Atinder Pal Singh, Birinder Singh / International Journal of Engineering Research and Applications (IJERA) ISSN: 2248-9622 www.ijera.com
Vol. 2, Issue 5, September- October 2012, pp.1135-1142

# Design and Implementation of Linux Based Hybrid Client Honeypot Incorporating Multi Layer Detection

**Atinder Pal Singh*, Birinder Singh****
*(Department of Computer Science, Baba Banda Singh Bahadur Engineering College- Punjab Technical University)
** (Department of Computer Science, Baba Banda Singh Bahadur Engineering College - Punjab Technical University)

## ABSTRACT

**In current global internet cyber space, the number of targeted client side attacks are increasing that lead users to adversaries' web sites and exploit web browser vulnerabilities is increasing, therefore there is requirement of strong mechanisms to fight against these kinds of attacks. In this paper, we present the design and implementation of a client honeypot which incorporate the functionality of both low and high interaction honeyclient solution and incorporate the multi layer detection mechanisms to fight against client side targeted attacks. As low interaction client honeypot are fast in processing of websites but unable to detect zero-day attacks whereas high interaction client honeypots are able to detect zero day attacks but very high resource intensive. On the basis of the problems of existing client honeypots, we formulate the requirements of this hybrid honeyclient solution in terms of defending client side attacks. Our system is tested by visiting of various malicious websites and detection of malwares dropped on the system is detected. Also an approach is also been discussed to deploy the hybrid honeyclient solution for detection of malicious websites and collections of malwares embedded into malicious websites. We are ensuring that most of software tools used in our implementation are open source.**

*Keywords* - **Honeypots, Client Honeypots, Network Intrusion Detection System, Network Security.**

## I. INTRODUCTION

The size of the internet has significantly increased in past few years as well as applications hosted on internet are increasing exponentially. Internet has become the most popular medium of communication and global information reservoir. With the increasing popularity of public social networking sites, the whole universe seems to congregate around internet to get his/her share of web. Though the general impression is the growing cyber security awareness among the masses, but the advanced hacker techniques and sophistication seems to counter the defensive mechanisms easily and befool the users. The malwares propagating in network have become the biggest threat to the increasing internet. From past few years, malwares targeting the client side applications are increasing in evolutionary manner. In response to this increasing malware attacks, honeypots has emerged as one of the popular practical defense technique. The Honeypots are the information system resources capable to attract, capture and collect malware attacks. While the fight is ongoing on the Internet between blackhats and whitehats, attackers have started to transfer the battlefield to the client user; as they believe the client applications are more likely to have security breaches and vulnerabilities. Client user has become the weakest link in the network security chain, and since the security chain is only robust as its weakest link, we need to detect attacks against client side to protect the whole security system [1] [17] [18].

The first implementations of bait system concepts were introduced in information security in the late 1980s [2] [19]. The idea has always been to use deception against attackers by mimicking valuable and vulnerable resources to lure them into exploiting those resources. The purpose of such a strategy is twofold; to gather information on the nature of the attack, attacker, tools and techniques used in the process, and to protect operational hosts and servers.. The first implementations of honeypots focused on imitating server side services and were mainly designed to protect servers and production systems, and gather information on attackers and attacks directed at those services. In recent years, a change in attack behavior has been detected across the internet. Instead of putting all the efforts to target and gain access to production servers, attackers are targeting end-user systems.

Online attacks targeting users' operating system through browser and browser vulnerabilities are common threats. Attackers use social engineering techniques to lure users into downloading and installing malicious software or malware without disclosing their actual intend. Prompting users to install plug-in to watch online videos and mimicking themselves as free antivirus software are two examples of common techniques used by attackers to induce users into installing them. Malicious

website are also deployed which contain code that exploit vulnerabilities of popular web browsers, their extensions and plug-ins. Once such a website is visited by a vulnerable browser, the malicious code is rendered and executed by the browser's engine, resulting in exploitation of a particular vulnerability associated with the browser or the operating system and its applications. The infected client system may then fetch and install malware from a malware server or allow an attacker to gain a full control over the system.

Client honeypots have been designed to identify malicious websites, using signature, state, anomaly and machine learning based detection techniques. There are two kinds of honeypot, server-side honeypot and client-side honeypot. Server-side honeypot is the traditional honeypot. This kind of honeypot must have some vulnerable service, and attacker can detect them, so they are passive honeypots. The concept of client-side honeypot [3] was brought forward by Lance Spitzner. Client-side honeypot aims at vulnerabilities of client applications. It needs a data source, and visits the data source actively, and detects all activities to judge if it is safe. Client-side honeypot actively "requests" to accept attack. This kind of honeypot actively acquires malware spreading through client application software which traditional honeypot can't get [4].

The format of the remaining paper is: section 2, defines and explains the technology that has been employed and discusses the client honeypots in brief and other detection approaches. Section 3 deliberates the framework design and discusses our detailed design of the implemented system. Section 4 discusses conclusion and future work of the research problem.

## II. BACKGROUND AND MOTIVATION

### *A Malware Types*

Malware can be defined as "a set of instructions that run on your computer and make your system do something that an attacker wants it to do" [3]. Once a malicious web page attacks the user's system, then it is able to download malware including the following, as described by Provos, et al. [4] and [5]:

- Virus
- Worm
- Trojans
- Spyware
- Adware
- Root kits

### *B. Malicious URL Detection Approaches*

Figure 1 depicts the approaches used in detection of malicious websites such as in-built browser protection based detection; applying some heuristic based detection and client honeypot based detection approaches. By applying Google safe browsing we can detect malicious URL as well as on the basis of in-built browser plug-ins. We can also detect malicious URL by applying static analysis or some machine learning approaches like pattern-matching or java script features. Client Honeypot based detection of malicious URLs is widely used for analysis of malicious URL which uses the web based propagation medium to infect the end users.

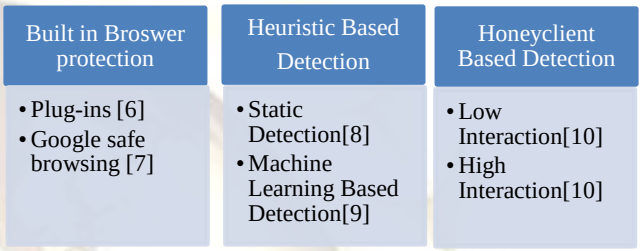


Fig. 1. Malicious URL Detection Approaches

### *Honeypot*

Honeypots are the types of resources whose values are being attacked or probed by the attacker. Generally honeypots are used for gathering the intelligent information about the attacker or black hat community targeting the intrenet cyber space. In terms of network security, it is well accepted that honeypots play a biggest role including other security devices such firewall, IDS etc. To make the network full proof against attacks, honeypots play the major role to protect the network from unknown and unclassified kind of attacks. Honeypots also play an important role in protecting servers and hosts against attacks targeted at resources available on a production network by directing attacks to decoy systems. When placed with other technologies such as intrusion detection system, intrusion prevention system, firewalls, honeypots become highly effective tools against attacks performed by black hat community.

Most of the network security devices such as firewall, intrusion detection system, are based on pre-defined signatures embedded into them to detect the attacks and prevent the network from these kind of attacks but what will happen in case of zero day attacks when there are no signatures exists in their database, honeypots plays biggest roles here to tighten the networks from these kind of nknown attacks. Honeypots are effective tools to detect internal attacks and propagation of worms within an internal network which other tools such as firewalls fail to achieve Usually Honeypots are very different

than other network security devices because they are not directly providing any kind of security to the organizational network but they give us the useful information to study the behavior of attackers so that we can take the remedial actions further.

Honeypots can be classified as per the attack classes and targeted attacks such as server side attacks and client side attacks. Classification of Honeypots can be as server honeypots and client honeypots. Server Honeypots which provide us the deep knowledge of server side attacks, which are a kind of passive honeypots. In contrast to server honeypots, client honeypots provide us the deep knowledge of client side attacks; therefore they are also called as active Honeypots or Honeyclient. Both of the Honeypot Technology has emerged as a widely research areas in the field of cyber security.

***Client Honeypot***

In recent times, black hat community has mainly targeting client side applications such internet browser, pdf, media player etc. The client honeypot is new concept [21]and is quite different than server honeypots. In case of active honeypot, client honeypot acts as client and actively visit the website to see whether the attack has happened or not. Following diagram depicts high interaction and low interaction client honeypots.

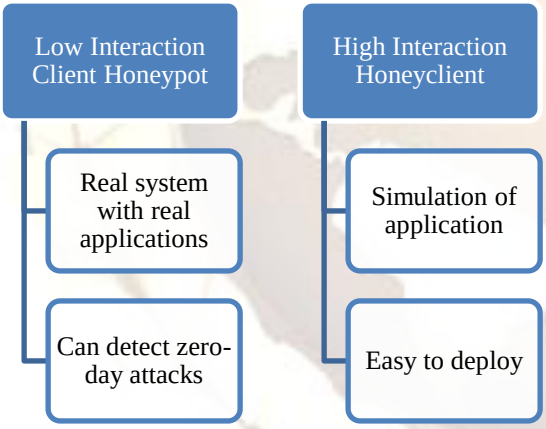


Fig 2. Client Honeypots Classifications

## III. END USER: A SOFT TARGET

There are hosts and servers in production network but end user is simplest and soft target to attack for black hat community as they do not like network security devices such as firewall, network intrusion detection system and they do not have such kind of luxuries as of host or network based intrusion detection system as other security devices. Also they are dependent upon others for updating their system software. Sometimes end users are not so knowledgeable to tackle the attacks occurred on their system. Security of home users mainly revolves around the following security mechanisms [11]:

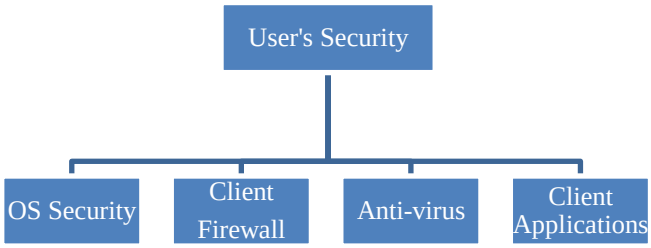


Fig. 2 Home User's Security Mechanisms

### 3.1. Operating system security

Operating system is the platform on which all other user applications run and resembles the foundation of a building. Microsoft is the dominant player in operating system market and has a total share of 94 percent of all operating systems in use. While its latest operating system, Windows 7 is the most stable and most secure operating system, its slow adaptation in home users and corporations has led to wide spread use of older operating systems. According to statistics of the last three month (Nov-Dec-Jan 2010), Windows XP, a popular but vulnerable operating system has a 50% market share while windows Vista owns 15% followed by Widows 7, 28%. Microsoft provides constant updates for these operating systems; it is up to the user to install these updates. A quick look at the operating system security however, confirms that even with constant update and release of service packs, the number of discovered vulnerabilities have not decreased dramatically. Based on the report be Secunia, 75 vulnerabilities were discovered in Windows 7 alone in 2010. Windows Vista follows by 89 followed by Windows XP 97 [12].

### 3.2. Client Firewall

Most recent operating systems come with built in and "enabled by default" firewall package. Starting with Windows XP service pack 2 and since, firewall has been enabled by default on all Microsoft operating systems. This provides basic protection for an average home user. Based on a latest study in European Union countries in 2010 [Internet usage in 2010 – Households and Individuals], less than 50 percent of the users had their firewalled enabled. Users decide to disable windows firewall because of compatibility issues with other programs. This results in significant threat to the security of the host system.

### 3.3. Antivirus

Antivirus software is the basic security tool installed in end user computer. They mostly rely on signature based detection where executable files are matched against a signature database of known

viruses. New versions have run-time scanning feature that scans the file in real time and avoids execution, if a threat is detected. Signature based detection however results in the antivirus engine failing to detect variants of known viruses, therefore a constant update of antivirus signature database is essential to provide basic protection. Although an antivirus is fairly effective in detection of known attacks if updated regularly, they are unable to protect users from remote port attacks or attacks directed at user applications from internet. Latest survey shows that, 25 % of users disabled their antivirus software because they believe this software have negative impact on their PCs' performance [13].While another study by a research group had similar results showing around 23% had absolutely no active security software installed. Rightfully assuming that every computer without any active protection would be infected with one type of virus or malware, 23% makes a huge impact not only on the infected computers but overall security of the internet as these infected hosts will be used to attack other hosts across the internet, be a part of DDoS attacks or exploited to deliver Spam [14].

### 3.4. Applications

Operating system vulnerabilities are most discussed in security community but vulnerabilities in user applications cover most of the vulnerabilities found in an end user system. A study by Secunia points to 729 (Windows XP), 722(Windows Vista) and 709 (Windows 7) available vulnerabilities, in top 50 applications on a typical user system [14]. All these applications are offered by 14 vendors. Based on these numbers, more secured operating system does not necessarily provide a more robust platform unless applications are developed in a secured manner or patched properly [15]. With increasing number of users online, application vulnerabilities are the easiest and the main target of such attacks. Based on [15], 84% of all attacks were classified to be "from remote" while local network or local system each had 7 % of total attacks. These results show the importance of implementing a robust internet security while browsing the internet, especially in applications which directly interact with web contents. Internet browsers are the main tools used to browse and retrieve contents from the internet. With increasing popularity of dynamic internet contents and online services (i.e. online banking), browsers' roles have been more than just to view static contents but rather a part of a new wave of user interaction and experience with the web. With popularity of cloud computing, browser's role will even be more immanent in the usability of novel operating systems. Since browsers are the main tools to interact with online contents, they are a gateway into the host operating systems and local networks. If a browser is exploited, attackers would be able to gain access to system resources and install malicious code on the local system. The infected host then fetches more malicious content from remote locations and targets local hosts. All this is done without firewall blocking any connections. Any NAT implementations are bypassed since once a host is infected all connections are initiated from inside, which is permitted by firewall.

## IV. System Framework Design

Here we present the detailed system design which takes the benefits of both low interaction honeyclient and high interaction honeyclient. As in case of high interaction honeyclient, there are no emulated softwares or applications running into system, it provides a real environment to attackers to take full control of the system, therefore we are able to detect the zero-day malwares targeting the client side applications. Drive-by-download kinds of malwares are classical examples which drop on user's system without his knowledge and concern. As per the honeyclient technology, the end results is to collect the number of malwares which includes classified as well as unclassified class of malwares. Unclassified malwares here we mean the zero day malwares which are not detected by any signature based mechanisms.

In case of client honeypots, we actively visit the URLs to get the client machine gets infected by the malwares dropped on the system. Here we are presenting an example of analysis of URL which is declared malicious by our designed system and we observed that many activities are being performed by that URL and all the activities are being logged into our honeyclient solution.

In this research, we present the virtual box [16] based honeyclient solution which incorporate the extraction of malware binaries from raw PCAP data as well as malwares dropped on a client machine by file system monitoring. Therefore we are able to detect exploits performed by the malicious URL on a client system. We can achieve the functionality of both low and high interaction honeyclient.

Figure 3 depicts the system design of honeyclient solution. As shown in the figure, we are giving all the URLs into virtual machines where we are actively visiting the web links using real browsers. We can browse multiple bulk lists of URLs instead of single web link which increase the efficiency of the high interaction honeyclient and we are extracting the malware binaries by two ways:

- File System monitoring at user space
- Forensic Investigation of raw PCAP data

All honeypots are running in virtualized environment using open source Virtual Box.

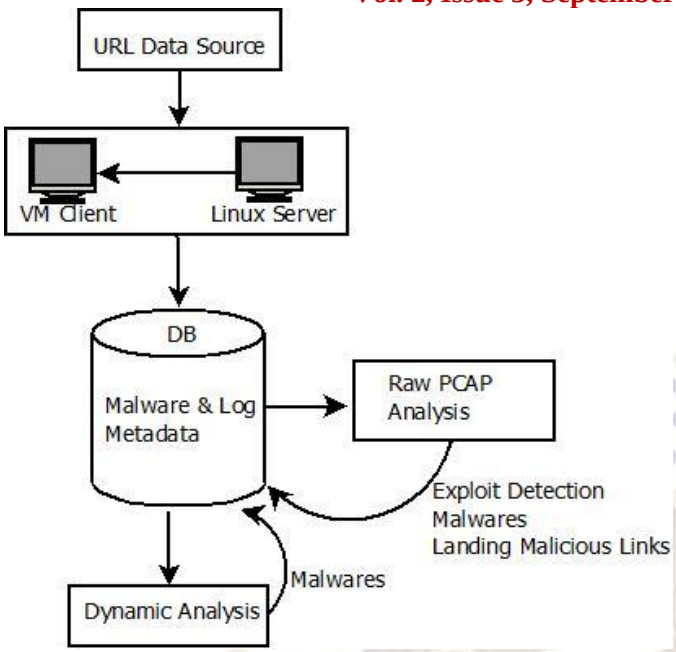


Fig. 3 Virtual Honeyclient

URL lists to the client honeypots are provided by the server called active honeypot controller. In proposed system, there are basically four functional components: URL data source, Virtual Honeypots, Central database, Analysis Modules.

- URL data source
  - Internet Cyber Space { Google safe browsing)
  - Spam Mails
  - Typo squatting URLs
- Virtual Client Honeypots
  - Virtual Box based client honeypot ( Virtual Machines)
- Database
  - Mysql database
- Behavior based analysis
  - Dynamic execution monitoring
    - Network Monitoring
    - File System Monitoring
- Out of Band analysis
  - Extraction of malwares from PCAP data
  - Offline analysis of PCAP raw data
  - Web Interface ( Need to be Developed)
  - User interface for Online URL submission
  - Report generation and download

Here we are claiming that developed honeyclient running in virtualized environment, web interface still need to get developed for report generation and single window interface for automated URL submission instead of manually feeding of web links.

**URL data source:**

As depicted in figure, for URL data source, we are taking malicious URL lists from google safe browsing for testing of our system. We have tested the system and we are able to extract the malwares binaries dropped on a user's system, here on a real virtual machine of Window XP.

**Virtual Honeypots:**

This module plays biggest roles in detection of malicious URLs and collection of malwares dropped on a system. Here Window XP based OS is used for actively visitations of URLs in virtualized environment using Virtual Box. As shown in the figure, base machine which has Linux Operating system feed the list of URLs to client machine which is Window XP virtual machine. Monitoring modules such as file system monitoring, network monitoring is enabled on client machine during active visit of the web links. With the help of file system monitoring at user space, we are able to capture malware binaries dropped on client machine and with network monitoring, we are able to catch raw PCAP data which is later analyzed forensically to extract the malware binaries and to detect the exploits.

- Client Virtual Machine: Window XP with real browser plug-ins
- Server Machine: Linux Red Hat Enterprise Linux, Pentium IV, 4GB RAM, 200GB HDD

Following generic algorithm is used in URL extraction and feed them into virtual machine for actively visiting of them:

1. Select list of URL say N in round robin manner from data source module.
   a. N>=1;
   b. If N is already visited then select another list of URLs, say M;
   c. Rename M=N;

d. Open clean Virtual Machine, visit N list of URLs

e. Stop the machine after complete visit of N URLs

f. If N is not already visited, submit the list of URLs into virtual machine;

g. Repeat steps (b-e)

2. Repeat step 1;

The framework presents multiple layers of detection for malicious URLs for detection of malicious websites:

- State based monitoring and detection like network monitoring, file system monitoring to detect malwares dropped into system (URLs are being executed in real environment using real browser on Window XP operating system).
- Forensic extraction of malwares from raw PCAP data.

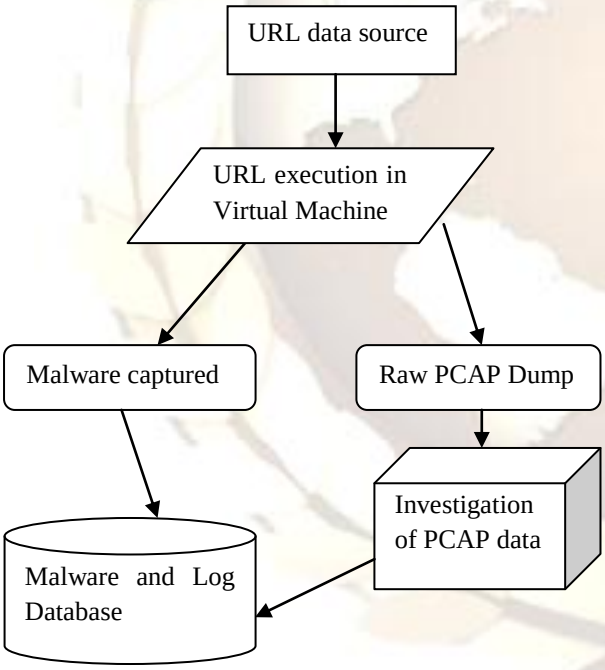


Fig. 4 Process Flow of System

**Database Implementation:**

All the collected malwares and other logs such as state logs, pcap data is saved into database. For data design, we have chosen Mysql for our implementation with which we are quite comfortable. Malwares metadata stored at database are fetched and analyzed further if required. Further raw pcap data is analyzed and malware binaries are extracted from it after making the complete session.

## V. Working and Experimental Results

Here we discuss the working and few experimental results to signify the proper working of our developed and implemented honeyclient solution. To evaluate the system framework, several experimentations have been performed. The URL seed list has been taken from Blacklist Providers consisting of around 250 URL addresses. Following sections describe in detail the resulting statistics established:

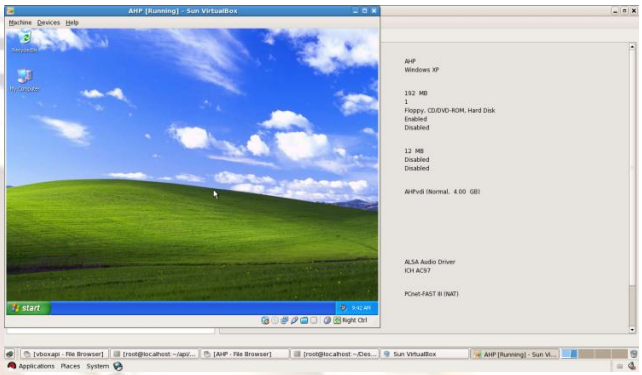

Fig 5. Honeyclient System Snapshot

Figure 5 depicts the working snapshot of the system with Linux operating system on base machine, Window XP operating system on virtual machine. Virtual machine window is opened when there are list of URLs to be visited. Whenever there are list of web links, virtual machine will be pop up for real browsing of URL in round robin fashion.

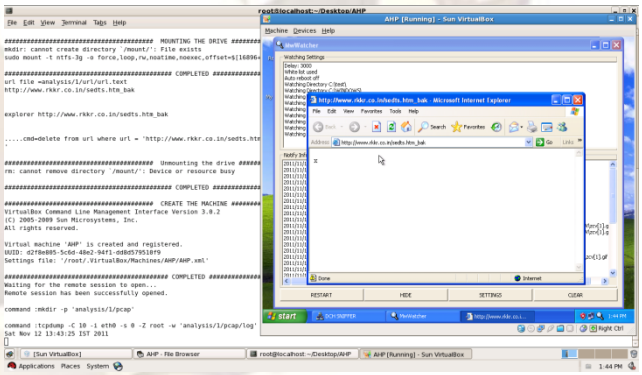

Fig. 6 Snapshot of Working System

Figure 5 depicts the working of our developed system, as shown in figure; base machine is having Linux operating system, send the list of URL to client window XP virtual machine. Client virtual machine is visiting the URL using Internet Explorer. File system monitoring and network monitoring is enabled on client machine. All the logs after complete visiting of URLs has been sent to base Linux server, from there they are inserted into database for further analysis.

All the functionality of the system is well depicted in the above snapshot, whenever there are list of URLs, a virtual client XP machine will be opened and URLs lists will be actively browsed by using real browsers.

After completion of visit process, virtual client machine will be stopped and it will be off completely. For next list of browsing, a clean virtual machine will be opened.

Below figure 6 represent the functionality of honeyclient system with file system monitoring. We can see the complete HTTP communications, number of unique IPs in communications, HTTP servers as well as DNS requests made during the active browsing of the web link.

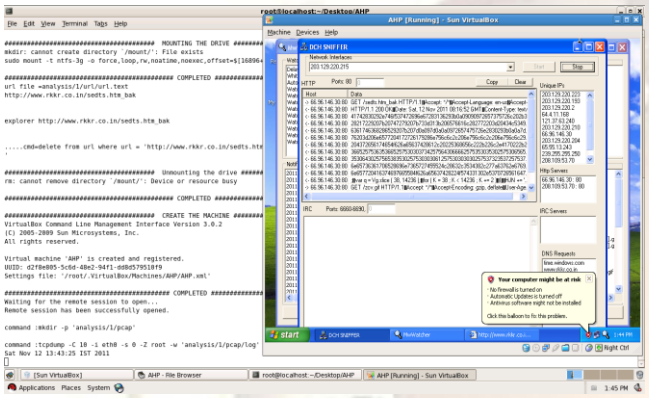

Fig. 7 Honeyclient with File system monitoring

All the logs captured are saved into database which is running on base machine at present for testing purpose. Presently only the prototype system, we have tested for proper functioning of individual modules.

| Malware Name | MD5 |
|---|---|
| sysevwx.exe | D687BCD7CC2A90FF63C9A5CB1BDE122A |
| sysykeh.exe | D687BCD7CC2A90FF63C9A5CB1BDE122A |

Table 1: Captured Malware Samples

The malware samples captured after visiting a one of malicious URL on our honeyclient solution is being depicted in table 1. This is just one example; we have a more number of malwares samples captured which can be used for analysis. Column 1 depicts the executable malware binaries dropped on a client victim machine, column 2 represent the corresponding MD5 value of this. As we are claiming that this is just one example of downloaded malwares samples, we have larger set of data set which can be shared on demand basis.

**PROJECT SUMMARY**

The summarization of various software and hardware used in our project is depicted by the following table 2. In this project we are ensuring that most of the tools are free and open source. For Virtualization technology, we are using VirtualBox [16]. We are running different Virtual machines on single Red hat linux based Operating System. We are using minimum memory of 4GB but large amount of memory is preferred to run Virtual Machines in Virtualized environment.

| **Project Summary** | | |
|---|---|---|
| **Feature** | **Product** | **Specs** |
| **Host Operating system** | Red Hat Enterprise Linux 5 | **HW Vendor: HP Server Proliant ML 350**<br>**Processor: 2.33 GHz Processor**<br>**RAM :**4GB RAM<br>**Storage:** 2x146 GB<br>**NIC :** 1 GB Etherenet controller |
| **Guest Operating System** | Microsoft XP SP2 | Single processor Virtual Machine<br>**RAM** 256MB<br>**NIC** 100Mbps host-only vmnet |
| **Virtualization Software** | VirtualBox | Virtualbox3.0.2 for linux |
| **Architecture** | Hybrid Honeyclient | High Throughput honeyclient |
| **Packet Capturing** | TCPDUMP | Tcpdump tool |
| **File System Monitor** | Mwatcher | Real Time File System Monitor tool |

Table 2: Software/Hardware used in project implementation

## V. CONCLUSION

During the course of our research work, we implement client honeypot which incorporate the

multi level detection mechanisms to detect the malicious websites based on linux operating system and by using various open source tools like tcpdump, VirtualBox. We introduced the category of Internet malware, the client side attack techniques and overall framework of the system in detail. We mainly gave the design and implementation of client honeypots based malware collection. During the work done so far, client honeypot based solution is very useful to collect the internet malwares and to detect the malicious websites.

Hybrid or multiple level analysis also has the advantage of providing an opportunity to detect malicious content at different stages which can be used as an inline or offline strategy.

Our developed Virtual Box powered Honeyclient is very useful for collection of internet malwares but it is having a limited capabilities or we can say that it is just a prototype. There is a requirement of integration of crawler as data acquirement, in present system, there is no such component in our developed module. Further there is also a possibility of addition of various client side applications such as firefox, pdf etc because currently we only using Internet Explorer for actively visiting the websites. And there is also a possibility of addition of automatically analysis of collected malwares. We can confirm that we cannot cover all the challenges such human user simulation, logic bomb, time triggered websites but we have developed a prototype solution to get better understanding of client honeypots.

**ACKNOWLEDGEMENTS**

We would like to sincerely thank Dr. Birinder Singh, HoD, Department of Computer Science for his contribution and help in writing this paper.

## References

[1] R. Danford, ―2nd Generation Honeyclients‖, SANS Internet Storm Center,2006 http://handlers.dshield.org/rdanford/pub/Honeyclients_Danford_SANS 06.pdf

[2] Cheswick, B. (1990), An Evening with Berferd in which a cracker is Lured, Endured, and Studied: *Citeseer*.

[3] Ramaswamy, C. and R. Sandhu. (1998), Role-based access control features in commercial database management systems: *Citeseer*.

[4] Skoudis, E., and Zeltser, L., *"Malware: Fighting Malicious Code"*, Prentice Hall,2003, Page 3, ISBN = 978-0131014053.

[5] [Provos, N., McNamee, D., Mavrommatis, D. W., K and Modadugu, N., *the Ghost In The Browser Analysis of Web-based Malware*. 2007. [Online]. Available at:http://www.usenix.org/events/hotbots07/tech/full_papers/provos/provos.pdf [Accessed 11 Feb 2009]

[6] Secure Browsing | Malware Protection | Trustwave https://www.trustwave.com/securebrowsing/

[7] Google Safe Browsing www.google.com/tools/firefox/safebrowsing/

[8] Detection and Analysis of Drive-by-Download Attacks and Malicious JavaScript Code www.cs.ucsb.edu/~vigna/.../2010_cova_kruegel_vigna_Wepawet.pdf

[9] Prophiler: A Fast Filter for the Large-Scale Detection of Malicious Web Pages www.cs.ucsb.edu/.../2011_canali_cova_kruegel_vigna_Prophiler.pdf

[10] Xiaoyan Sun, Yang Wang, Jie Ren, Yuefei Zhu and Shengli Liu, "Collecting Internet Malware Based on Client-side Honeypot", 9th IEEE International Conference for Young Computer Scientists (ICVCS 2008), pp. 1493 – 1498, 2008.

[11] Secunia. (2010), "Factsheets by Windows Operating System - 2010". 20 - March - 2011]; Availablefrom: http://secunia.com/resources/factsheets/2010_win_os/.

[12] PcPitstop. (2010), "The State of PC Security". 20 - December 2010]; Available from:http://techtalk.pcpitstop.com/2010/05/13/the-state-of-pc-security/.

[13] Secunia. (2010), "Secunia Yearly Report - 2010". Available from:secunia.com/gfx/pdf/Secunia_Yearly_Report_2010.pdf.

[14] Secunia. (2010), "Research Reports, Factsheet by Browser - 2010". [Cited 2011 5 - January]; Available from: http://secunia.com/resources/factsheets/2010_browsers/.

[15] VirtualBox. (2004). Sun VirtualBox® User Manual. Available:http://www.virtualbox.org/manual/UserManual.html Last accessed 20 July 2008.

[16] Sanjeev Kumar, at al, Hybrid Honeypot Framework for Malware Collection and Analysis, ICIIS-2012, IIT Chennai

[17] www.honeyclient.org

[18] en.wikipedia.org/wiki/Client_honeypot

[19] www.honeynet.org

**Journal Papers:**

*[20]* Masood Mansoori and Ray Hunt, *International Journal of Network Security & Its Applications (IJNSA), Vol.3, No.5, Sep 2011, AN ISP BASED NOTIFICATION AND DETECTION SYSTEM TO MAXIMIZE EFFICIENCY OF CLIENT HONEYPOTS IN PROTECTION OF END users*

***Thesis:***

*[21] Yaser Alosefer, Analysing Web-based Malware Behaviour through Client Honeypots Cardiff University School of Computer Science & Informatics, Feb-2012*