Design and Implementation of Malware Collection System Based on Client Honeypot

Manpreet Kaur Rangian, Upasna Attri

Abstract— These days' client users on internet are the main target for attackers, through a victim client user an attacker can spread his malware to a wide range on a network. The various security mechanisms are available to secure our systems on a network, but a more robust mechanism is required in support and to find security loopholes. Client Honeypot is a really very effective and beneficial to provide security. In this paper we provide design and implementation of a malware collection system based on client Honeypot. This is an active Honeypot. After designing and implementation we examined this system by visiting malicious URLs and we find malwares dropped on the system. Informative data obtained by the system can be further analyzed and used to enhance the security system. Collected malwares are categorized based on type of malware program.

Index Terms— Client Honeypot, Client side attacks, Honeypot, Malwares, security.

1 Introduction

THE internet represents a wide resource of information, identifying a collaborative process between organizations and individuals, But using the internet can become a dangerous experience for those with low levels of security. Number of malwares exits on internet such as virus Trojan horse, worm, spyware and root-kit. Generally, web attacks are divided into two forms

Server-side attack: such attacks use window RPC service vulnerabilities to attack the server that provides some services

Client-side attack: Here the attacker uses the client side application vulnerabilities. Examples of client applications are: web browsers, media players. The aim of the Honeypot is to collect data on attacks and attackers by monitoring the machine being attacked.

Honeypots are divided into two main types: passive and active Honeypots. A passive Honeypot involves setting up a vulnerable system or service, or possibly their simulation, and then monitoring activity to detect any attack on the system.

An active Honeypot is a Client Honeypot [5]. Active Honeypot behave as client and interact with the server to test whether and attack has happened.

This paper is organized as follows:

Second section explains some background aspects of the technology on client side Honeypot. Third section introduces design issues for Client Honeypot. Fourth section gives design and implementation details. Section fifth shows experimental results and sixth section gives conclusion and future work.

- ***Manpreet Kaur**, Department of Computer Science engg. (Batch 2011), Indo Global College of Engineering, Abhipur, Mohali (Punjab), Punjab Technical University, India.*
  manpreet198820@yahoo.com

- ***Upasna Attri,** Department of Computer Science engg., Indo Global College of Engineering, Abhipur, Mohali (Punjab), Punjab Technical University, India.*
  upasnaa08@gmail.com

2 Background

2.1 Honeypot

Honeypot is a computer connected to a network. It can be used to examine vulnerabilities of the operating system or network.

In the realm of Honeypots the security holes are opened on purpose. In other words Honeypots welcome hacker and other threats. The purpose of a Honeypot is to detect and learn from attacks and use that information to improve security. A network administrator obtains first-hand information about the current threats on his network. Undiscovered security holes can be protected gained by the information from a Honeypot. On a Honeypot every packet is suspicious. The reason for this is that in a Honeypot scenario, the Honeypot is not registered to any production system. Therefore any device establishing a connection to a Honeypot is either wrong configured or source of an attack [10]. The most important benefit is that a Honeypot detects attacks which are not caught by other security systems.

2.2 Client Honeypot

The concept of client-side Honeypot was brought forward by Lance Spitzner (2004). The concept of Client Honeypot is quite different from the traditional Honeypots. Instead of passively waiting for attackers, the Client Honeypot will go and search for attackers. The Client Honeypot acts as a client and Interacts with the server to study it and determine if an attack has happened. Client Honeypot needs a data source, and visits the data source actively, and detects all activities to judge if it is safe. Capture-HPC [4]: is an open source high-interaction Client Honeypot developed at Victoria University, Wellington. HoneyMonkey [9]: is a high-interaction Client Honeypot produced by Microsoft This tool simply uses Internet Explorer to visit servers and monitors processes, the registry and files.

2.3 Security attacks against client user

A typical example of a client-side attack is a malicious web page targeting a specific browser vulnerability that, if the attack is successful, would give the malicious server complete control of the client system. Client-side attacks are not limited to the web setting, but can occur on any client/server pairs, for example e-mail, FTP, instant messaging, multimedia streaming, etc.

Client-side attacks currently represent an easy attack vector because most attention in protection technology has been focused on the protection of exposed servers from remote attackers [12]. Over the past few years, attacks against web applications have become more prevalent and sophisticated. There are several methods of attacking web applications, SQL injection being one of the more well-known.

Form field injection .In this type of attack, malware interacting in a web browser adds additional form fields to valid form fields on a web page. The purpose of injected fields is to trick users into revealing sensitive personal information like passwords, ATM PINs, and credit card numbers.

HTML injection attacks take form field injection one step further. Instead of inserting a form field into a web page, HTML injection replaces legitimate HTML coming from the server, similar to a "cut and paste" function. The replaced HTML is overwritten by the attacker's HTML and the original, intended content is never rendered by the web browser [13].

Malicious websites often use the *Code Obfuscation*
Attacker usually wants to hide the exploit vector by using various encoding options to make the code unobvious and hard to interpret. This technique aims for evading static detection tools such as IDSs, anti-virus tools, and firewall filters. Obfuscation means using encoding to make the code ambiguous, and more difficult to interpret [14].

Vulnerability exploitations often download malware or directly execute some command to install malware.Client application files includes word, PowerPoint, Excel PDF files and so on. The exploitations of these files contain some shell-code or decoded shell-code in the binary files. So attackers study the file format, and insert shell code in the file without influencing the file format, malware will be executed when people open themalicious files [15].

Drive-by Download
A drive-by download is an attack where servers can change the state of client system without user consent, or Knowledge, which usually means the ability of malicious server to download and install a program to client system without user consent. In April 2007, researchers at Google found hundreds of thousands of malicious web pages that initiated drive-by downloads attacks [3].

3 Issues Related To Design Of A Client Honeypot

1. The Client Honeypot system must appear like a real system. That is why sniffer based systems are used along with Honeypots

2. Client Honeypot must be designed in such a way that an attacker cannot easily use the Honeypot as a launch point for further attacks against networks.

3. Client Honeypot will help to teach you how to detect such attacks and how to "clean up" after them

4. Client Honeypot system should work in real time to found malwares.

4 Malware Collection System Based On Client Honeypot

The proposed Malware collection system based on Client Honeypot is a complete solution for collecting malware samples which drop on a user’s system by visiting the malicious domains, or we can say malwares which are using the HTTP based propagation vector, which propagates by exploiting known and unknown vulnerabilities existed in victim’s browsers. Therefore system is able to detect unclassified attacks. Unclassified attacks means attacks those are not detected by any static signature based machine. We have used some open source tools.

Client Honeypot installed with required browser plug-ins

This is established using virtualized environment with the help of Virtual Box [6], we can create multiple OS on a single machine which reduce the cost of hardware requirement. We had created the Window XP machine virtual machine in which all the required applications are being installed such as Internet Explorer, Adobe etc. During the visitation of every URL, this virtual machine will get opened up and visit the submitted URL for a supplied durations. After the completion of visitation process, virtual machine will get powered off and a new clean machine will be opened during the next URL visitations. Thereby it is also providing a kind of containment environment which tells that if this virtual machine is being affected by any malwares then it may not be able to affect the other machine deployed in same network zone. There are various parts of the system to complete the task. 1) URL insertion, 2) Window XP based visitor, 3) Network monitoring, 4) Database, 5) Malware extractor

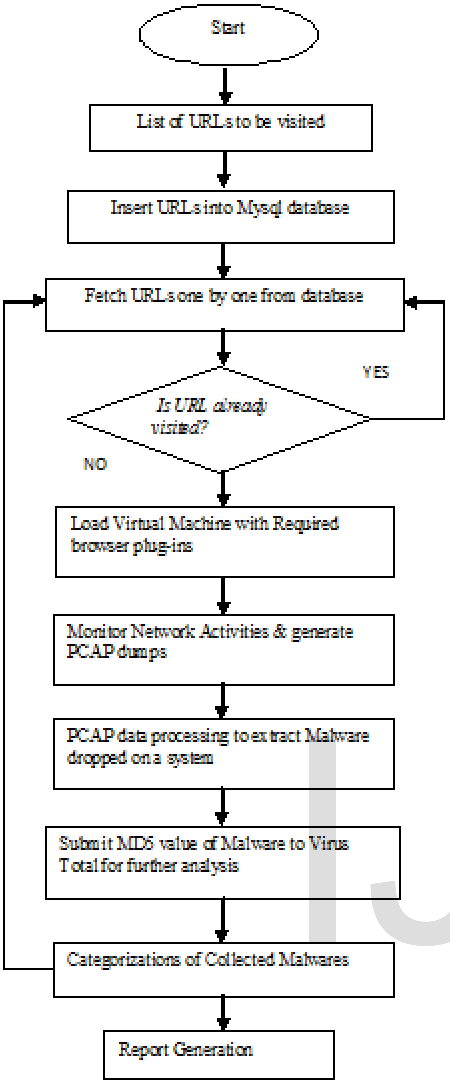

Start
List of URLs to be visited
Insert URLs into Mysql database
Fetch URLs one by one from database
YES
Is URL already visited?
NO
Load Virtual Machine with Required browser plug-ins
Monitor Network Activities & generate PCAP dumps
PCAP data processing to extract Malware dropped on a system
Submit MD5 value of Malware to Virus Total for further analysis
Categorizations of Collected Malwares
Report Generation

Fig 1. Flow Chat for system process

URL insertion

URL list is inserted into database. We take URL list from google safe browsing .This URL list is feeded to Client Honeypot. Spam mails also are source for suspected URLs.

Window XP based visitor

The URL are being fetched from database and visited on a window XP machine with help of Internet Explorer installed with required plug-ins of a browser. Attack data collection and logging into system events and network events is performed in the form of system data and network PCAP data

Network monitorin

Network monitoring of all communications from windows visitor machine to the outside world is performed
1) Standard Data capturing tools – TCPDUMP.

2) Hardcoded within the code to get live network monitoring

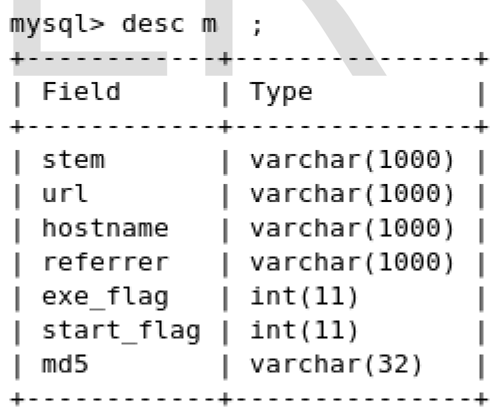
3) Network dumps in the form of PCAP data are being generated corresponding to each URL which is later being processed with data processing engine to extract the malwares dropped on a victim machine.
Fig 1 shows steps and flow.of the working system.

Malware Extractor

The malware extractor module perform the Network PCAP data processing and extraction of executables from network data which is being later submitted to Virus total [7] (a 3rd party analysis tool) to get the categorization of collected malwares samples.

Database

A data base schema is designed and implemented to store URLs and results after visitation of those URLs. Informative data of extracted executable, those potentially are malwares is stored in database along with their unique md5 values. MySQL is used for database creation. Fig 2 shows description of one of the table of database.



```
mysql> desc m  ;
+------------+---------------+
| Field      | Type          |
+------------+---------------+
| stem       | varchar(1000) |
| url        | varchar(1000) |
| hostname   | varchar(1000) |
| referrer   | varchar(1000) |
| exe_flag   | int(11)       |
| start_flag | int(11)       |
| md5        | varchar(32)   |
+------------+---------------+
```

Fig 2. Description of a database table “m”

Process flow of the proposed system is shown in Fig 3. It is important to mention that the application interface is still needed to develop for automatic report generation instead of using manually report generation.
When Client Honeypot interact with URLs and vist them using internet explorer, results are send back to the Linux base machine. During the activity, network data is captured and dumps are generated. The data is processed and Md5 values are generated. The retrieved information is stored into database. After processing all, the executable and binary files are shown on the base machine.

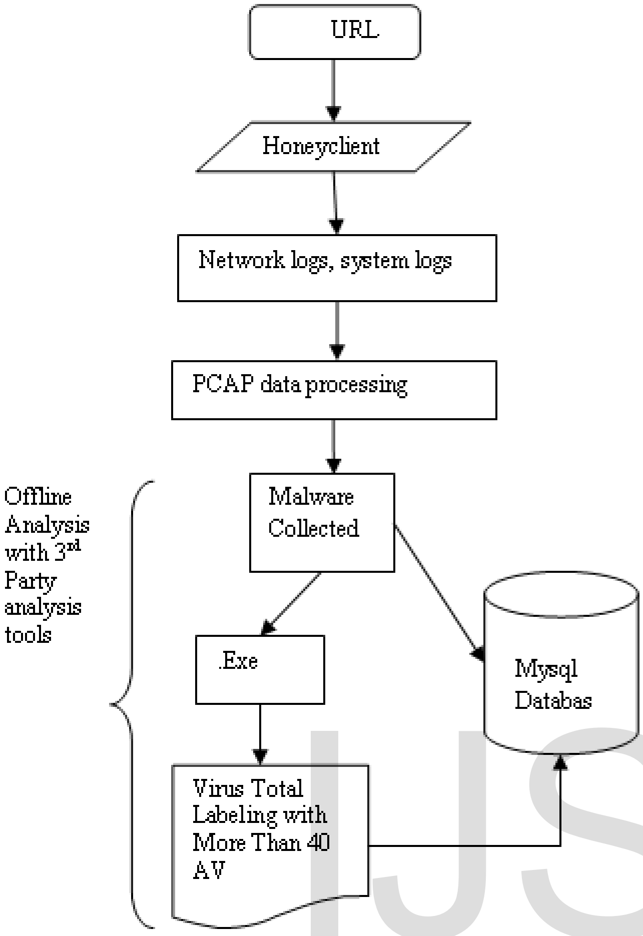

URL
Honeyclient
Network logs, system logs
PCAP data processing
Offline Analysis with 3rd Party analysis tools
Malware Collected
.Exe
Mysql Databas
Virus Total Labeling with More Than 40 AV

Fig 3. Client Honeypot Process flow

5 Experimental Results And Working

To convey developed system some experimental results and working of system is shown in this section.

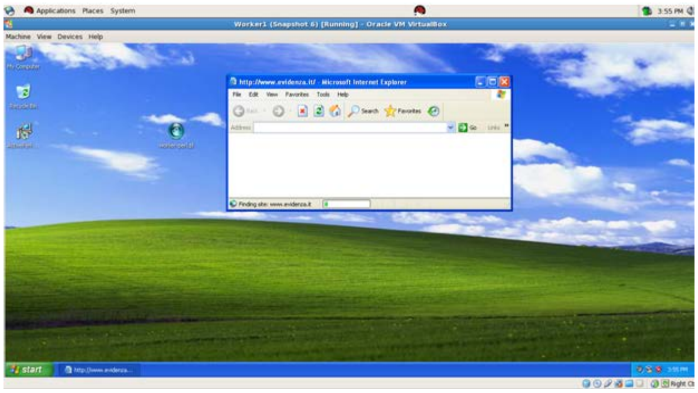
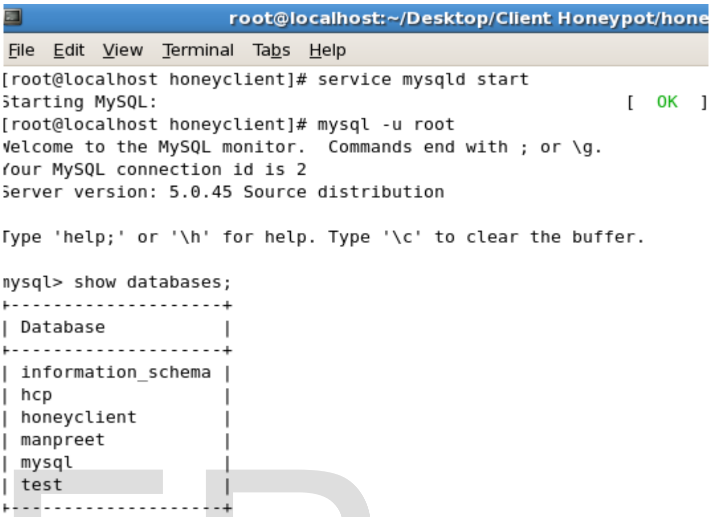

Applications Places System
Worker1 (Snapshot 6) [Running] - Oracle VM VirtualBox
Machine View Devices Help
http://www.evidenza.it/ - Microsoft Internet Explorer
start

Fig 4. Running Client Honeypot

Fig 4 shows snapshot of working system where Linux is host machine and XP window is a guest machine.

Virtual machine is running with internet connectivity. Whenever URLS are there the internet explorer will be up to vist those links. Fig 5 is a snapshot of databases avialable to the root user. Mysql service gets started and the database is accessible on base machine.

After visiting URLs information is stored in database on base machine and honeypot window will be off. A new clean window will open for next time.

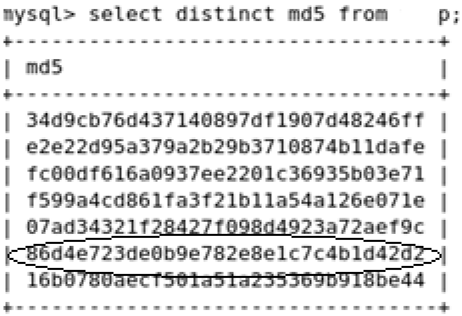

root@localhost:~/Desktop/Client Honeypot/hone
File Edit View Terminal Tabs Help

```
[root@localhost honeyclient]# service mysqld start
Starting MySQL:                                            [  OK  ]
[root@localhost honeyclient]# mysql -u root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.0.45 Source distribution

Type 'help;' or '\h' for help. Type '\c' to clear the buffer.

mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| hcp                |
| honeyclient        |
| manpreet           |
| mysql              |
| test               |
+--------------------+
```

Fig 5. Mysql service starts

Fig 6 shows the md5 values for the extracted malwares Using ‘distinct’ command on a table ‘p’ of the database. Malware files with md5 values can be submitted to an analyzer.

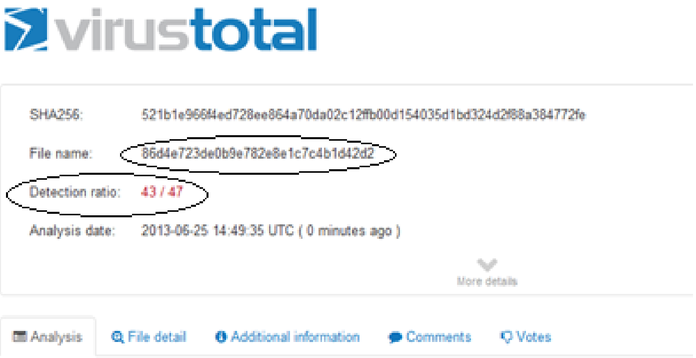
Md5 hash is a digital fingerprint of a file. It is very unlikely that any two non identical files have same md5 hash, unless they have specially created to have same md5.



```
mysql> select distinct md5 from    p;
+----------------------------------+
| md5                              |
+----------------------------------+
| 34d9cb76d437140897df1907d48246ff |
| e2e22d95a379a2b29b3710874b11dafe |
| fc00df616a0937ee2201c36935b03e71 |
| f599a4cd861fa3f21b11a54a126e071e |
| 07ad34321f28427f098d4923a72aef9c |
| 86d4e723de0b9e782e8e1c7c4b1d42d2 |
| 16b0780aecf501a51a235369b918be44 |
+----------------------------------+
```

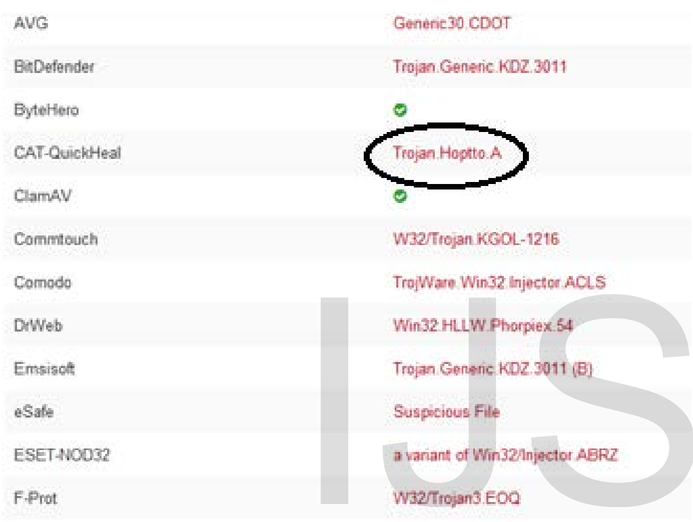
Fig 6.Values of captured malwares.

The executable malware files are submitted to virus total [7] to get categorization of malwares. The virus total module works under the Binary up-loader module. During the process when binary i.e. .EXE files are inserted into the database, the Virus total detail of the binary is also generated.



SHA256:	521b1e966f4ed728ee864a70da02c12ffb00d154035d1bd324d2f88a384772fe
File name:	86d4e723de0b9e782e8e1c7c4b1d42d2
Detection ratio:	43 / 47
Analysis date:	2013-06-25 14:49:35 UTC (0 minutes ago)

Fig 7. [7] Detection Ratio of Popular Anti-viruses



AVG	Generic30.CDOT
BitDefender	Trojan.Generic.KDZ.3011
ByteHero	✓
CAT-QuickHeal	Trojan.Hoptto.A
ClamAV	✓
Commtouch	W32/Trojan.KGOL-1216
Comodo	TrojWare.Win32.Injector.ACLS
DrWeb	Win32.HLLW.Phorpiex.54
Emsisoft	Trojan.Generic.KDZ.3011 (B)
eSafe	Suspicious File
ESET-NOD32	a variant of Win32/Injector.ABRZ
F-Prot	W32/Trojan3.EOQ

Fig 8.Trojan detection by antivirus QuickHeal [7]

Table 1: Categorization of malware samples

Md5	Malware Types
34d9cd76d437140897df1907d48246ff	VirTool.Ceelnject.A Generic.pc Trojan –spy.win32.Zboot.jrvc
07ad34321f28427f098d4923a72aef9c	Trojan.win32.Agent.xtqn W32/Hamweq.worm.av Worm.Hamweq.DD5
16b0780aecf501a51a235369b918be44	HEUR:Trojan.Win32.Genric PWS.Zboot.gen.ary Worm.GAMARUE.B .

Detection ratio for one of the submitted malware file is shown in fig 7 and fig 8 shows the snapshot of malware detection by antivirus quick-heal for malware file extracted by this system.These are the results obtained from our developed prototype system. This needs some enhancements to make a more reliable system.

Table 1.Shows categorization for 3 of collected malware samples .it contains Trojan, worm and root-kit programs.

6 Conclusion And Future Work

The basic task of a honeypot is to provide data which is useful to find type of attacks and techniques used by attackers. The data obtained by system can be used to find shell code or scripts using reverse engineering.

In this paper we introduced attacking techniques used by malicious websites and issues related to design of a Client Honeypot. We give the design and implementation details of our malware collection system.

In this work, malwares were collected using Client Honeypot. Metadata obtained by system can be analysed to make inviolable security system.

Our system is really effective in detecting malwares. It is low resource requiring and provide high throughput. The malware collection system can be made to work at application level.For further enhancement of the system, tool for automatic visiting of web sites can be added and an analyzing tool for analysis of collected malwares can be integrated with the system.

Acknowledgment

We wish to thank Satinderpal Ahuja, Head of Department of Computer Science & engineering, for his continuous encouragement and supervision throughout the course of present work.

References

[1] Qassrawi, M.T. ; Hongli Zhang , "Using Honeyclients to Detect Malicious Websites " e-Business and Information System Security (EBISS), 2010 2nd International Conference, Digital Object Identifier: 10.1109/EBISS.2010.5473642
Publication Year: 2010 , Page(s): 1 – 6

[2] Alosefer, Y.; Rana, O. "Automated state machines applied in Client" Future Information Technology (FutureTech), 2010 5th International Conference, Digital Object Identifier: 10.1109/FUTURETECH.2010.5482695
Publication Year: 2010 , Page(s): 1 - 8

[3] Narvaez, Julia ; Endicott-Popovsky , Barbara ; Seifert, C.; Aval, Chiraag ; Frincke, D.A. "Drive-by-Downloads" System Sciences (HICSS), 2010 43rd Hawaii International Conference on , Digital Object Identifier: 10.1109/HICSS.2010.160
Publication Year: 2010 , Page(s): 1 - 10

[4] Hengya Liu ; Dongmei Zhang ; Gengyu Wei ; Jinxin Zhong, "Detecting malicious rootkit web pages in high-interaction Client Honeypots" Information Theory and Information Security (ICITIS), 2010 IEEE International Conference

, Digital Object Identifier: 10.1109/ICITIS.2010.5689538
Publication Year: 2010 , Page(s): 544 – 547

[5] Alosefer, Y. ; Rana, O.F. "Predicting client-side attacks via behavior analysis using honeypot data." 2011 7th International Conference on Next Generation Web Services Practices (NWeSP), Digital Object Identifier: 10.1109/NWeSP.2011.6088149
Publication Year: 2011 , Page(s): 31 - 36

[6] Oracle VM Virtual Box Documentation http://www.oracle.com/technetwork/server-storage/vm/template-1482544.html

[7] Virus Total, a free service for scanning binaries with multiple antivirus products. www.virustotal.com

[8] Budiarto, R. ; Samsudin, A. ; Heong, C.W. ; Noori, S. "Honeypots: Why We Need A Dynamics Honeypots?"
Information and Communication Technologies: From Theory to Applications, 2004. Proceedings. 2004 International Conference , Digital Object Identifier: 10.1109/ICTTA.2004.1307887
Publication Year: 2004 , Page(s): 565- 566

[9] Gaurav Kaushik and Rashmi Tyagi, "Honeypot : Decoy Server or System Setup Together Information Regarding an Attacker", VSRD International Journal of CS & IT ,Vol. 2 (2), 2012

[10] Master thesis by Christian Döring , "Improving network security with Honeypots"
old.honeynet.org/papers/individual/Mastersthesis_Doering.pdf

[11] Van Lam Le ; Welch, I. ; Xiaoying Gao ; Komisarczuk, P. "Two-Stage Classification Model to Detect Malicious Web Pages" Advanced Information Networking and Applications(AINA), 2011 IEEE International Conference Digital Object Identifier: 10.1109/AINA.2011.71
Publication Year: 2011 , Page(s): 113- 120

[12]] Client-side attacks
http://www.honeynet .org/node/157

[13] client side web application attacks , http://computer-forensics.sans.org/blog/2010/03/23/client-side-web-application-attacks

[14] C. Seifert, "Know Your Enemy: Behind the Scenes of Malicious Web Servers", The Honeynet Project, 2008,
http://www.honeynet.org/papers/wek

[15] Exploit computer security http://en.wikipedia.org/wiki/Exploit_(computer_security)