RESEARCH ARTICLE OPEN ACCESS

Client Honeypot Based Drive by Download Exploit Detection and their Categorization

Mandeep Sidhu*, Rajneesh Narula**
*M.Tech Scholar (Computer Science and Engineering, Adesh Institute of engineering and technology faridkot – Punjab Technology University)
** Assistant Professor (Computer Science and Engineering, Sant Adesh Institute of engineering and technology faridkot –Punjab Technology University)

ABSTRACT
Client side attacks are those which exploits the vulnerabilities in client side applications such as browsers, plug-ins etc. The remote attackers execute the malicious code in end user's system without his knowledge. Here in this research, we propose to detect and measure the drive by download class of malware which infect the end user's system through HTTP based propagation mechanism. The purpose of this research is to introduce a class of technology known as client honeypot through which we execute the domains in a virtual machine in more optimized manner. Those virtual machines are the controlled environment for the execution of those URLs. During the execution of the websites, the PE files dropped into the system are logged and further analyzed for categorization of malware. Further the critical analysis has been performed by applying some reverse engineering techniques to categories the class of malware and source of infections performed by the malware.
Keywords –Client Honeypot, Drive-by-Download, Client-side exploits, Intrusion Detection, Malware Analysis

I. INTRODUCTION

If we see the current internet usage statistics, the users who are using the internet with the help of laptop, PC and mobile phones, do not know about the security related aspects such as how to protect the computer from malicious software's which are spreading over internet. Over the past few years, as the number of computers are connected on internet, the computer security is gaining popularity. The growing number of computers connected to the internet, new benefits as well as the new threats has been arisen. The number of attacks are increasing on a daily basis from everywhere in the world. Prevention tools like firewalls cannot protect networks on their own anymore. Attackers get smarter and they are able to overcome such prevention systems. Although in the current internet world, the number of universities, organizations owned by government is providing the workshops and tutorials to their students to get awareness about the network security issues and to defend against the computer attacks.

In the field of network security, there are two kinds of communities to be deal with- black hats and white hats. It becomes quite interesting when the white hat community are not only keen to defend the networks but are keen to gather the attack logs to make the fool of black hats.

The work in this paper is based on implementation of class of honeypot so called client side honeypots which is able to actively browse the malicious weblinks and able to get exploited by the infections in those weblinks. Once the exploits have been performed, the activities of the attackers are being logged which are further studied to get the inference and detailed behavior of the attackers. The complete process and methodology of the thesis implementation can be summarized into below steps:

- Actively browse the weblinks in controlled environment
- Responding the attacker to capture the in depth infection
- Collection of attack data in the form of network traffic and PE executable files.

In today's modern life, computer and internet is becoming the important part of everyone life and more and more volume of peoples are connected through the internet world. The term internet is basically a World Wide Web (WWW) which is becoming a useful resource of information for everyone's life with the current development of internet. With many of positive effects of the internet in our current modern life, there are also some sort of negative effects in the form of network security as more and more kind of malwares are spreading over the internet which can affect the innocent users connected over the internet. A number of malware such as virus, Trojan horse exist in the Internet. The malware has the characters of profit [1]. It behaves in these ways: stealing personal information, user names and passwords.

1. **Intrusion Detection System**

An Intrusion detection system (IDS) is a security system that monitors computer systems and network traffic and analyses that traffic for possible hostile attacks originating from outside the organization and also for system misuse or attacks originating from inside organization.

1.1. Classifications of Intrusion Detection System

- Host Based IDS
- Network Based IDS

Host Based IDS

Host based IDS consists of software or AGENT components, which exist on Server, Router, Switch or Network appliance. The agent versions must report to a console or can be run together on the same Host. This is NOT the preferred method though.

Network Based IDS

Network based IDS captures network traffic packets (TCP, UDP, IPX/SPX, etc.) and analyses the content against a set of rules or signatures to determine if a possible event took place. False positives are common when an IDS system is not configured or "tuned" to the environment traffic it is trying to analyse. Network Node is merely an extended model of the networked IDS systems adding aggregated and dedicated IDS servers on each NODE of a network in order to capture all the networked traffic not visible to other IDS servers.

Techniques based classification: Intrusions can be detected by various techniques. Most important of those are:

Anomaly detection

In the case of anomaly detection, the anomaly detector constructs the profiles of normal system behaviour and then uses this behaviour to detect any kind of abnormal activities in the network. To determine the attack traffic, the system must be learned to identify the normal behaviour. This kind of learning the system can be performed in many ways such using the subject knowledge of AI techniques. Another method can be of using the mathematical models. It can also be known as strict anomaly detection method.

Signature detection

Figure 1.2 depicts the working of the signature based detection approach in detection of intrusions in the network. Normally these kinds of solutions are used in real time protection of the network.

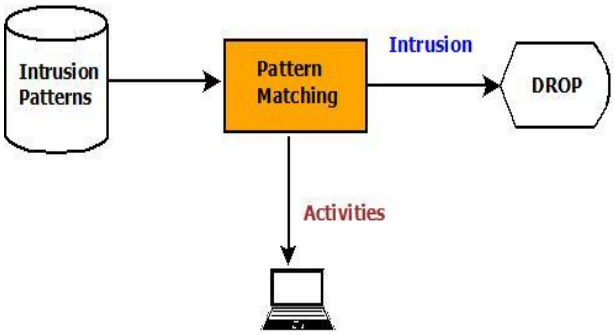

Figure 1.1 Diagram of Misuse Detection

1.2. Honeypots

1.2.1. What is Honeypots:

A Network security resource whose value is being probed, scanned, attacked, compromised, controlled and misused by an attacker to achieve his objective or we can say the black hat community target the network resources and honeypot is a resource to be attacked by the attackers so that the activities of the attackers is being logged by the honeypot which are further analysed to study the behavior of the attackers. Lance Spitzner defines Honeypots as "A Honeypot is an information system resource whose value lies in unauthorized or illicit use of that resource" [2].

A honeypot is a computer which has been configured to some extent to seem normal to an attacker, but actually logs and observes what the attacker does. Thanks to these modifications, accurate information about various types of attacks can be recorded. The term honeypot was first presented by Lance Spitzner in 1999 in a paper titled To Build a Honeypot [3].

In a general way, a honeypot is basically a computing resource, whose main purpose is to be attacked, probed, and compromised by the attacker in any form of unauthorized way. As we said the computing resource which could be of any types: a service to be exploited by the attackers, a application, a system or set of systems, or it can be just a piece of information or data. The basic concept here is that any kind of interaction with those resources in the form of honeypots is treated as suspicious by definition. All the interaction occurred between any entity and honeypots is monitored and logged which are further analysed in details to get the internal behavior about the adversaries.

Classifications of Honeypot:- The honeypot can be categorized by the two basic fundamental classes which is "types of attacked resources" and "level of interaction" [4] . The first class- types of attacked resources- depicts the types of resources of the honeypots, whether the honeypot's resources which are being exploited are server side resources or client side resources. In case of server side honeypots, the

network services such as SSH or NetBIOS, are being listened on their standard ports and any connections initiated by remote client is monitored for any kind of suspiciousness whereas in case of client side honeypots, a client side application such web browser which connect to the remote server, are being monitored.

Next classification is based on the level of interaction- low interaction, high interaction and hybrid honeypots. Here level of interaction we mean the kind of environment provided by the honeypots to the remote attackers.

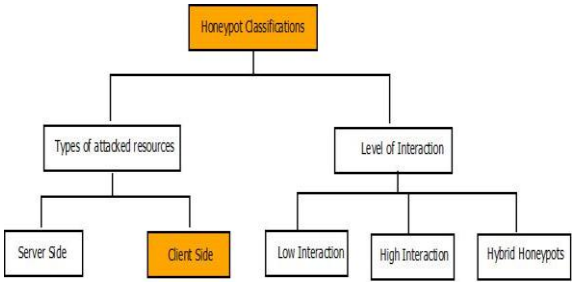

Figure 1.2 Honeypots classifications

As shown in the figure 2, the server side and client side honeypots are based on the criteria of the services or application exploited are either on server side or client side whereas the level of interaction determine whether the honeypots are real or emulated. Real honeypots provide the real environment in the form of real services and real Operating system to the attackers so that these real services will get exploited whereas in case of emulated kind of honeypots, there is no real environment, only fake services and fake applications will be visible to the attackers. The real honeypots are called high interaction honeypot and the emulated honeypots are low interaction honeypots. A hybrid honeypots is basically combines the functionalities of both categories of honeypots which provide both real and emulated honeypots.

Server Side Honeypots: This category of Honeypots able to detect and study the attacks occurred on network services. They expose the opened ports and services or the whole applications to the attackers and they listen passively for any incoming connections, established by the remote user or attackers. Those remote users are likely malicious in nature as the sole purpose of honeypots is to collect the attacks data, hence there is not any kind of production values associated with the honeypots. Server side honeypots are also known as passive class of honeypots as they wait passively for attacks.

Client Side Honeypots: These categories of honeypots are able to detect the attacks on client side applications such as browsers, plug-ins. These client applications are software's which make the connections to the remote server. The client honeypots also known as honeyclient, actively browse the remote servers and collect the attacks data associated with those remote server, thereby these are also known as active honeypots.

If we see the operations and working capabilities of the server and client honeypots, the client honeypots are very different than server or passive honeypots. Client honeypots actively make the connection to the remote server in order to detect any kind of malicious behavior of the remote server. The most popular honeyclients are those detecting attacks on web browsers and their plug-ins, propagated via web pages. Some also have the capability to look at various forms of attachments, and they have been attempts to create instant message honeypots as well.

1.1 Drive by Download:-

The Drive-by download is class of malware which download on a user's computer without his concern or without his knowledge. In the field of computer security, drive by download infection is usually triggered when the user visits the potential malicious weblink and the attack has occurred by exploiting the browser's vulnerability. The binary file which is downloaded on a user's computer is usually a program which is malicious in nature and which installs itself on the user's computer. In this research thesis, we discuss the problem of drive-by download class of malwares and possible detection mechanism of them through high throughput client honeypot.

Here in this thesis, we also discuss the difficulties of detection of drive-by download malware samples and propose the solution which could solve the purpose of drive by download kind of malware collection. If we look at the historical cases of drive by download infection, many cases of infections have been observed in the first half of 2008 [5]. In each infection of these attacks, the websites have been used in order to distribute and spread the malware, and many numbers of computers have been infected when the users are simply browsing the internet with the help of normal browser applications. Infection of Drive-by download class of malware is very complex since there is no single and concrete mechanism to detect these kinds of attacks.

In order to infect the user's computer, the attacker's turns to target the upper layers of the OSI stack to drop the malwares such as the exposed web services are increasing the targeted medium by the attackers to propagate the attacks [6, 7]. As stated in the above paragraphs, the attacks know as "Drive-by-Download" target the web applications to make the exploits and to drop the malwares into user's computer. They make the attacks by exploiting the browser vulnerability for insertion of the malicious program into the user's machine without the user's

consent or notice, when the client interacts with the server to retrieve an infected web page.

The infection process by actively browsing the remote server by the users is depicted in the figure 3 and figure 4. On the user's computer, a browser initiate the request to the remote web server (step-1) and the in response the server dropped the malicious code into the user's computer (step-2) by exploiting the browser's vulnerability. The vulnerabilities on the client side are targeted by the Drive-by-Download class of attacks by sending the exploits in response to the request generated by the client's machine. Within the client side browser's applications, there is a vulnerability associated with the browser itself. There can also be other installed software programs on the client machine which can be targeted by the drive by download class of malwares such as shared libraries and plug-ins, active X components etc.

1.4.1 Phases of Drive-by-Download Attacks [8]: The infection and attack propagation mechanism of Drive-by-Download include the various steps. Firstly the attacker makes the legitimate websites as malicious by injecting some kind malicious code into it. When the normal user browses the weblink through the application browser, it injects this malicious program into user's machine by exploiting the browser side vulnerabilities which is also known as client side vulnerabilities.

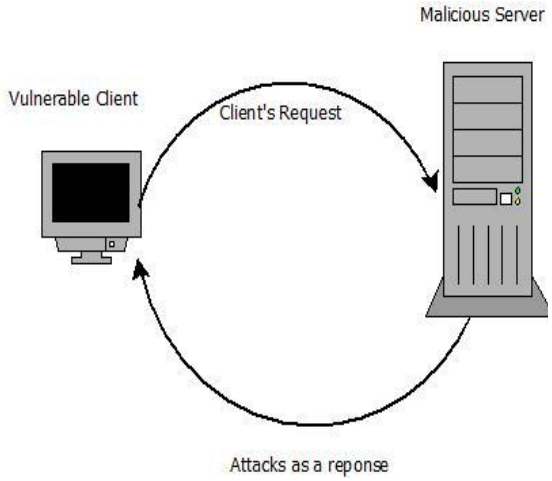

Figure 1.3 Step-1 of Client side attack

The attack cycle incorporating the various steps:-

1) The legitimate web servers are targeted by the attackers to insert the scripts in the web applications. Those inserted scripts are the malicious which will execute when any user will browse the web application.

2) The user visits the compromised weblink which become victim of the attacker.

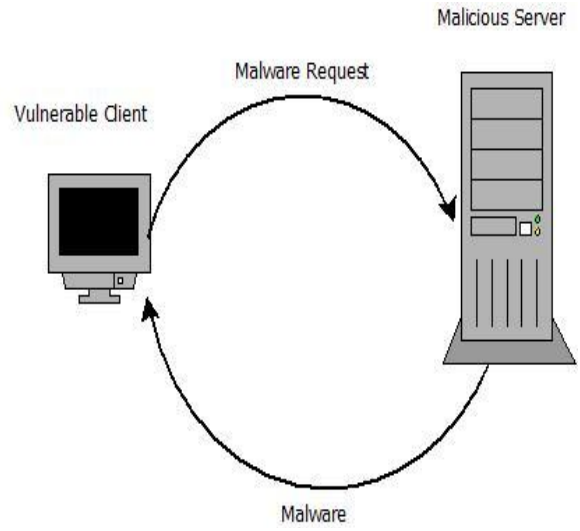

Figure 1.4 Step-2 of Client side attack

3) The web server on which the user have been visited, send the response along with the malicious script which he had inserted into the weblink. The malicious injected script is either in the form of exploits or it can be a script that imports another code form central server.

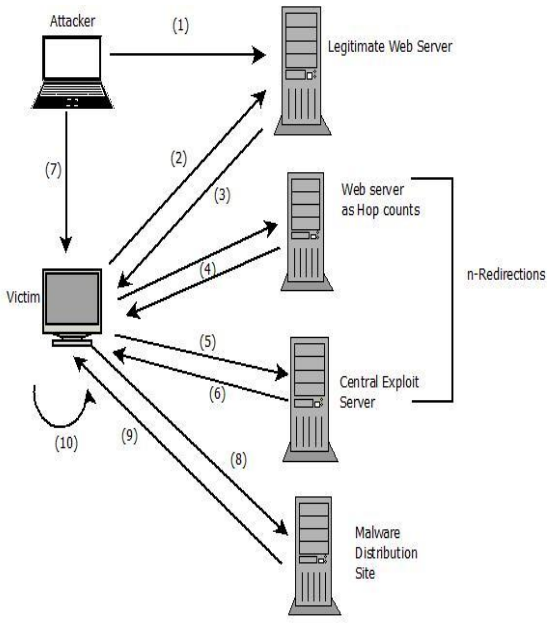

Figure 1.5 Drive-by-Download Infection steps

4) Hop count is the redirections made from one web server to another which also plays the source of infection.
5) The central exploit server has been reached after multiple redirections so called hop count.
6) The exploit scripts have been loaded into the user's machine by the central server.
7) By exploiting the browser's vulnerabilities, the remote attackers take the full control of the victim's system.

8) The exploits which is occurred on the victim machine take the browser to visit the malware distribution sites. This is actually the phase of starting the drive-by-download attacks.
9) The PE executables files are downloaded.
10) The victim or user's machine automatically executes the malicious code after the malware download.

II. IMPLEMENTATIONS

2.1 Methodology:- The purposed work completes through the following steps in order to analyze the network attacks.

- System development and creation of environment for logging the file system activities.
- Virtual machine development for browsing of websites
- Network settings and creation of control network.
- Establish and configure the guest client honeypot machine.
- Website submission for visitation.
- Design of data base tables.
- Development and integration of monitoring modules
- Integration of signature based detection tools
- Development of automatic report generation code
- Analysis and report generations

2.2 System Design:

Here we discuss and present the complete system design of the implemented system. As shown in the below architecture and system design, the major modules of the system are presented.

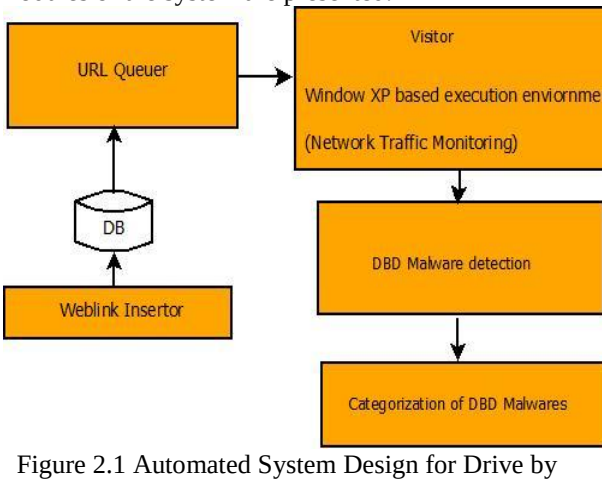

Figure 2.1 Automated System Design for Drive by Download Malware Collection

Modules of the System:

1. Weblink inserter: - This module inserts the URL or list of URL into database from where they can be taken for further analysis and browsing purpose. The code is written in perl language which picks the URL one by one from the pool and inserts them into Mysql database.
2. Database: - For the purpose of logging the URLs and their analytical results, the MySQL database is designed. The historical logs of the collected data set can be seen by applying the queries to the database.
3. Queuer: - This is third module in the implemented system, the main purpose of it basically making the URLs into queue for the execution.
4. Window XP based Visitor: - This is real environment in which we actually browse the web link. The web links are visited in a sequential or random manner. When there is multiple list of URLs to be visited into a database, they are pulled one by one and browse. At the same point of time, a single web link or multiple weblinks can be visited. The key features of the module are:
   a. Virtual Machines based execution
   b. Browsers and plug-ins
   c. Applications
   d. Network Traffic Monitoring
5. Malware Categorization: - In the malware categorization, the collected malware samples which are dropped into user's machine are submitted to the popular anti-virus scanner through a popular web service www.virustotal.com [9]. The result of this module is labeled malwares class such as Trojan downloader, worm etc.

2.2.1 Major Components of the system

- **Visitor (Controlled environment for URL browsing):-** This is a controlled environment for the execution of malicious URLs by using the Virtual machines. Here we use the technology so called virtual machines through which we can use the resources of the single base machine and which helps in creation of multiple operating systems. This basically reduces the cost of the hardware requirement because we can create multiple resources on a single base machine. Here we had created the Window XP service pack 2 virtual machine for the execution of the malicious URLs.

- **Window XP based Visitor**

Here in the module, the malicious domains are executed through the window XP machine using the internet explorer. Following major monitoring modules are being used to monitor the network level activities.

- Network Monitoring from windows virtual machine to the internet
- Standard Data capturing tools – TCPDUMP

- Network Traffic Monitoring and logging of data.
- Network dumps in the form of PCAP data are being generated corresponding to each URL which is later being processed with data processing engine to extract the malwares dropped on a victim machine.

2.3 Algorithm and Flowchart

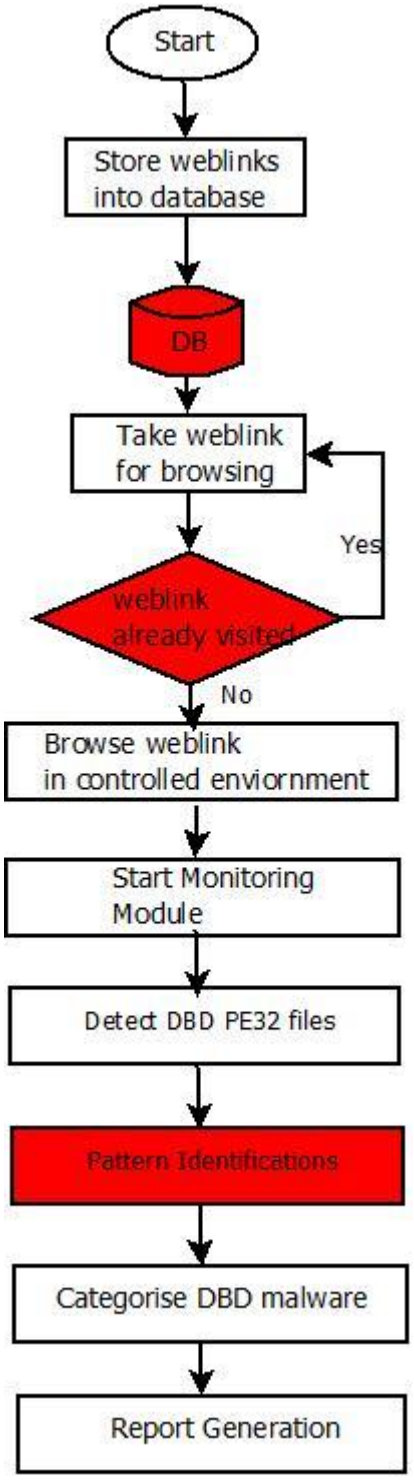

Figure 2.2 Flowchart

2.4 Report Generation and Data Processing Engine

In this module, we process the network logs collected during the active browsing of the URLs to generate the report of attacks like attack distributions, port-wise distributions, bar charts, pie-charts of attacks etc. The processed data is also saved into MySQL database for further future purposes. The flow diagram of the report generation module is depicted in the figure 2.3. The network logs collected are processed through IDS engine to generate the log file of attack data. Then these alert files are again processed through report generation engine which uses the open source libraries to produce the graphical reports of the attacks.

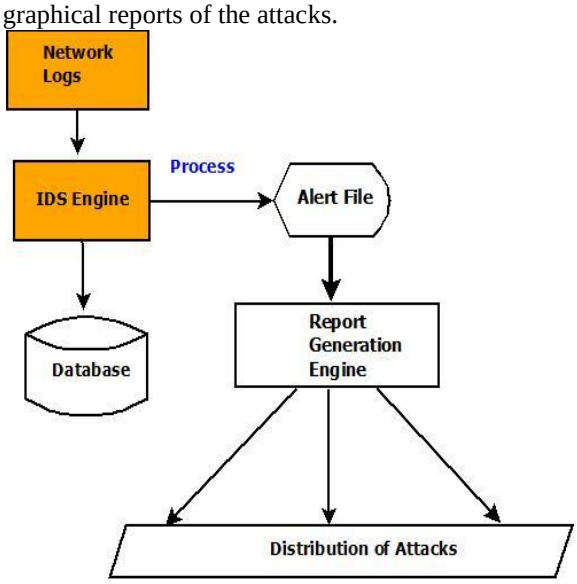

Figure 2.3 Report generation engine

Some collected DBD Malware samples:

weblink	DBD malware MD5	VT detection
http://x.x.x./yy-movie0085.html	c54afad405cd87fc1b6d2dcc22d1d53a	44/46
http://xxx/yy.html	c54afad405cd87fc1b6d2dcc22d1d53a	44/46
http://xxx/yy1.html	c54afad405cd87fc1b6d2dcc22d1d53a	44/46

Table 1: URL and MD5 malware collections

Report Generation and Distribution of Attacks:
Figure 4.15 represents the distribution of attacks after the processing of network logs during the actively browsing of URLs in our system. The hourly distribution of attacks is depicted in the below figure.

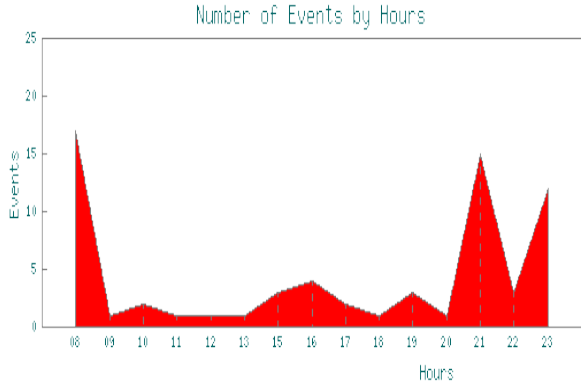

Figure 2.4: Hourly distribution of attacks.

III. Conclusion

In this paper, we present the developed system is able to collect the infections known as drive by download malwares which are dropped on the user's machine without his knowledge or concern.

With the implemented research work we have tried to present our efforts to defend the regular rising of client side attacks on the internet. At the same time, we also observed the affect of drive by download malware which may affect the normal user's machine and can steal important information from his computer such as credit cards information, passwords etc. First of all, we showed the significance and need to defend against the client side malwares which are spreading on the internet and they are increasing in exponential manner. These client side malware exploit the client side vulnerabilities such as browsers, installed client side applications, plug-ins etc. Thereafter, we introduced the reader to various classes of malware to get the internals about the malicious softwares. Then we introduced the concept of honeypot technology and importance of honeypots in terms of network security. The most of network security devices such as firewalls, IDS etc are working the principle of signature based detection which is so called reactive model of network security, in compare to those honeypots work on the principle of proactive kind of network security.

As the next step, we presented some existing solutions in the form of client honeypots which are useful for detection of malicious websites and malwares spreading by actively browsing these potential websites. Then based in the drawback in current solutions of the client honeypot, we introduced our solution - the Drive by download malware collection through client honeypots - and explained its implementation and modules of working. We presented the detailed system design which is mixture of network traffic analysis and some scope of current existing solutions. In our system design, one block is for actively browsing the weblinks and monitoring the complete network traffic. Another block is extraction of PE executables from the collected network traffic and then last block is categorization of malwares through virus scanner with virustotal.com.

In the final, we tested the execution of potential malicious weblinks and seen whether our system is able to collect the malwares. We learned some good lessons and also found some interesting results during the malware collection. We also cross checked the system by applying the manual analysis of network with deep packet inspection and we found that our system is able to collect the drive by download malware samples.

In the end, we conclude here that researchers need to concentrate more on client side attacks to protect the end users. There is increase rise in botnet attacks which use the malicious weblinks to propagate. In our current research work, the functionality of crawler is missing which we would like to propose by integrating any generic crawler. Also our execution environment is limited in the form of window XP service pack 2, whereas the functionality of the most advance malwares majorly relying on the kind of executions environment they will get. If they do not get the correct environment, the malware do not behave accordingly. Therefore, we also would like to propose more profiles for execution of potential weblinks which is a resources intensive and computing problem. In the next, we also propose that any researchers may apply the automated signature generation mechanism through inspection and exploring the semantic behavior of the collected malware samples which is also lacking in the current scope of research. Once all the research capabilities are added in the current research, then it might greatly lead to the maturity and significant improvement to defend the client side attacks.

ACKNOWLEDGEMENTS

I would like to acknowledge Assistant Professor, Rajneesh Narula for her support and guidance in writing this research paper

References

[1] Ren Liu. China virus status & Internet Security Report in 2006.2007-02-01.http://www.donews.com/Content/200702/eda7daf7970448608b2881d97c9a1868.shtm.

[2] Spitzner, L. (2002). Honeypots: Tracking Hackers.US: Addison Wesley. pp 1-430.

[3] Spitzner, L. (2002). Honeypots: Tracking Hackers.US: Addison Wesley. pp 1-430.

[4] ***Proactive Detection of Security Incidents*** *Honeypots, 2012-11-20*

[5] *R. Naraine. Internet explorer feature causing drive by malware attacks. http://blogs.zdnet.com/security/?p=1361*
[6] J.R. Greene, The encyclopedia of Police Science, 3rd ed.,vol. A-I, Taylor & Francis Group, New York, 2007
[7] Organization for Economic Cooperation and Development, "*Malicious software (Malware): A security threat to the internet economy*", OCED Ministerial Meeting, Seoul, Korea, 2008. [Online].Available: http://www.oecd.org/dataoecd/53/34/40724457.pdf
[8] Paul Baecher, Thorsten Holz, Markus Koetter, and Georg Wicherski. Know your enemy: Tracking botnets. The Honeynet Project, 2005.
[9] . www.visustotal.com