ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

Intrusion Detection System with Deceptive Virtual Host

Priya B. Bhosle and Prof. M. V. Desai

Dept. Computer Engineering, RMD Sinhgad School of Engineering, Pune, India

ABSTRACT: Honeypots are gaining importance as a useful security tool alongside firewalls, IDSs and antivirus software. Honeypot are intended for the early detection of attacking activity. Honeypots are intentionally configured to be vulnerable to compromise. Honeypot can be a computer, printer, router or practically any networked device that has value to a potential intruder. Protecting the typically large number of assets for which administrators are responsible is challenging task. In the control system these cyber devices may be coupled with the physical processes. Honey pots are effective and efficient tool used for gain more information about the attacker and observing network intruder activity. This paper introduces a design and implementation for self-configuring honeypots that actively examines network traffic. In this paper a novel four-step algorithm used which was developed for autonomous creation and update of a honeyd configuration. Honeyd when deployed creates virtual hosts. Virtual honeypots are configured in such way that it can dynamically configure itself and actively learn from the network traffic to detect the malicious data packets or users. This proposed system uses an unsupervised network attack detection system based on the multiple clustering algorithms. For unsupervised network attack detection the combination of Subspace and evidence accumulation clustering is used.

KEYWORDS: Intrusion Detection System; Dynamic virtual Honeypot; Honeyd.

I. INTRODUCTION

Honeyd is a low-interaction virtual honeypot that simulates virtual computer systems at the network level. Attacker are deceived by simulating the network stack of various operating systems, thus making him believe that he is interacting with a real system over the network. Honeyd simulates only the network stack rather than the entire OS which ensures that even if the honeyd gets compromised attacker cannot damage to large extent. Honeyd can be combined with a virtual machine (VM) to simulate multiple operating systems. To order to add the realism, honeyd simulates arbitrary network topologies which convince the hacker that he/she is on a real system.

A. Level of interaction

Classification is based on the level of interaction which is provided to the anomalous user by the honeypot system. If an interactive environment is presented, then there is more chance of becoming the honeypot as target, which enables to gather more accurate information about attacker.

- Low Level Interaction: One or more simple services are made available which log all communication attempts to specific services, like a web or SSH server. These are just simple daemons which provide the person who configured them a passive way to monitor attack attempts. Basically host operating system is not vulnerable to attacks. Therefore low-interaction honeypots are safe to run. But at the same time unable to be used where a more complex, interactive environment is needed, such as SMTP server.
- Medium Level Interaction: Medium level honeypots begin to emulate collections of software to present more attractive front to the attacker, but still able to protect the host OS. Emulating a collection of software is quite complex task as the emulated programs should respond the same way as their real counter parts. There are more points of attack for the anomalous user, hence chance of system compromise is raised.
- High Level Interaction: Honeypots presented the complete operating system to the attacker, with actual instances of programs. The goal of high-interaction honeypots is for the attacker to gain root access on the machine and then monitoring the activities. This level of honeypot has the highest risk, with highest potential

for collecting information. Such honeypots need constant supervision as the attacker will actually control it. Also attacker could try to use it as a jumping to point for further attacks.

B. Why use a Honeypot?

Honeypots helps to early detection of attacks over network. Any activity occurring on a honeypot, is unauthorized by definition. Honeypots provide information about active attacks and let you collect evidence against an attacker. Few honeypots are able to identify and locate the intruder. A honeypot typically doesn't prevent attacks rather slowing down intruders as they attack virtual targets instead of production computers. As proposed technique essentially inviting intruders to hang out in system, this is putting your organization at increased risk should the intruder compromise the honeypot. If an intruder uses honeypot to attack other systems, it may also face potential legal liabilities.

II. RELATED WORK

Dynamic virtual honeypots is used to gain more information about the attacker using Honeyd and intrusion detection system. DHP solutions that gather network information, process that information into a configuration, and deployed appropriately. This paper proposes active, passive scanning method to gather network information.

These is another existing system i.e SNORT[13] used for rule based intrusion detection. SNORT system can perform only a signature base attacks, this type of attacks are already exists. It detects attacks in packets using pattern matching. Snort cannot analyze further the attempt and analysis of attack. Snort demand a high degree of skill.

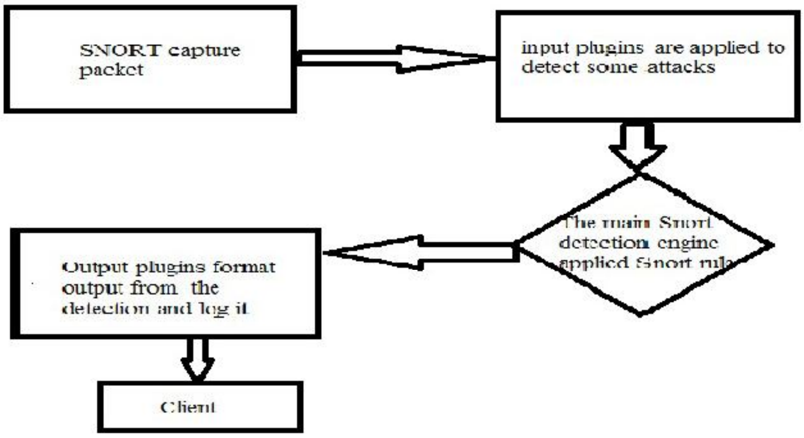

SNORT capture packet
input plugins are applied to detect some attacks
The main Snort detection engine applied Snort rule
Output plugins format output from the detection and log it
Client

Fig.1. Simplified block diagram for Snort

SNORT system can perform only signature based attacks. SNORT can gives the protection to system against the well known attacks. If any new attack is arrival in system SNORT cannot provide security to system.

Anomaly Detection System [8] for detection network anomalies on the basis of network traffic parameters. Every network has some global variable and relation of this variable is fixed, these variables are used to detect anomaly attack in network system. Anomaly Detection System use network traffic parameter to draw a specific network traffic curve for each network that does not change over the time. This curve drawing total flow in the network and there is some peak in the curve which is caused by some network problem. The peak in curve responsible to anomaly behavior in network. This system uses Ntop[10] tool for monitoring network traffic. The Ntop is capturing and analyzing the network packet and store information in database. The Ntop measure all network parameter to detect network anomalies and store information into the database. This system gives advantage to detect both signature attacks and anomaly attacks in network, but main disadvantage of this system is that it does not describe what an attack is and gives high false positive rate.

Control system honeypots have two related paper such as the supervisory control and data acquisition (SCADA) [11]Honeynet project by Matthew Franz and Venkat Pothamsetty of the Cisco Critical Infrastructure Assurance Group (CIAG) was initially released in March 2004 [12].Which is not actively maintain. SCADA honeypot does not create automatic virtual hosts. So it required manually conguration.

Digital Bond, Inc.[12] is a control system security consulting and research group founded by Dale Peterson. This system uses two virtual machines instead of Honeyd. One virtual machine includes SNORT tool for intrusion detection, which perform only signature based detection. Another virtual machine is PLC in which no dynamic hosts or services.

This paper remove drawback of SNORT system by providing both signature based detection as well as anomaly based detection. As in SCADA honeypot not provide automatic virtual hosts. This paper provides dynamic virtual hosts that are automatically created by honeyd and provide dynamic services.

Different machine learning, Intrusion detection has emerged as a significant field of research. To detect intrusion activities (ML) algorithms, Support Vector Machine , Genetic Algorithm ,Fuzzy Logic , and Data Mining have been extensively employed [6].In (2002) Lipson, this trend of Attack sophistication vs. intruder technical knowledge. Because of scenario of network attack, it became important for the researchers and operators to know about trends in network traffic.

III. PROPOSED SYSTEM

To provide security against cyber devices,proposed a new system as dynamic honeypot with deceptive virtual hosts and intrusion detection system. Proposed system consist of two major task

A) Honeyd configuration.

B) Intrusion detection.

This section will discuss these two steps

A. Honeyd configuration

To implement honeyd and virtual hosts [1] following steps are use:

a) Network entity identification (NEI);
b) DVH configuration;
c)

a) Network Entity Identification:

The NEI analyse network traffic from which it gives information about source, destination, OS identification, port identification. The NEI deliver this information to implementation part to create a DHP configuration. NEI provide the basic information used to create virtual hosts.

Create and update virtual hosts with following Pseudo code [1]:

```
Network   Entity Identification.
 Write entities to XML.
 Read_data from input files
 For each IP create a Dynamic Virtual Host
    Find_closest representative OS.
    Map_OS values to Honeyd names
    Create_MAC address for new hosts
   Create_Features for devices specific behaviours
   Create_Config for virtual hosts
End
```

DOI: 10.15680/ijircce.2015. 0307011

b) Dynamic Virtual Hosts

Honeyd is open source to create feature rich virtual hosts. Honeyd provide high flexibility to virtual hosts. Honeypot software can be installed to simulate one or more virtual honeypot systems on one host computer. Each virtual honeypot appears as a separate computer, router, or other networked device to the attacker. Most of the honeypot applications come with emulated TCP/IP stacks and services and even contain fake content and data. A Honeypot software offers built-in monitoring and logging mechanisms for easy administration. If any honeypot get compromised, it can simply reinstall the virtual honeypot from one cloned image to restore the host system.

"Niels Provos created the original Honeyd in 2002 as an open-source UNIX tool. Davis is responsible for the Windows port of the Snort IDS. With the exception of subsystems and a few small syntax changes [6], the Windows port of Honeyd is identical to its UNIX cousin."

Honeyd is a low-interaction honeypot having the basics such as OS, IP stack, and simple services. Low-interaction honeypot doesn't include forged content or running applications. A low interaction also means the honeypot is simple and easy to deploy and to recreate in the event of a compromised host. To implement Honeyd following are steps are used:

1. The Configuration File

   When Honeyd started, immediately it looks for a configuration file in which this system has defined virtual host templates, bound IP addresses to those templates, assigned OS personality, and defined ports and services.

2. Templates

Templates are used by Honeyd to track one or more virtual OS. Each template have a unique IP address associated with a virtual OS's personality, ports, and services of respective OS. One instance of Honeyd can support many templates. Each configuration file should contain a template called default, which is used when no other template applies. Create <template-name>

3. Emulated IP-Addresses

   Honeyd is configuring to use specific, predefined IP addresses or a range of IPs. Remaining options are to have Honeyd respond to any request for a currently invalid address or to any packet observed on the wire. The latter behaviour is the default behaviour. The subnet Honeyd will emulate as defined below,

```
  Honeyd 192.168.169.0
 Honeyd 192.168.169.1-192.168.169.255
```

To bind a Honeyd template to a specific IP address, the configuration file statemen

```
  bind   <IP address> <template name>
```

For example,

```
bind 192.169.169.202 win98
```

Network will configure to redirect the appropriate traffic to the virtual Honeyd IP address. Honeyd is hosted on a PC that uses one IP address and Honeyd should have a unique IP address and subnet. Static routes or Address Resolution Protocol (ARP) proxies are used as mandatory.

An important part of Honeyd is the concept of OS personalities Every OS responds somewhat differently to various TCP/IP requests. All packets that Honeyd outputs passes through the personality engine for inspection to make sure it matches the configured personality.

Honeyd have each template and each template have a different personality. Honeyd can configured to impersonate more than 130 systems with different versions of Windows, Linux, Mac OS, Sun Microsystems' Sun Solaris, Digital UNIX, FreeBSD, SCO Group's SCO and many network infrastructure devices. Honeyd have Honeyd spoof the personality of a Cisco Systems router. Honeyd create custom personalities to extend Honeyd's functionality.

Personalities use the fingerprinting database of the open-source Network Mapper tool and another fingerprinting utility called Xprobe2 by Fyodor Yarochkin. Nmap performs TCP/UDP fingerprinting, Where Xprobe2 uses Internet Control Message Protocol (ICMP) packets. Combine these utilities send different types of TCP/IP packets, find the results, and use the combination of results to identify a particular host OS.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015

Malicious users often use Nmap[9] or Xprobe2 to identify remote machines. Honeyd takes advantage of this reliance to spoof results back to the originating requestor. Thus this system can fool a remote attacker into believing he or she has broken into a host.

4. IP Ports

Honeyd supports number of ports and treats them in a meaningful way. In the configuration file, Honeyd configure Honeyd's default behaviour for all undefined port probes, also the values for specific ports tied to specific templates. Values include block, reset, open, cmd-string and proxy.

```
set     default     default     tcp     action     open
set     default     default     udp     action     reset
add     default     tcp         port    21         open
add     default     tcp         port    137        open
add     default     tcp         port    139        block
```

5. Emulated Services

As a low-interaction honeypot. Honeyd does not offer full OS or application emulation. Honeyd can use proxies and scripts to provide emulated services. Proxies are external host machines configured to accept redirects from Honeyd for predefined services.

A) Proposed Unsupervised Network Attack Detection

There are two knowledge based approaches.

1. Signature-based detection
2. Anomaly detection.

IDSs, IPSs, and firewalls uses signature based detection system. Signature based detection system can detect those attack for which training is given. Anomaly detection creates normal operation traffic profiles using labelled data. Anomaly detection approach requires training for profiling, because of this reason it becomes time consuming task. This paper concentrates on anomaly detection problem.

In this system there is the requirement of analysis technique which is not depending on knowledge that is knowledge independent technique. In order to discover Knowledge Independent system, proposed unsupervised network attack detection algorithm. The figure describing algorithm is as shown in below figure.

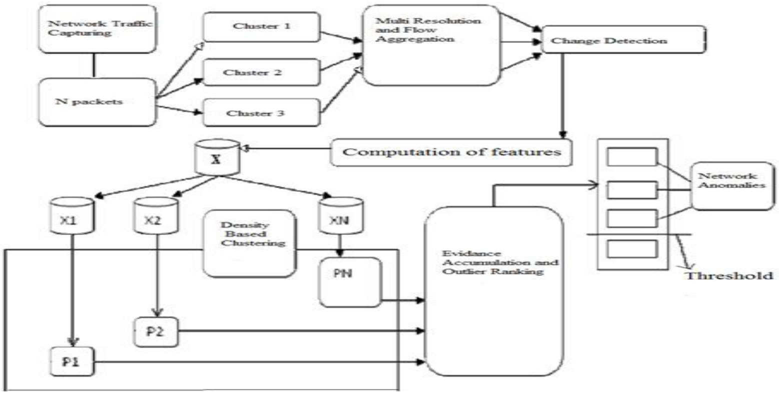


Fig. 2. Unsupervised network attack detection algorithm

B) Steps in proposed algorithm

1. Traffic Capturing

Initially traffic is captured and are analysed by aggregating them in multi resolution flow. On the top of these flow, various time series is built. Anomalous change is defined by time-series analysis.

2. Determining degree of abnormality

An unsupervised network attack detection used robust clustering algorithm like Sub-Space Clustering (SSC), Evidence Accumulation Clustering (EAC) and Density-based Clustering as combination of these approaches for providing traffic structure. Traffic structure is used as the evidence for determining normal and abnormal evidence.

3. Declaring anomalies

Simple threshold detection approach is used to outlying flow which are top ranked are flagged as anomalies.

An algorithm performs unsupervised anomaly detection. Anomalies are captured in consecutive time slot of fixed length, which are further aggregated in IP flows. To detect anomalous time slot, a time series is built based on metrics which include IP flows per time slot, number of bytes, packets. An aggregation key is used to accomplish this task. Change detection method is then used on time series, in such way that at arrival of every new time slot, change detection method analyses different time series with the help of each aggregation key.

C) *Unsupervised Attack Detection Using Clustering*

IP flows in the flagged time slot are used as the input for unsupervised attack detection. In first step unsupervised network attack detection algorithm ranks the degree of abnormality of every flow by clustering and outliers analysis techniques. This task is accomplished at two different resolutions, using either IP-source or IP-destination aggregation key IP flows are analysed. Two different anomalies exist which can be classified, 1-to-N anomalies and N-to-1 anomalies. In first case many IP flows are transferred from same source to different destination they are said to be 1-to-N anomalies, ex. worms or virus. Second, N-to1 means IP flows when transferred from different sources to one destination, ex. DDoS attacks.

1-to-N anomalies are highlighted by IP-source and N-to-1 anomalies are more easily detected with IP destination key. There are highly distributed anomalies, but the use of both key i.e, IP-destination key and IP-source key number of IP flows which can be represented as outliers.

Unsupervised network attack detection algorithm is based on clustering technique. Homogeneous groups of similar characteristics or clusters are formed by partitioning a set of unlabeled samples. The samples which do not belong to any of these cluster are outliers. Identifying the cluster properly is important to determine the outlier. Different partitions of data are produced using different clustering algorithms. Also different results are produce, even the same clustering algorithm are used by using different initialization parameters. Hence present clustering algorithms aren't robust. To remove this major drawback of robustness. This is done using multiple clustering combinations.

IV. RESULT

Step of proposed system working:

1. Configuration of three systems in LAN network.
2. Honeypot creates virtual hosts.
3. Honeypot response to client (hacker) request using fake address of virtual host.
4. Intrusion Detection system record detail of attacker.

A. *Configuration of three system in LAN network:*

Firstly configure three systems within LAN network. In this three systems first system will be server on which Tomcat Apache server is running, Honeypot on second system and third system is as client. Set IP address of first server system as 192.163.43.102. On server system have Tomcat Apache server and server system has one application which is run on Apache server. Application on server system running and attacker use fake address to access this application. This server application is shown as below.

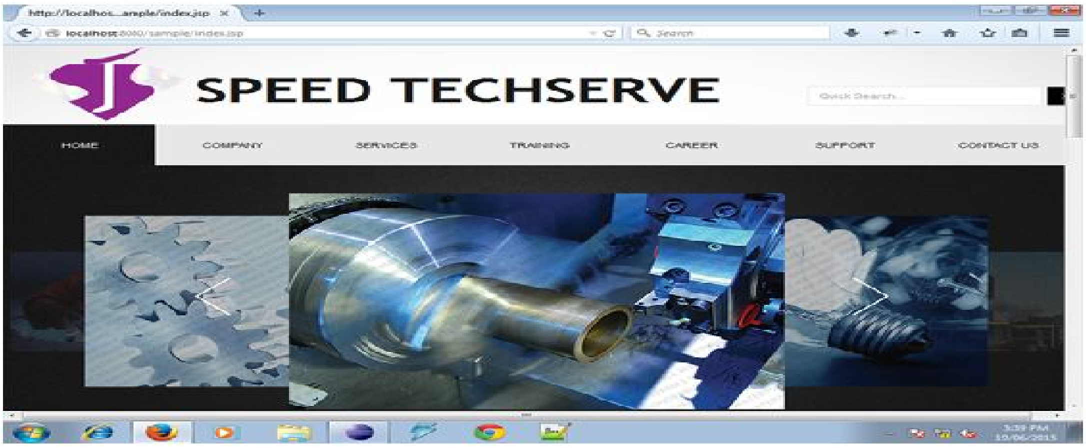
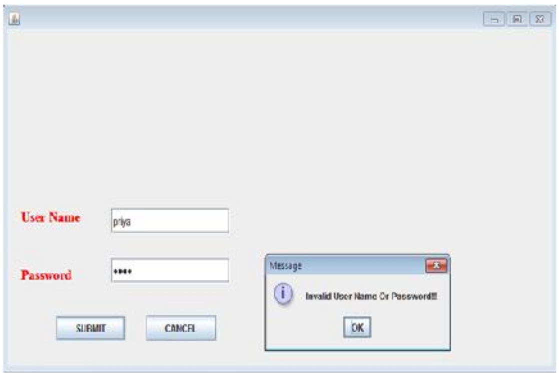


Fig. 3. Application on server system

Set IP address of second system on which honeypot is configure 192.168.43.103 and set IP address of client system as 192.168.43.161.

B. Configuration of honeypot and create virtual hosts:

After executing the program honeypot is configured and started after secure log in. Admin is user and password to start honeypot. If user is not as Admin then it will show invalid user which show as below figure (a) and after valid user and password honeypot system starts honeypot framework as shown in figure (b):



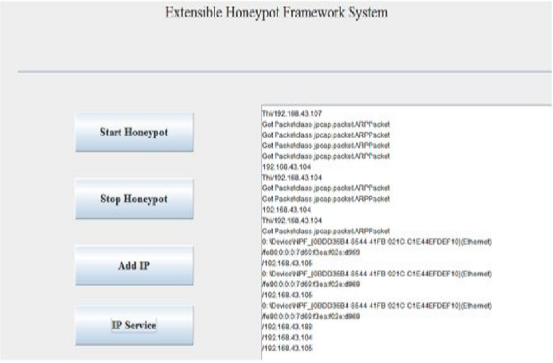
User Name: priya

Password: ****

Message

Invalid User Name Or Password!!!

Fig. 4. Login window

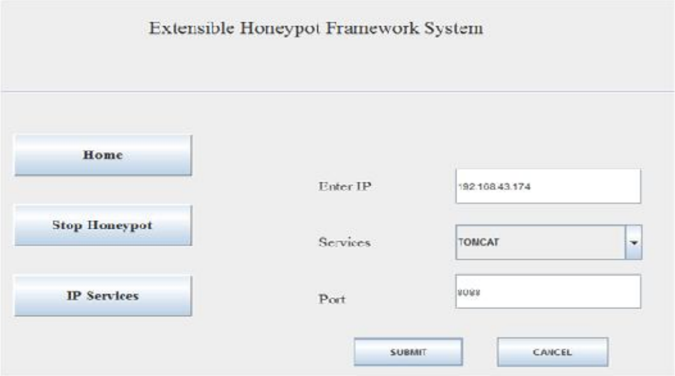


Extensible Honeypot Framework System

```
Thi/192.168.43.107
Got Packetclass jpcap.packet.ARPPacket
Got Packetclass jpcap.packet.ARPPacket
Got Packetclass jpcap.packet.ARPPacket
Got Packetclass jpcap.packet.ARPPacket
192.168.43.104
Thi/192.168.43.104
Got Packetclass jpcap.packet.ARPPacket
Got Packetclass jpcap.packet.ARPPacket
192.168.43.104
Thi/192.168.43.104
Got Packetclass jpcap.packet.ARPPacket
0: \Device\NPF_{0BDD35B4 8544 41FB 921C C1E44EFDEF10}(Ethernet)
/fe80:0:0:0:7d60:f3ea:f02e:d969
/192.168.43.105
0: \Device\NPF_{0BDD35B4 8544 41FB 921C C1E44EFDEF10}(Ethernet)
/fe80:0:0:0:7d60:f3ea:f02e:d969
/192.168.43.105
0: \Device\NPF_{0BDD35B4 8544 41FB 921C C1E44EFDEF10}(Ethernet)
/fe80:0:0:0:7d60:f3ea:f02e:d969
/192.168.43.189
/192.168.43.104
/192.168.43.105
```

Fig.5. Starting of honeypot framework

Honeypot create virtual host by adding IP in honeypot system as below. Suppose honeypot creating virtual host having IP address 192.168.43.174 and having port number 8088. After successfully creation of virtual host with IP address 192.168.43.174 and port 8088 shown as follows:



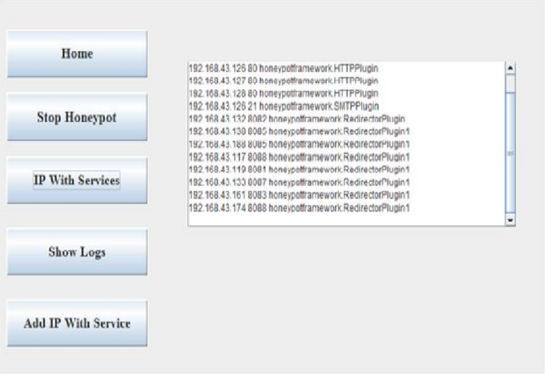
Extensible Honeypot Framework System

Enter IP: 192.168.43.174

Services: TONCAT

Port: 8088

Fig.6. Creation of virtual host



```
192.168.43.126 80 honeypotframework.HTTPPlugin
192.168.43.127 80 honeypotframework.HTTPPlugin
192.168.43.128 80 honeypotframework.HTTPPlugin
192.168.43.126 21 honeypotframework.SMTPPlugin
192.168.43.132 8082 honeypotframework.RedirectorPlugin
192.168.43.138 8085 honeypotframework.RedirectorPlugin1
192.168.43.188 8085 honeypotframework.RedirectorPlugin1
192.168.43.117 8088 honeypotframework.RedirectorPlugin1
192.168.43.119 8081 honeypotframework.RedirectorPlugin1
192.168.43.133 8087 honeypotframework.RedirectorPlugin1
192.168.43.161 8083 honeypotframework.RedirectorPlugin1
192.168.43.174 8088 honeypotframework.RedirectorPlugin1
```

Fig.7. Record of created virtual hosts

C. Honeypot response to client (hacker) request using fake address of virtual host:

Honeypot is now detect attacker and reply attacker client request using fake IP address and Port number. Intrusion detection system will be detect anomaly attacks from attacker and recorded in log file. Third system in LAN network act as client and send request to server system. Now honeypot check client is either legitimate user or real user. If client send request to server having real IP address of server, then this client is real client. If client send request to server using fake IP address then honeypot will check IP address coming from client within network if this IP address will not found within network ,then honeypot fix this client as attacker and reply to this client using fake IP address and port number.

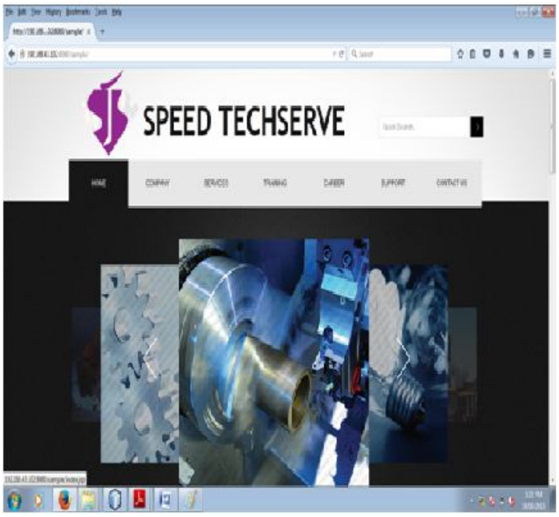


Fig.8. Real Client

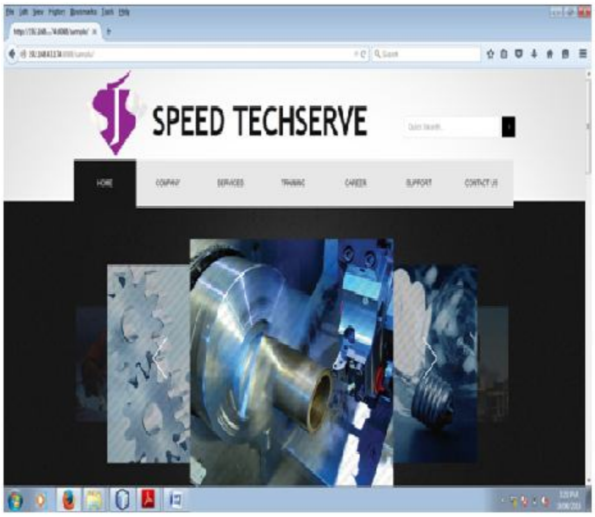


Fig. 9. Attacker client

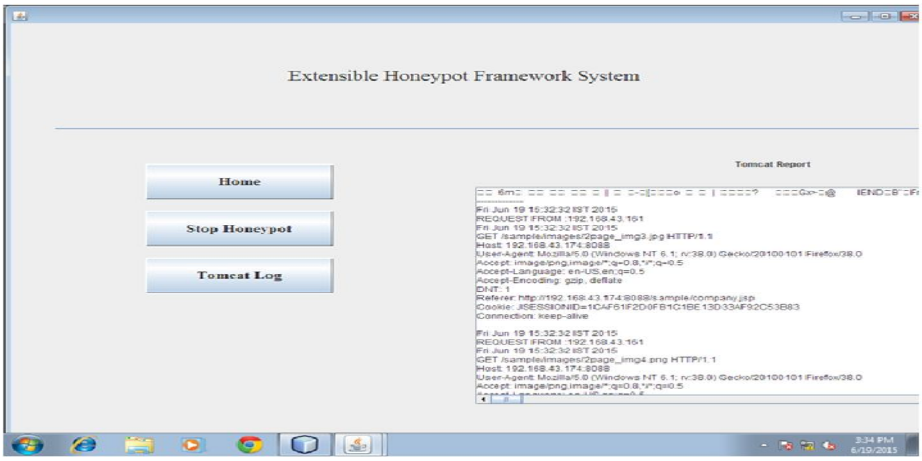
Honeypot will reply to attacker using fake address and port number as 192.168.43.174 and port 8088. As shown in below screen.

ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 3, Issue 7, July 2015



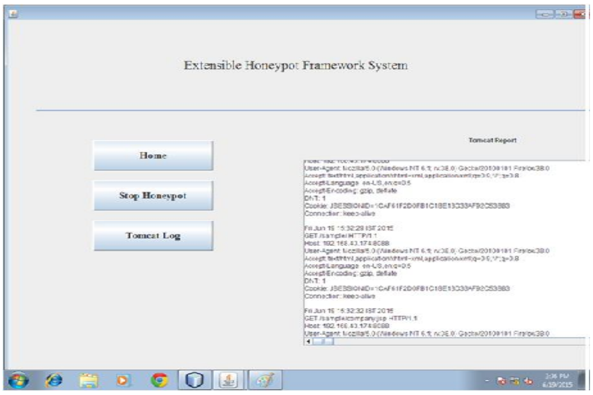
```
□□ 6m□ □□ □□ □□ □ ‖ □ □-□[□□□◦ □ □ | □□□□?   □□□G×-□@   IEND□B`□F
-------------
Fri Jun 19 15:32:32 IST 2015
REQUEST FROM :192.168.43.161
Fri Jun 19 15:32:32 IST 2015
GET /sample/images/2page_img3.jpg HTTP/1.1
Host: 192.168.43.174:8088
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:38.0) Gecko/20100101 Firefox/38.0
Accept: image/png,image/*;q=0.8,*/*;q=0.5
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Referer: http://192.168.43.174:8088/sample/company.jsp
Cookie: JSESSIONID=1CAF61F2D0FB1C1BE13D33AF92C53B83
Connection: keep-alive

Fri Jun 19 15:32:32 IST 2015
REQUEST FROM :192.168.43.161
Fri Jun 19 15:32:32 IST 2015
GET /sample/images/2page_img4.png HTTP/1.1
Host: 192.168.43.174:8088
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:38.0) Gecko/20100101 Firefox/38.0
Accept: image/png,image/*;q=0.8,*/*;q=0.5
```

Fig.10. Reply from honeypot to attacker

D. Intrusion Detection system record detail of attacker:

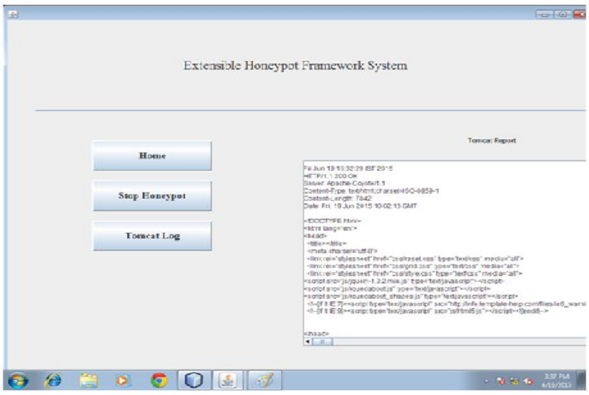
After detection of attacker, intrusion detection system capture anomaly packet coming from attacker and identify anomalous time slot from this anomaly packets. Extract suspicious flow from this attacker and produce filter rule on the basis of time series of attacker packets to from signature based attack. Intrusion detection system will detect intrusion and create log file which contain detail of attacker. Following windows shows records of attacker indicating time series of attackers request having time, date information.



```
Host: 192.168.43.174:8088
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:38.0) Gecko/20100101 Firefox/38.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Cookie: JSESSIONID=1CAF61F2D0FB1C1BE13D33AF92C53B83
Connection: keep-alive

Fri Jun 19 15:32:29 IST 2015
GET /sample/ HTTP/1.1
Host: 192.168.43.174:8088
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:38.0) Gecko/20100101 Firefox/38.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Cookie: JSESSIONID=1CAF61F2D0FB1C1BE13D33AF92C53B83
Connection: keep-alive

Fri Jun 19 15:32:32 IST 2015
GET /sample/company.jsp HTTP/1.1
Host: 192.168.43.174:8088
User-Agent: Mozilla/5.0 (Windows NT 6.1; rv:38.0) Gecko/20100101 Firefox/38.0
```



```
Fri Jun 19 15:32:29 IST 2015
HTTP/1.1 200 OK
Server: Apache-Coyote/1.1
Content-Type: text/html;charset=ISO-8859-1
Content-Length: 7842
Date: Fri, 19 Jun 2015 10:02:15 GMT

<!DOCTYPE html>
<html lang="en">
<head>
 <title></title>
 <meta charset="utf-8">
 <link rel="stylesheet" href="css/reset.css" type="text/css" media="all">
 <link rel="stylesheet" href="css/grid.css" type="text/css" media="all">
 <link rel="stylesheet" href="css/style.css" type="text/css" media="all">
<script src="js/jquery-1.3.2.min.js" type="text/javascript"></script>
<script src="js/[illegible].js" type="text/javascript"></script>
<script src="js/[illegible]_shapes.js" type="text/javascript"></script>
 <!--[if lt IE 7]><script type="text/javascript" src="http://info.template-help.com/files/ie6_warni
 <!--[if lt IE 9]><script type="text/javascript" src="js/html5.js"></script><![endif]-->


</head>
```

Fig.11.Anomalous Time slot

Fig.12. Tomcat log record attacker's contents

V. Conclusion

The proposed work presents the configuration of Honeyd on windows operating system to create multiple deceptive virtual hosts. The proposed work implements the clustering based approach to capture and detect the anomalies over the network. The proposed system contribution is to implement the unsupervised network anomaly detection with Honeyd.

REFERENCES

1. Todd Vollmer and Milos Manic,"Cyber-Physical System Security With Deceptive Virtual Hosts for Industrial Control Networks",IEEE Trans. Ind. Informat., VOL. 10, NO. 2, MAY 2014 .
2. Priya B.Bhosle and Manisha V. Desai,"Dynamic Honeypot Security With Deceptive Virtual Host",International Journal of Science and Research (IJSR),Volume 3 Issue 11, November 2014.

3. N. Provos and T. Holz,"Virtual Honeypots", Reading, MA, USA: Addison-Wesley, 2007.
4. J.HiebandJ. H.Graham,"Anomaly-based intrusion detection for network monitoring using a dynamic honeypot", Intell. Syst. Res. Lab., Univ. Louisville, Louisville, KY, TR-ISRL-0403, Dec. 2004.
5. Ram Kumar Singh and Prof. T. Ramanujam,"Intrusion Detection System Using Advanced Honeypots," Int. J. Comput. Sci. Eng., Vol. 2, No. 1, 2009.
6. Pragati H.,"Autonomous Network Security using Unsupervised Detection of Network Attacks",International Journal of Computer Science and Information Technologies, Vol.3(1),2012, 29922995.
7. A. Lakhina, M. Crovella, and C. Diot.,"Diagnosing Network-Wide Traffic Anomalies",In ACM Special Interest group on Data Communication, Portland, August 2004.
8. Gaia Maselli, Luca Deri, Stefano Suin,"Design and Implementation of an Anomaly Detection System: an Empirical Approach",University of Pisa.
9. Pedro Casas,Johan Mazel,"Steps Toward Autonomous Network Security :Unsuperised Detection of Network Attacks",IEEE International conference of communication , 2011.
10. Ntop Network Trafc Probe [Online]. Available: http://www.ntop.org.
11. V. Pothamsetty and M. Franz. SCADA Honeynet Project[Online].Available:http://scadahoneynet.sourceforge.net/.29 Dynamic Honeypot Security With Deceptive Virtual Host Exam No-6030.
12. Digital Bond Incorporated. SCADA Honeynet [Online]. Available: http:// www.digitalbond.com/tools/scada-honeynet.
13. M. Roesch,"Snort: Lightweight intrusion detection for networks",in Proc. 13th Conf. Syst". Admin., Berkeley, CA, USA, Nov. 712, 1999, pp. 229238.