacks. In research, there are a numerous approaches to detect the malicious infection in the websites through exploitation of client side applications. Broadly, we can categorize them into two categories based on analysis features known as static and dynamic based detection. In static based detection approach, the static features are extracted from the content or the properties of the web pages without rendering fully or executing web pages. On the other hand, the later uses run-time features which are captured during rendering fully and executing web pages in particular systems. As a result, static feature based approach is very light-weight but less accurate. In contract, run-time feature based approach has better detection accuracy but consumes significant recourses, time and human labors. In addition, the number of malicious web pages is too small in comparison to the total current number of web pages [1, 2]. It is extremely costly to scan all current web pages on the Internet to capture run-time features in order to identify malicious web pages.

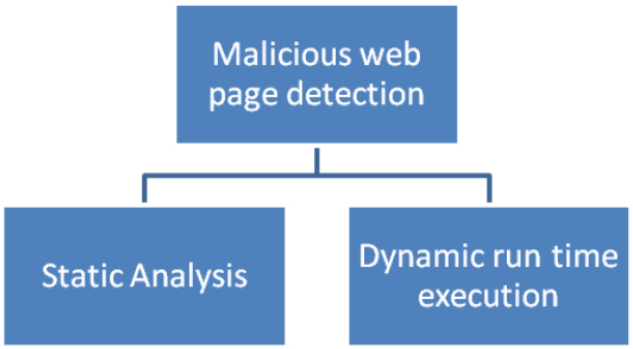


Figure 2. Malicious Page Detection

This paper presents a system to detect the malicious infection in the web-pages through the incorporation of hybrid detection approaches. Two approaches known as signature based and state based detection mechanisms run parallel in the client honeypot machine.

A. Significance of the problem:

As the size of internet is growing exponentially as well as the resources hosted on the internet is becoming major concern which need to be protected from security attacks. Malicious web pages are an emerging security concern on the Internet due to their popularity and their potential serious impacts. Detecting and analyzing them is very costly because of their qualities and complexities. There has been some research approaches carried out in order to detect them. A web application is defined as a network application which is typically interacting with the web browser over the Internet [3]. Information service providers use web applications to deliver their services to users. To do that, they implement their business logic through web applications at a web server with an advertised URL [4]. To enrich their services, the providers can use more than one web server and backend servers and applications which work in cooperation in order to deliver services to the customers. In the client side, there is the main application – web browser which users use to access information services from the providers. In order to expand their functionalities, almost all web browsers support adding third-party plug-in components such as Adobe Acrobat, Adobe Flash, Apple QuickTime, and Microsoft ActiveX.

B. Objectives of the research:

The main contributions of this paper are as follows:

- A review of current researches on detecting infections in malicious web pages through malware downloads.
- Parallel detection model to detect the malicious web pages based on malware drop on user's computer.
- Identification of malicious web pages which can generate the alarm to avoid the visitation of those websites.
- Comparative analysis of signature based and state based detection approached to detect malicious web pages.

The remaining paper is in various Sections. In Section II, defines and explains the technology that has been employed and discusses the client honeypots in brief and other detection approaches. Section III deliberates the framework design and discusses the detailed design of the implemented system. Section IV discusses conclusion and future work of the research problem.

II. BACKGROUND AND RELATED WORK

A. Client-side attacks

As the number of Internet users has increased significantly, web-based attacks that use malicious web pages to exploit users' system have become a primary concern in the Internet security. A web-based client-side attack happens when an Internet user visits malicious web pages which attempt to exploit the user's browser vulnerabilities, plug-in application vulnerabilities or user's operating system vulnerabilities in order to compromise the user's system [5]. To deliver malicious content to the client-side, an adversary first needs to publish malicious contents on the Internet. Compromising a web server is one of the common ways to deliver malicious contents. Various methods are reported to be used to increase attack effectiveness [6, 7, 8, 9, 10, 11]. Intruders can compromise a website by exploiting some vulnerabilities in the web server, exploiting a vulnerable web application [9], vulnerable database applications such as SQL injection [12, 8, 13]. The results from this compromising are inserting malicious contents which can be delivered to the client-side system [12, 13]. After publishing their malicious contents on the Web, attackers must get users to visit the malicious web pages in order to make exploitation [12]. Spam is a common technique which intruders use to lure user to their malicious web pages. For instance, spam emails can contain a links to a malicious web page. Web blogs and social networking sites are also abused to get users to visit malicious sites [14].

B. Related Work

This Section reviews some current analysis methods which are used to detect malicious web pages. They are classified into three main approaches as shown in the following figure 3.

Figure 3: Malicious web page detection approaches

a) Signature Based Technique

In the signature approach, detection systems use known signature to detect malicious web pages. Signatures can be from some well-known Intrusion Detection Systems (IDS) or anti-virus applications. This approach is commonly used in the detecting system using low interaction client honeypot. Seifert, Welch and Komisarczuk [15] used Snort signature to detect malicious web pages in their HoneyC system. The HTTP

(Hyper Text Transfer Protocol) responses from web servers are formatted using a standard XML (ExtensibleMarkup Language), and then analyzed against Snort signatures. In Monkey-Spider system, Ikinci, Holz and Freiling also used signature approach to detect malicious websites. The contents of websites are crawled and stored in files. The crawled contents are then scanned by ClamAV –an anti-virus application [15]. Unfortunately, this technique can miss some types of attacks due to complexities of malicious web pages and their obfuscations. Moreover, unknown attacks are not covered by signatures so they are missed.

b) State Based Detection Technique

In addition, state-change approach is commonly used in the detecting systems using high interaction client honeypot – one of the efficient instruments to detect malicious web pages. The main idea of this approach is monitoring the state change in the client system during visiting an URL time. If there is any unauthorized state change during visitation, the visit URL is classified as malicious. In the Strider Honey Monkeys system, a monkey program loads a browser, instruct it to visit each URL and wait for a few minutes for downloading process. The state changes in the system is then detected against unauthorized creating executable files or registry entries in the system [16]. Moreover, to detect drive-by-download attack, Moshchuk, Bragin, Gribble and Levy used event triggers. They created some trigger conditions to track unauthorized activities in process creation, file system and registry system. The trigger conditions also include any event that makes browser or the system crash. During visitation, if an URL make a trigger fire, it is classified as unsafe [17]. The state change approach is also used by Xiaoyan, Yang, Jie, Yuefei and Shengli, in their client honeypot system to collect Internet-based malware. A behaviour monitoring module is conducted to track malicious behaviour. It hooks native API, DLL (Dynamic Link Library) functions and TDI (Transport Driver Interface) in order to monitor all activities causing buffer overflow, accessing system resources such as process, network, file, and registry [18]. However, state-change technique has limitation. It is difficult to create a completely rule which includes all of vulnerabilities of combination of operating systems, browsers and their plug-ins. Thus, missing attacks is a known limitation of this technique.

c) Machine Learning Approaches

Seifert, Welch and Komisarczuk [19] proposed a novel classification mechanism to detect malicious web pages. This method is based on HTTP responses from potential malicious web servers which are then analyzed to extract potential malicious characteristics. The method was used in a hybrid system in which all URLs are classified by static heuristic method and sent to high interaction client honeypot for verification. To classifying URLs by static heuristics method, some common attributes are chosen based on three proposed main elements in malicious web pages: exploit, exploit delivery mechanism and obfuscation. J4.8 decision tree implementation from Weka was used. This classifier had very good false positive rate (5.88%) but very high false negative rate (46.15%). Hou, Chang, Chen, Laih and Chen proposed a machine learning approach to detect malicious web content [14]. The key point in this research is the method used to choose features according to the usages of DHML (Dynamic Hypertext Markup Language) knowledge. Three groups with 171 features were chosen. There are 154 features used to count the use of native Java functions, 9 features measuring some elements inside a HTML document, 8 advanced features counting the use of ActiveX object. In order to study about choosing type of features, the authors took some experiments with different chosen features. Decision tree algorithm is used in these experiments. While using all features cannot get high true positive and low false positive result, the combination of three features can get very good result. The authors also compared the results of different classification algorithms: decision tree, Naïve Bayes, SVM and boosted decision tree. To detect malicious web pages, Bin, Jianjun, Fang, Dawei, Daxiang and Zhaohui [4] proposed the concept of abnormal visibilities. According to their studies, malicious web pages are usually changed in their display modes in order to be invisible or almost invisible. The authors showed three main forms of abnormal visibility: width and height attributes of iframe, setting the display style of iframe ”display: none’, generating iframe tag dynamically to make obfuscation. Abnormal visibility fingerprints are created and used to detect malicious web pages. Each web page is scanned to detect any form of abnormal visibility. The detected value in any kind of abnormal visibility is compared with a threshold value. To carry out the experiment, the authors detect 60 websites reported malicious by StopBadWare.org. They scanned 66882 pages from these websites and found 30561 malicious one. They also figured out that their system has low false positive (1.99%) and false negative rates (2.63%). Ma, Saul, Savage and Voelker [21] pinpointed a new approach to detect malicious web pages named lightweight URL classification. In this approach, they classify web pages based on relationship between U RLs, their lexical and host based features. It does not use contents of web pages in detection. Naive Bayes, SVM and Logistic Regression are used for classification. The authors used two experiments in their study. The first experiment is for comparing between feature sets by using _1-regularized logistic regression (LR) classifiers. The results showed that using more features got better classification accuracy. In addition, their experiment [22] was conducted to build online learning algorithm to detect malicious web pages. They used the same feature as the experiment [21]. There were three online algorithms implemented: Perception, Logistic Regression with Stochastic Gradient Descent and Confidence-Weight. They compared their online learning algorithm with Support Vector Machine (SVM). The results showed that SVM needed more training data set in order to get better accuracy but their algorithms did not. Chia-Mei, Wan-Yi and Hsiao-Chung [23] proposed a model to detect malicious web pages based on unusual behaviour features such as encoding, sensitive key word splitting and encoding and some dangerous JavaScript functions. To classify web pages, they created a scoring mechanism which cored based on 9 predictor variable. Moreover, weights for each predictor variable were decided by training phrase. The results

from their experiment showed that their model worked very well. However, their dataset was very small with 460 benign and 513 malicious web pages. Shih-Fen, Yung-Tsung, Chia-Mei, Bingchiang and Chi-Sung [22] proposed a novel semantics-aware reasoning detection algorithm to detect malicious web pages (SeAR) which was based on structures of HTML codes. Firstly, they defined templates for HTML codes. For each tested HTML code, the distance between the tested HTM L code and templates were calculated. Secondly, the best match was chosen based on the distance and weight of the template. Finally, threshold was used to make decision whether web pages were classified as malicious or benign. The outcome from this research is very good but their dataset had only 147 malicious instances (no benign one).Almost all of current machine learning approaches use static features which are extracted from contents or properties of web pages. However, it is uncertain to use static features to distinguish malicious web pages from benign ones. There are two reasons for this issue. Firstly, obfuscation makes static features unable to be used to distinguish between malicious and benign web pages efficiently. Secondly, legitimate website can deliver malicious contents to client-side system if it is compromised. In this case, static features extracted from properties of website are not valuable to be used to detect malicious web pages. In fact, compromising legitimate websites is one of the effective methods to spread malicious contents.

III. SYSTEM DESIGN

This section describes the proposed system design which incorporates the client honeypot technology with state based and signature based detection capabilities in parallel. The main significance of this research is put together the existing tools and techniques for detection of malicious infection based on malware drop on a user's computer without his knowledge and concern which is an indication of malicious websites. The system has been tested through various malicious websites which have malicious content embedded into them and got the good results in terms of malware collections. Those collected malwares can further be studied with the help of various analysis techniques.

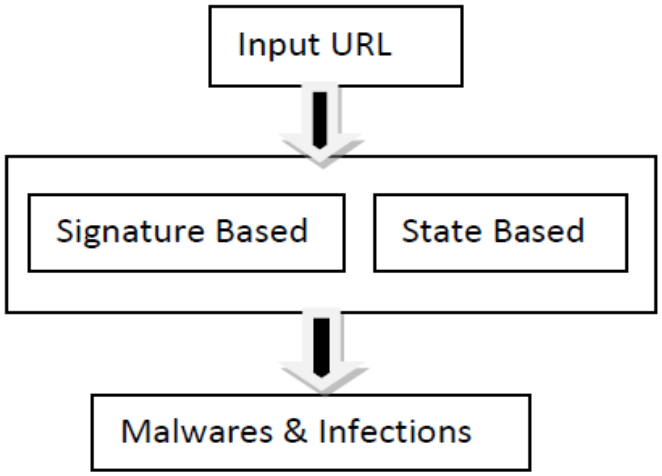
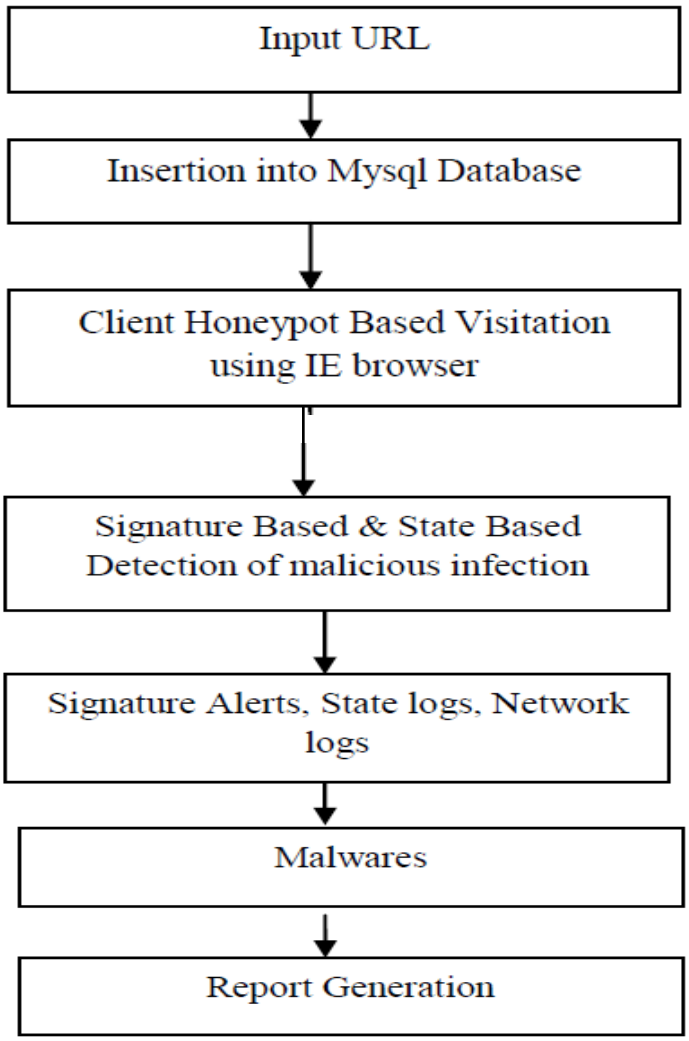


Figure 4: System Design of Malware Infection Detection

A. Operation Flow:



B. Operation The operation flow of the system can be described as below:

a) A list of URL which needs to be inspected is sent to the client honeypot virtual machine.
b) The URLs are being submitted to signature based engine as well as state based engine
c) The URL is declared malicious based on the malicious drop on the victim machine as well as based on correlation between signature alerts and state based detection.
d) If URL is not classified by signature based but there is malware drop in state based, it can act as alarm for signature generation.

C. Tools and Techniques Used

Hardware/Software Specifications		
Feature	**Product**	**Specs**
Host Operating system	Red Hat Linux	**Normal Production Machine** **RAM : More than 1GB**
Guest Operating System	Window XP	Single processor Virtual Machine **RAM** 256MB **NIC** 100Mbps host-only vmnet
Virtualization Software	VirtualBox	Virtualbox3.0.2 for Linux
Packet Capturing	TCPDUMP	Tcpdump tool
IDS	SNORT	Open source IDS
Browser	IE6	Visitation of URL

Table 1: Hardware & software used in implementation

D. Experimental Results

Here we discuss the testing of the working system with real world malicious URLs. When we submit the URL or list of URL to the system, there are both the detection mechanism known as signature based and state based detection are running in parallel. Network packet capturing is being performed to monitor the network traffic originated from the client guest machine during the active visitation of URL We can execute a single URL as well as multiple URLs at the same timeframe.

Table 2 indicates the execution results of few malicious URLs in the system. As shown first column depict the URL name, 2nd column indicate the IDS signature alerts generated and column 3 indicate the MD5 value of binary collected execution.

URL	IDS ALERTS	Md5 of Collected Binary
http://xyz.in/	1) (http_inspect) INVALID CONTENT-LENGTH OR CHUNK SIZE 2) (http_inspect) NO CONTENT-LENGTH OR TRANSFER-ENCODING IN HTTP RESPONSE 3) SENSITIVE-DATA Email Addresses 4) SDF_COMBO_ALERT 5) WEB-CLIENT PCRE character class double free overflow attempt	c6f71 f4b7a6 6e566636f 1f 2dea805152
http://abc.in/download/3fa90a3c-93b2/autobotWOT.exe	1) SENSITIVE-Email Addresses 2) SDF_COMBO_ALERT	07669038a3 e26084fc239 2eeb0e917c2
http://Abc/file54769	1) SENSITIVE-Email Addresses 2) SDF_COMBO_ALERT	6406ceabcd0 6f58dbf112c 6a8fe69596
http://aBC-/file69702	1) SENSITIVE-DATA Email Addresses 2) SDF_COMBO_ALERT	58779e31a5 35d446e01fb 0cb5ca34e39
http://abC.in/file74703	1) SENSITIVE-Email Addresses 2) SDF_COMBO_ALERT	492233d6ae b7852443eee 3ed6ca1e810

Table 2: Execution result of malicious URL

Binary	URL	Hostname	MD5
3306.exe	http://199.x.x.x/	199.x.x.x	184f2b285a8cb956c6ecf62c314c539a
user/soapCaller.bs	http://1.1.1.x	203.x.x.x	5343c1a8b203c162a3bf3870d9f50fd4

Table 3: Binary samples collected

Table 3 depicts the binary samples collected from some during malicious URLs which indicate the malicious infection in those websites. Below figure depict the port-wise distribution of attack events by processing the IDS alerts file. We can see that TCP protocol and ICM P protocol is widely used for attacks.

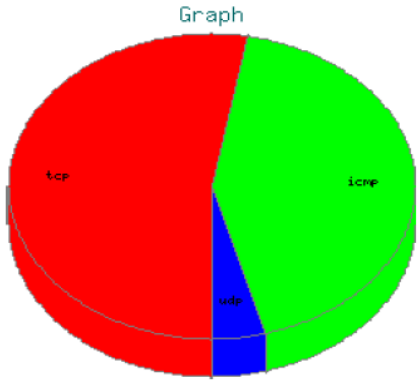


Figure 5: Port-wise Distribution of Attack events

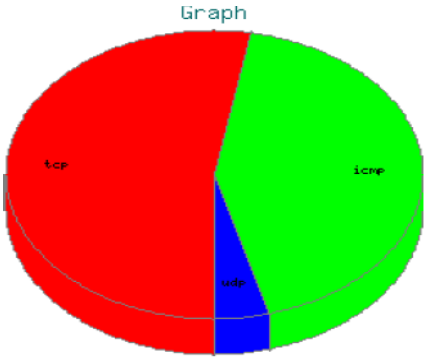


Figure 6: Distribution of attack by Destination Port

IV. CONCLUSION

In this paper, we present the result of the existing solution of client honeypots for detection of malicious websites and that most of the solution is available for public users and closely bound, thereby we solve the problem of system which is able to detect the malware programs with the help of client honeypot as well as by applying the intelligent forensic investigation of the collected network PCAP data.

ACKNOW LEDGEM ENTS

I would like to sincerely thank Mrs. Harpreet Kaur for her contribution and help in writing this paper.

REFERENCES

[1] C. Seifert, R. Steenson, T. Holz, B. Yuan and M. A. Davis, Know Your Enemy: Malicious Web Servers, The Honeynet Project (2007).
[2] Y.-M. Wang, D. Beck, X. Jiang and R. Roussev, Automated WebPatrol with Strider HoneyMonkeys: Finding Web Sites that Exploit Browser Vulnerabilities, IN NDSS (2006).
[3] J. Mehdi, Some Trends in Web Application Development, 2007 Futureof Software Engineering, IEEE Computer Society, 2007.
[4] D. Gollmann, Securing Web applications, Information Security Technical Report, 13 (2008), pp. 1-9.
[5] Van Lam Le, Two-Stage Classification Model to Detect Malicious Web Pages, 2011 International Conference on Advanced Information Networking and Applications.
[6] Websense, State of Internet Security, Q1-Q2, Websense Security Labs, 2008.
[7] Sophos, Security threat report: 2009, Sophos, 2009.
[8] ScanSafe, Annual Global Threat Report, Trends for January 2008 -December 2008, 2009.
[9] Symantic, Security Threat Report - Trend for 2008, Volume XIV, April 2009.
[10] ScienceDirect, Most malicious web sites are hacked, Network Security, 2008 (2008), pp. 1-2.
[11] Websense, State of Internet Security, Q3-Q4, Websense Security Labs, 2009.
[12] P. Niels, R. Moheeb Abu and M. Panayiotis, Cybercrime 2.0: Whenthe Cloud Turns Dark, Queue, 7 (2009), pp. 46-47.
[13] Microsoft, Microsoft Security Intelligence Report, January through June 2009, 2009.
[14] B. Garrett, H. Travis, I. Micheal, P. Atul and B. Kevin, Social networks and context-aware spam, Proceedings of the ACM 2008conference on Computer supported cooperative work, ACM, SanDiego, CA, USA, 2008.
[15] C. Seifert, I. Welch and P. Komisarczuk, HoneyC - The Low-Interaction Client Honeypot, NZCSRSC, Hamilton, 2007.
[16] Y.-M. Wang, D. Beck, X. Jiang and R. Roussev, Automated WebPatrol with Strider HoneyMonkeys: Finding Web Sites that Exploit Browser Vulnerabilities, IN NDSS (2006).
[17] E. Moshchuk, T. Bragin, S. D. Gribble and H. M. Levy, A crawler based study of spyware on the Web, (2006).
[18] S. Xiaoyan, W. Yang, R. Jie, Z. Yuefei and L. Shengli, Collecting Internet Malware Based on Client-side Honeypot, Young ComputerScientists, 2008. ICYCS 2008. The 9th International Conference for, 2008, pp. 1493-1498.