Leveraging On Honeypot Technology To Improve Network Security

Ayeni, O. A, Alese, B. K, Dada, O. M Aliyu, E. O

Abstract

In this paper, a new framework is proposed for the design and implementation of decoy honeypot. In this research work, emphasis is laid on ways to improve network security by deploying honeypot technology. The design and configuration of a honeypot was implemented using a virtual machine(VM) ware workstation to detect attack or malicious traffic on a network. VM ware enables the creation, configuration, distribution, support and manages virtual machine similar to the one used on real computer system. Intrusion detection system (IDS), Entropy based detection scheme and Virtual machine (VM) ware work station were used to capture and analyse traffic over the network. The result shows that the deployment of an honeypot successfully fools an attacker to believe he is attacking a real system. Also, it shows that Honeypot can be deployed on a network to help in enhancing system security.

Keyword**: honeypot, traffic, network, detection & entropy.**

INTRODUCTION.

It is increasingly becoming difficult to secure computer networks due to largely increase in the activities of e-commerce over the internet. In recent times, a lot of losses have been recorded in term of cost and integrity of confidential data has been compromised due to the activities of hackers over the internet.

Today, information is a vital element in every aspect of life. Up-to-date and correct information are the key to any successful businesses, academia, government, personal finances or leisure activities. While this has been true for hundreds of years, it has never been as true as in the last half of the 20th century with the invention of the modern digital computer. Security is one of the hottest issues in network today. Worries about security have soared because of the increasing magnitude of electronic commerce occurring over the Internet and the swiftly evolving business trend towards telecommuting. Therefore, more sensitive and critical information is crossing the world than ever before. (Nor *et al*, 2006).[27].

The expansion of the World Wide Web has given unlimited access to attackers to prey on ignorant administrator who lacks basic knowledge of network security. Vulnerabilities in common security components such as firewalls, security patches, access control and encryption are inevitable, so hackers take advantage of these loopholes to break into computer networks. This paper presents the result of using a honeypot to limit the activities of hackers/attackers over computer networks. All traffic from and to a honeypot is considered to be unauthorised activity. Compromised honeypots are not threats to the security of the network as long as it is not high interaction honeypot but rather it aids us by collecting the data generated. All data collected by a honeypot is consequently interesting data. Data gathered by a honeypot is valuable and can lead to a better understanding and awareness which in turn can assist administrator in increasing overall network security.

Security of computing machines and networks are increasing in importance as more and more business is conducted via these systems. Despite decades of research and experience, we are still unable to make secure computer systems or even manufacture ways to measure a computer system’s level of security. The automation of exploit scripts and massive global scanning for vulnerabilities enable adversaries to compromise computer systems shortly after vulnerabilities become known. One way we can strengthen our defenses against these scans and exploit scripts is to learn from our adversaries by deploying and monitoring computer systems on a network that we expect to be broken into. These machines or systems we manufacture to be broken into are called *Honeypots*. When studying our adversaries we need to monitor and log every connection attempt to these machines and the known vulnerabilities present in our deployments.

1. Honeypot

The basic idea of a honeypot is quite old and has been in used already for quite a long time. Prior to honeypots, there was the seminal narrative by

Clifford Stoll of monitoring and tracking an intruder (Stoll, 1998).[32]. Stoll 1998 described how he created a complete but non existent government project with realistic but false files which attackers spent an extended period of time downloading and analysing, thus providing an opportunity to monitor and trace the attackers. The original Honeypot computer systems are documented in the two proceedings that are presented by Bellovin and Cheswick (Bellovin, 1992 and Cheswick, 1992) [7]. Although, the word "Honeypot" is a new phrase but the technology is not new and is getting more and more crucial. Possible definitions of what a honeypot is:

Spitzner, Lance (2003). [30] defines the term "Honeypot" as follows: A honeypot is a resource whose value is being attacked or compromised. This means, that a honeypot is expected to get probed, attacked and potentially exploited. Honeypots do not fix anything. They provide us with additional, valuable information.

2 Intrusion detection system

Intrusion detection, according to Kuwatly *et al* (2003) [17], is the process of monitoring computers or networks for unauthorized entrance or activity. Intrusion Detection System (IDS) can be used to monitor network traffic, thereby detecting if a system is being targeted by a network attack. There are two basic types of intrusion detection: host-based (HIDS) and network based (NIDS). Each has a distinct approach to monitoring and securing data, and each has distinct advantages and disadvantages. Host-based IDSs examine data held on individual computers that serve as hosts; they are highly effective for detecting insider abuses. Examples of HIDS implementations include Windows NT/2000 Security Event Logs, and UNIX Syslog. On the other hand, Network-based intrusion detection systems analyze data packets that travel over the actual network. These packets are examined and sometimes compared with empirical data to verify whether they are of malicious or benign nature. An example of NIDS is Snort, which is an open source software tool that can capture real-time network traffic. It can be configured to detect a variety of attacks and probes, such as buffer overflows, stealth port scans, and OS fingerprinting attempts.

A simple virtual network of three systems will be design to implement the concept of the system. The first system on the network served as a gateway (point of presence) to the other two systems, it is the responsibility of this system to redirect traffic flow to either the virtual server or the honeypot. This system will be connected to the internet to allow interaction with the network. The second system will serve as a virtual server with five hundred ports opened to imitate application services available and running on the system. The purpose of opening several port is to make clients and attackers believe there are real and important services running on the system. The third system will be the honeypot system which will be used to performs analysis on the attack traffic flow arriving into the network.

The configuration of the POP server include: 2000MB of memory, 15GB hard disc (SCSI), CD-ROM (IDE), single processor (Pentium(R)), Windows Server 2003 Enterprise edition. A software called snort will be installed to extract packet data information from traffic flow and perform entropy test to determine if a traffic flow is an attack or legitimate.

The requirement for the design of the medium-level dynamic honeypot system include: An operating system such as Windows 2003 Professional, with a 1GHz processor, 512 Mb of RAM, with a 10/100 network card already and a CDROM or DVD/RW drive. Windows 2003 Professional was the best choice to since it can be secured the most from the operating systems, other operating system that can be use include: Windows XP, Windows 2000 Server and Windows 2003 Professional. A program called Snort will be installed in the system. This program is an open source network intrusion prevention and detection system utilizing a rule-driven language, which combines the benefits of signature, protocol and anomaly based inspection methods. Snort is the most widely deployed intrusion detection and prevention technology worldwide and has become the de facto standard for the industry. (http://www.snort.org/) Snort is a free program which is extremely powerful in what it does. This is part of an intrusion detection system. Honeypot works by opening over 1000 user datagram protocol (udp) and transmission control protocol (tcp) listening sockets on the computer and these sockets are designed to mimic vulnerable services. When an attacker connects to these services they are fooled into thinking they are attacking a real server. The honeypot safely captures all communications with the attacker and logs these results for future analysis.

3. PROPOSED SYSTEM

The proposed framework provides for proactive mitigation against the effects of DoS attack. A system designed to redirect network traffic flow is positioned at the network gateway as point of presence (POP). All the traffic flows arriving at the Point of Presence (POP) of a destination network server to be protected from DoS attack are tagged as either legitimate or attack. Whenever a packet belonging to suspicious flow arrives at the POP, instead of sending that packet to the active FTP server or dropping it, it is redirected to honeypot server. This provides a proactive approach to mitigation against the attack because the FTP server is isolated from attack traffic and bandwidth of the links with FTP server will not be exhausted by the voluminous attack traffic.

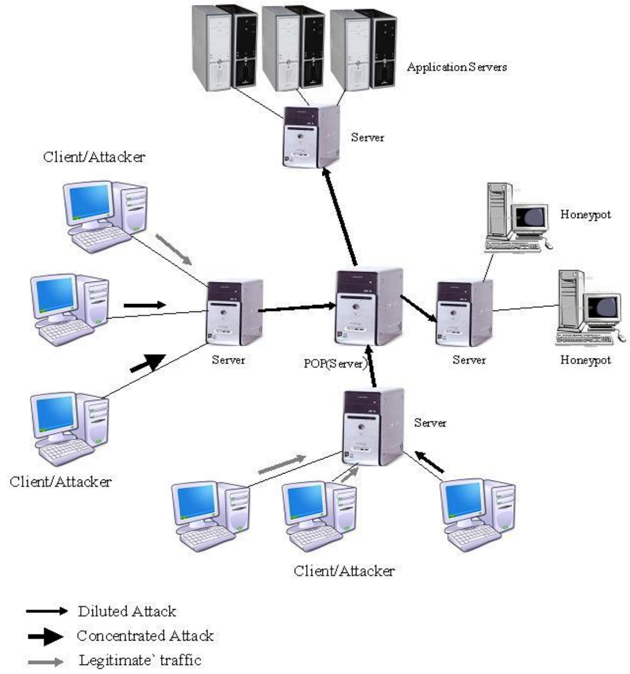


Fig.1 Proposed system.

The topology is similar to the one used to depict a typical client-server scenario in the Internet. The clients (attack and legitimate) send their FTP requests to the server. The arrows in the figure indicate the presence of variable rate attacks coming from client domain

3.1 Entropy based detection

The system detect and characterize attacks treats DoS anomalies as events that disturb the distribution of traffic features. For example, a DoS attack, regardless of its volume, will cause the distribution of destination address to be concentrated on the victim address or server. As proposed by Sardana et al (2008), entropy scheme will capture the degree of dispersal or concentration of a distribution flow. The sample entropy $H(X)$ is

$$H(X) = -\sum_{i=1}^{N}(P_i)\,x\,log_2(P_i)$$

where P_i = N_i/S, N is a set of positive integer that represents total number of system (server) on the network, n_i represents a flow of traffic at i. The value of sample entropy lies in the range range $0\ through\ log_2 N$. The metric takes on the value 0 when the distribution is maximally concentrated, *i.e.*, all observations are the same. Sample entropy takes on the value $log_2 N$ when the distribution is maximally dispersed. Let *NS* and *NH* represent the number of servers and honeypots respectively. A variable *lengvec is* defined such that *lengvec = NS +NH.* An array *vector []* of size *lengvec* is also defined whose elements are in the form of ordered pair set of destination IP address and port number of the honeypot or the server i.e. *vector[i] = {dest IP, port}* . Let *S* represent the set of indexes of the *vector[]* array. Further, two arrays *subvecNS []* and *subvecNH []* is also define whose elements are indices of the array *vector[]* that correspond to destination IP address and port number of servers and honeypots respectively such that the following holds true:

(lengvec=(Length(subvecNS)+Length(subvecNH))AND
(subvecNS ⋂ *subvecNH)*

Algorithm

The redirection algorithm performs the per-flow treatment of each flow in the Flow List (FL) in a time window at POP.

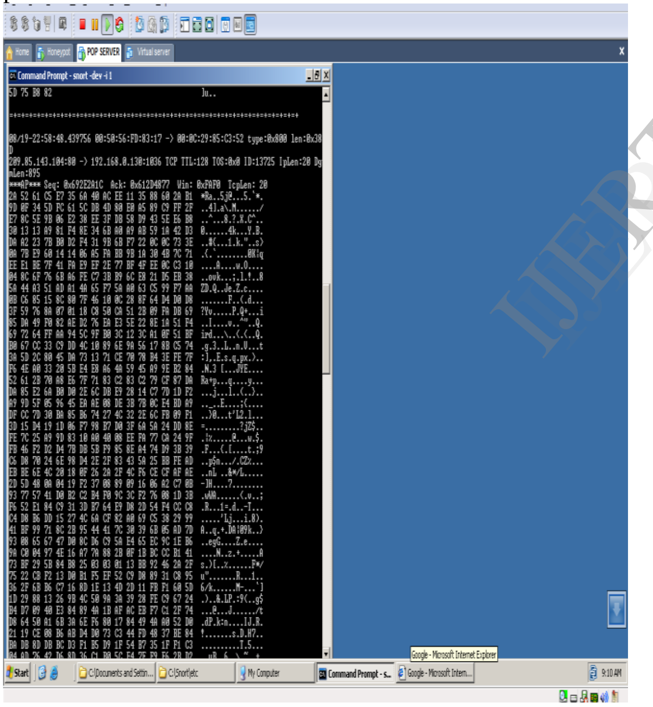
FDA – Flow Destination Address
FSA – Flow Source Address
NDA – Network Destination Address
PDA – Packet Destination Address
The pseudo code is as follows:

```
Honeypot Controller PerFlow (FL)
For a flow in FL
If (flow Tag = attack)
Parse the primary packet and search source and destination address (FDA and FSA)
            PDA = FDA
  NDA = PDA
A: If (NDA = Destination address of honeypot)
            Forward the packet to NDA
Else
            Replace NDA by destination address of honeypot
 Forward the packet to NDA
If (More Fragment = 0)
  Goto S
Else
            Parse next header of the flow for PDA
            NDA = PDA
If (Tag = attack)
             Goto A
Else
            Goto B
Else
 Parse the primary packet and search source and destination address (FDA and FSA)
  PDA = FDA
  NDA = PDA
B: If (NDA = Destination address of active FTP server)
  Forward the packet to NDA
```

```
Else
        Replace NDA by destination address of server
  Forward the packet to NDA
If (More Fragment = 0)
  Goto S
Else
  parse next header of the flow for PDA
  NDA = PDA
If (Tag = attack)
  Goto A
Else
        Goto B
S: Stop
```

POP Server Sniffer and Detection Mode

The POP server which direct network traffic to the other two systems is configured and executed in sniffer mode to extract network traffic details from incoming network connection. If the packet extracted from the connection is malicious, it is directed to the honeypot otherwise it directs the packet to the virtual server.


Home
Honeypot
POP SERVER
Virtual server
Command Prompt - snort -dev -i 1
Start
C:\Documents and Settin...
C:\Snort\etc
My Computer
Command Prompt - s...
Google - Microsoft Intern...
Google - Microsoft Internet Explorer
9:10 AM

```
5D 75 B8 82                                      ]u..

=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+

08/19-22:58:48.439756 00:50:56:FD:83:17 -> 00:0C:29:85:C3:52 type:0x800 len:0x38
D
209.85.143.104:80 -> 192.168.0.130:1036 TCP TTL:128 TOS:0x0 ID:13725 IpLen:20 Dg
mLen:895
***AP*** Seq: 0x692E2A1C  Ack: 0x612D4877  Win: 0xFAF0  TcpLen: 20
2A 52 61 C5 E7 35 6A 40 AC EE 11 35 88 60 2A B1  *Ra..5j@...5.`*.
9D 0F 34 5D FC 61 5C DB 4D 80 E0 A5 89 C9 FF 2F  ..4].a\.M....../
E7 8C 5E 9B 06 E2 38 EE 3F DB 58 D9 43 5E E6 B8  ..^...8.?.X.C^..
30 13 13 A9 81 F4 8E 34 6B A0 A9 AB 59 1A 42 D3  0......4k...Y.B.
DA A2 23 7B B0 D2 F4 31 9B 6B F7 22 0C 0C 73 3E  ..#{...1.k."..s>
0A 7B E9 60 14 14 06 A5 FA BB 9B 1A 30 4B 7C 71  .{.`........0K|q
EE E1 BE 7F 41 FA E9 EF 2E 77 BF 4F EE 0C C3 10  ....A....w.O....
04 8C 6F 76 6B A6 FE C7 3B B9 6C E8 21 D5 EB 38  ..ovk...;.l.!..8
5A 44 A3 51 AD A1 4A 65 F7 5A A0 63 C5 99 F7 AA  ZD.Q..Je.Z.c....
0B C6 85 15 8C 80 7F 46 10 0C 28 8F 64 D4 D0 D8  .......F..(.d...
3F 59 76 8A 07 01 18 C8 50 CA 51 2B 09 FA DB 69  ?Yv.....P.Q+...i
85 DA 49 F0 82 AE D2 76 EA E3 5E 22 8E 1A 51 F4  ..I....v..^"..Q.
69 72 64 FF AA 94 5C 9F B0 3C 12 3C A1 0F 51 BF  ird...\..<.<..Q.
B0 67 CC 33 C9 DD 4C 10 89 6E 9A 56 17 8B C5 74  .g.3..L..n.V...t
3A 5D 2C 80 45 DA 73 13 71 CE 70 78 B4 3E FE 7F  :],.E.s.q.px.>..
F6 4E A0 33 20 5B E4 E8 A6 4A 59 45 A9 9E B2 84  .N.3 [...JYE....
52 61 2B 70 A8 E6 7F 71 83 C2 83 C2 79 CF 87 DA  Ra+p...q....y...
DA 85 E2 6A B0 D0 2E 6C DB E9 28 14 C7 7D 1D F2  ...j...l..(..}..
A9 9D 5F 05 96 45 EA AE 08 DE 3B 7B 0C E4 BD A9  .._..E....;{....
DF CC 7D 30 BA 85 B6 74 27 4C 32 2E 6C FB 09 F1  ..}0...t'L2.l...
3D 15 D4 19 1D 06 F7 98 B7 D0 3F 6A 5A 24 DD 8E  =.........?jZ$..
FE 7C 25 A9 9D 83 10 A0 40 08 EE FA 77 CA 24 9F  .|%.....@...w.$.
FB 46 F2 D2 D4 7B DB 5B F9 85 8E A4 74 D9 3B 39  .F...{.[....t.;9
C6 D8 70 24 6E 98 D4 2E 2F 83 43 5A 25 BB FE AD  ..p$n.../.CZ%...
EB BE 6E 4C 20 18 0F 26 2A 2F 4C F6 CE CF AF AE  ..nL ..&*/L.....
2D 5D 48 0A 04 19 F2 37 08 89 09 16 06 A2 C7 0B  -]H....7........
93 77 57 41 D0 B2 C2 B4 F0 9C 3C F2 76 08 1D 3B  .wWA......<.v..;
F6 52 E1 84 C9 31 3D B7 64 E9 D8 2D 54 F4 CC C8  .R...1=.d..-T...
C4 D8 B6 DD 15 27 4C 6A CF 82 A0 69 C5 38 29 99  .....'Lj...i.8).
41 BF 99 71 8C 2B 95 44 41 7C 30 39 6B 05 AD 7D  A..q.+.DA|09k..}
93 08 65 67 47 D0 8C D6 C9 5A E4 65 EC 9C 1E B6  ..egG....Z.e....
9A C0 04 97 4E 16 A7 7A 88 2B 0F 1B BC CC B1 41  ....N..z.+.....A
73 BF 29 5B 84 B8 25 03 03 01 13 BB 92 46 2A 2F  s.)[..%......F*/
75 22 CB F2 13 D0 B1 F5 EF 52 C9 D8 89 31 C8 95  u".......R...1..
36 2F 6B B6 C7 16 8D 1E 13 4D 2D 11 FB F1 60 5D  6/k......M-...`]
1D 29 88 13 26 9B 4C 50 9A 3A 39 28 FE C9 67 24  .)..&.LP.:9(..g$
B4 D7 09 40 E3 84 89 4A 1B AF AC EB F7 C1 2F 74  ...@...J....../t
D8 64 50 A1 6B 3A 6E F6 80 17 84 49 4A A0 52 D0  .dP.k:n....IJ.R.
21 19 CE 08 B6 AB D4 D0 73 C3 44 FD 48 37 BE 84  !.......s.D.H7..
BA DB 8D DB BC D3 F1 B5 D9 1F 54 B7 35 1F F1 C3  ..........T.5...
```

Fig.2 Network packet at PoP server

Result

The following result was obtained in a log file after about two hour of connecting the virtual network to the internet.

The data below shows the sample result in log file from the network packet at POP server:

```
00000000  4d 45 53 47 00 00 00 64  4e 41 4d 45 00 00 00 10  |MESG...dNAME....|
00000010  42 6c 6f 67 20 49 6e 66  6f 72 6d 61 74 69 6f 6e  |Blog Information|
00000020  44 41 54 41 00 00 00 1f  42 65 20 73 75 72 65 20  |DATA....Be sure |
00000030  74 6f 20 63 68 65 63 6b  20 6f 75 74 20 6f 75 72  |to check out our|
00000040  20 62 6c 6f 67 20 61 74  20 68 74 74 70 3a 2f 2f  | blog at http://|
00000050  76 72 74 2d 73 6f 75 72  63 65 66 69 72 65 2e 62  |vrt-sourcefire.b|
00000060  6c 6f 67 73 70 6f 74 2e  63 6f 6d 2f  |logspot.com/|
```

```
Protocol: UDP
   Source: 192.168.100.161 (192.168.100.161)
   Destination: 192.168.0.132 (192.168.0.132)
User Datagram Protocol, Src Port: 54296 (54296),
Data (991 bytes)

0000  3c 3f 78 6d 6c 20 76 65 72 73 69 6f 6e 3d 22 31   <?xml version="1
0010  2e 30 22 20 65 6e 63 6f 64 69 6e 67 3d 22 75 74   .0" encoding="ut
0020  66 2d 38 22 20 3f 3e 0a 3c 73 6f 61 70 3a 45 6e   f-8" ?>.<soap:En
0030  76 65 6c 6f 70 65 20 78 6d 6c 6e 73 3a 73 6f 61   velope xmlns:soa
0040  70 3d 22 68 74 74 70 3a 2f 2f 77 77 77 2e 77 33   p="http://www.w3
0050  2e 6f 72 67 2f 32 30 30 33 2f 30 35 2f 73 6f 61   .org/2003/05/soa
0060  70 2d 65 6e 76 65 6c 6f 70 65 22 20 78 6d 6c 6e   p-envelope" xmln
0070  73 3a 77 73 61 3d 22 68 74 74 70 3a 2f 2f 73 63   s:wsa="http://sc
0080  68 65 6d 61 73 2e 78 6d 6c 73 6f 61 70 2e 6f 72   hemas.xmlsoap.or
0090  67 2f 77 73 2f 32 30 30 34 2f 30 38 2f 61 64 64   g/ws/2004/08/add
```

The data below shows the result from the network packet from honeypot system

```
08/21-02:55:40.702060 [**] [1:1000002:1] extracting packet data from network connection attk[**] [Priority: 0] {TCP} 192.168.0.130:1048 -> 192.168.0.129:1048
08/21-02:55:41.203766 [**] [1:1000002:1] extracting packet data from network connection attk[**] [Priority: 0] {TCP} 209.85.143.104:80 -> 192.168.0.130:1048
08/21-02:55:41.626241 [**] [1:1000002:1] extracting packet data from network connection attk[**] [Priority: 0] {TCP} 192.168.0.130:1049 -> 192.168.0.132:80
08/21-02:55:42.141453 [**] [1:1000002:1] extracting packet data from network connection attk[**] [Priority: 0] {UDP} 209.85.143.99:80 -> 192.168.0.130:1049
```

```
08/21-02:55:42.410615        [**]    [1:1000002:1]
extracting packet data from network connection
attk[**] [Priority: 0] {TCP} 209.85.143.99:80 ->
192.168.0.130:1049
08/21-02:55:42.438233        [**]    [1:1000002:1]
extracting packet data from network connection
attk[**] [Priority: 0] {UDP} 192.168.0.130:1049 -
> 192.168.0.132:80
08/21-02:55:42.620976        [**]    [1:1000002:1]
extracting packet data from network connection
attk[**] [Priority: 0] {TCP} 192.168.0.130:1050 ->
192.168.0.132:80

08/21-04:55:40.702060        [**]    [1:1000002:1]
extracting packet data from network connection
attk[**] [Priority: 0] {TCP} 192.168.0.130:100 ->
192.168.0.132:100
08/21-04:55:41.830478        [**]    [1:1000002:1]
extracting packet data from network connection
attk[**] [Priority: 0] {TCP} 209.85.143.104:80 ->
192.168.0.130:1048
08/21-04:55:41.626241        [**]    [1:1000002:1]
extracting packet data from network connection
attk[**] [Priority: 0] {TCP} 192.168.0.130:1049 ->
192.168.0.132:80
08/21-04:55:42.141453        [**]    [1:1000002:1]
extracting packet data from network connection
attk[**] [Priority: 0] {TCP} 209.85.143.99:80 ->
192.168.0.132:1049
08/21-04:55:42.410615        [**]    [1:1000002:1]
extracting packet data from network connection
attk[**] [Priority: 0] {TCP} 209.85.143.99:80 ->
192.168.0.132:1049
```

Sample of data obtained from POP logfile

```
Num  Protocol  Category  Port
=== =========== ========  ======
1        Tcp        attk      95
2        Tcp        attk      130
3        Tcp        attk      497
4        Tcp        attk      697
5        Tcp        norm      80
6        Tcp        attk      497
7        Tcp        norm      85
8        Udp        attk      130
9        Tcp        attk      130
10       Tcp        attk      100
11       Tcp        norm      100
12       Udp        attk      80
13       Udp        attk      123
14       Udp        attk      123
15       Tcp        attk       123
16       Tcp        attk       258
17       Tcp        attk        85
18       Tcp        attk        75
```

Network connection summary from logfile

Incoming connection at POP	Honeypot traffic	Virtual server request

323	241
	TCP= 158
	UDP=83
		82

Table 2. Network connection summary from logfile.

4. CONCLUSION

In this research, a medium-interaction honepot was designed. A simple virtual network of three systems was designed to implement the concept of the system. This system was connected to the internet to allow interaction with the network. The design and configuration of this honeypot was implemented using a virtual machine(VM ware) workstation to detect attack or malicious traffic on a network. The Point of Presence (POP) server serve as a link to the two other systems; honeypot system and the application server with virtual application running in it to give the impression of the presence of useful resources. An intrusion detection software called Snort were installed on each of the component of the honeypot to extract packet data information from traffic flow and perform entropy test on them to determine if a traffic flow is an attack or legitimate. The framework is aware of two internet protocols: TCP and UDP. Packets for other protocols are logged and silently discarded. At the end, malicious traffic were discovered and forwarded to honeypot for further analysis.

REFERENCES

[1] A. Lakhina, M. Crovella, and C. Diot, "*Mining Anomalies Using Traffic Feature Distributions*," ACM SIGCOMM, 2005.

[2] A. Sardana and R. C. Joshi, "Simulation of Dynamic Honeypot Based Redirection to Counter Service level DDoS Attacks". In *Proceedings of ICISS 2007, Springer LNCS 4812*, pp. 259–262, 2007.

[3]. Anjali Sardana, Krishan Kumar and R. C. Joshi, "Detection and Honeypot Based Redirection to Counter DDoS Attacks in ISP Domain" *In Proceedings of IEEE Third International Symposium on Information Assurance and Security.* Manchester, UK, pp. 191-196, *Aug* 2007

[4]. Anjali Sardana and R. C. Joshi1 *An Integrated Honeypot Framework for Proactive Detection, Characterization and Redirection of DDoS Attacks at ISP level*, 2008

[5] S. Bellovin (1992). "There be dragons". *Proceedings of the Third Usenix Security Symposium*, Baltimore MD.

[6]. B. Stephan, "Optimal filtering for denial of service mitigation," In *Proceedings of the 41st IEEE Conference on Decision and Control*, 2002, Vol. 2, pp. 1428 – 1433, Dec. 2002

CERT Statistics http: //www.cert.org/starts/cert

[7]. B. Cheswick (1992). An evening with Berferd in which a cracker is lured, endured, and studied. *Proceedings of the Winter USENIX Conference*, San Francisco

[8]. Christian Döring: Improving network security with Honeypots, pages B34-B36, Master's thesis University of Applied Sciences Darmstadt, Department of Informatics, 2005

[9]. Clifford Stoll: The Coocoo's egg, Pocket Books 1990

[10]. C.M. Cheng, H.T. Kung, and K.S. Tan, "Use of spectral analysis in defense against DoS attacks". In *Proceedings of IEEE GLOBECOM 2002*, pp. 2143-2148, 2002.

[11]. Eric Peter, epeter(at)wustl(dot)edu and Todd Schiller, tschiller(at)acm(dot)org (A project report written under the guidance of Prof Raj Jain). A practical guide to honeypot.

[12]. [ForeScout02] ForeScout Technologies. "Beyond Detection: Neutralizing Attacks Before They Reach the Firewall". Summer 2002

[13]. [ForeScout04] ForeScout Technologies, Inc. January 2004. [http://www.forescout.com/]

[14]. J. Mirkovic, J. Martin ,and P. Reiher, "A Taxonomy of DDoS Attacks and DDoS Defense Mechanisms". Technical Report 020018, Computer Science Department, University of California, LosAngeles,2002.

[15]. Johansson, Karsten. "Offensive Operations Model". KSAJInc.August2001. [http://www.penetrationtest.com/]

[16]. K.J. Houle, G. M. Weaver, N. Long, and R. Thomas. "Trends in denial of service attack technology". Technical Report Version 1.0, CERT Coordination Center, Carnegie Mellon University, 2001.

[17]. K.J. Ioannidis, and S. M. Bellovin, "Implementing Pushback: Router-Based Defense against DDoS Attacks". IEEE INFOCOMM, 2003.

[18]. L. Spitzner: "Honeypots, Tracking Hackers", pages 239-240, Addison-Wesley 2002

[19]. M. Roesch: Martin Roesch, Snort – Intrusion Detection and Prevention System, http://www.snort.org/, Sourcefire Inc.

[20]. M. Roesch, "Snort—Lightweight Intrusion Detection for Networks". In *Proceedings of USENIX Systems Administration Conf. (LISA '99)*, Nov. 1999.

[21]. National Institute of Standards and Technology (NIST)"Guidelines on firewalls and firewall policy" January 2002

[22]. Nor Badrul Anuar, Omar Zakaria, and Chong Wei Yao University of Malaya, Kuala Lumpur MY Honeypot through Web: The emerging ofsecurity application integration.

[23]. P. Dewan, P. Dasgupta, and V. Karamcheti. "Defending against Denial of Service attacks using Secure Name resolution", 2003

[24]. [PPC97] Pfleeger, P. Charles. "Security in Computing". Prentice Hall PTR. Second Edition. p 3. 1997

[25]. Rafeeq Ur Rehman, 2003. Advanced IDS Techniques Using Snort, Apache, MySQL, PHP, and ACID

[26]. R. B. Blazek, H. Kim, B. Rozovskii, and A. Tartakovsky, "A novel approach to detection of denial of-service attacks via adaptive sequential and batch sequential change-point detection methods". In *Proceedings of IEEE Systems, Man and Cybernetics Information Assurance Workshop*, 2001

[27]. Saleh Ibrahim Bakr Almotairi. Using honeypots to analyse Anomalous internet activities.

[28]. Spitzner, Lance. Honeypots: Tracking Hackers. Addison-Wesley Professional,
2002. *An older book providing a comprehensive discussion of honeypots. Includes an in-depth treatment of 6 available honeypots*

[29]. Spitzner, L. Honeypots – Tracking Hackers. Addison-Wesley, 2003.ISBN 0-321-10895-7.

[30]. Spitzner, L. Honeypots: Are they illegal? Security Focus, Infocus (June 2003). http://www.securityfocus.com/infocus/1703.

[31]. Spitzner, L. Honeypots: Catching the insider threat. In proceedings of the 19th Annual Computer Security Applications Conference (ACSAC) (2003), pp. 170–179.

[32]. Stoll, C. (1998). Stalking the wiley hacker. *Comunications of the ACM*, *31*(5), 484-497

[33]. The Honeynet Project: Research alliance, http://www.honeynet.org,non-profit Honeypot research organization, 1999

[34]. The Honeynet Project. Know Your Enemy: Sebek - a kernel based data capture tool, Nov. 2003.http://www.honeynet.org/papers/ sebek.pdf

[35]. Y. Xiang and W. Zhou. "Classifying DDoS packets in high-speed networks", In International Journal of Computer Science and Network Security, Vol. 6, No. 2B, February 2006

[36].[BaS].Bait and Switch Honeypot. http://baitnswitch. sourceforge.net/.

[37]. [CVE] CVE – Common Vulnerabilities and Exposures. http://cve.mitre.org.

[38]. [HNET] The HoneyNet Project. http://www.honeynet.org.

[39]. [LOBSTER] LOBSTER-Large-scale Monitoring of Broadband Internet Infrastructures. http://www.ist-lobster.org/.

[40]. [SEBEK] Sebek - A data capture tool. http://www.honeynet.org/tools/sebek.

[41]. [SNORT] Snort - A network intrusion detection system. http://www.snort.org.

[42].A virtual Honeypot framework, Niels provos Google, Inc. niel@google.com