ISSN No. 0976-5697

Volume 4, No. 8, May-June 2013

International Journal of Advanced Research in Computer Science

RESEARCH PAPER

Available Online at www.ijarcs.info

Emulation of Snort on Deter Testbed

Luvpreet Kaur
Computer Science Department
GIMET Amritsar, India
Er.luvpreetkaur@gmail.com

Monika Saluja
Computer Science Department
SBSSTC Ferozepur, India
Monika.sal@rediffmail.com

Krishan Saluja
Computer Science Department
SBSSTC Ferozepur, India
k.saluja@rediffmail.com

Abstract: In the present era Internet has changed the normal way of performing the essential services such as banking, defense, trade, and marketing being operated. These operations of our day to day life are replaced by cheaper, more effect Internet based applications. It happens all because of the growth of the internet but with advancement the number of attacks on these services is increasing day by day.

During internet design the more emphasis is on its functionality rather than the security due to which so many loop holes are generated in the internet. The possible way to detect those loop holes and then protects the system from the attack. The Intrusion Detection System is one type of guard which protect our network from attacking the by detecting the reason or way by which attack is performed.

To configure the Intrusion Detection system we required a system and to test an Intrusion Detection System we required a complete setup lab or a network of computers. DETER EMULAB is one kind of lab based on emulation which provides us facility for performing the experiment related to the network security.

In this paper we present our ongoing work of deployment and operation of an Intrusion Detection System Snort (which sniffs the packets and check the possibility of attack on our system and alert us regarding the occurrence of attack) on DETER EMULAB node.

Keywords: Snort; Deter; IDS ; Defense;.Mysql

I. INTRODUCTION

Intrusions have been biggest problems for internet security for almost a decade. During that time, attacks have taken birth from naïve and limited to sophisticated and large-scale, and many defenses have been proposed. While much time spend on theoretical or simulator based solution, little has been done to design sound, realistic environment and tool for detection of these intrusion. The underlying reason is not the researchers' neglect, but due the complexity of the Intrusion phenomenon. In reality, the researcher should spend more time in test setup and design than in solution development.

Intrusion detection has been an active field of research for about two decades[1], starting in 1980 with the publication of John Anderson's Computer Security Threat Monitoring and Surveillance,[2] which was one of the earliest papers in the field. Dorothy Denning's seminal paper, "An Intrusion Detection Model," [3] published in 1987, provided a methodological framework that inspired many researchers still, despite substantial research and commercial investments, ID technology is immature and its effectiveness is limited. [4] Within its limitations, it is useful as one portion of a defensive posture, but should not be relied upon as a sole means of protection.

With the development of network technologies and applications, network attacks are greatly increasing both in number and severity. As a key technique in network security domain, Intrusion Detection System (IDS) [5] plays vital role of detecting various kinds of attacks and secures the networks. Main purpose of IDS is to find out intrusions among normal audit data and this can be considered as classification problem. Intrusion detection systems (IDS) are an effective security technology, which can detect, prevent and possibly react to the attack. It performs monitoring of target sources of activities, such as audit and network traffic data in computer or network systems, requiring security measures, and employs various techniques for providing security services. With the tremendous growth of network-based services and sensitive information on networks, network security is becoming more and more important than ever before.

Symantec in a recent report uncovered that the number of fishing attacks targeted at stealing confidential information such as credit card numbers, passwords, and other financial information are on the rise, going from[6] 9 million attacks in June2004 to over 33 million in less than a year.

In this paper we are configuring and Operating snort over the nodes of Deter Emulab and results are showing the working of snort. The whole paper is divided in four sections:-1) a brief introduction to snort and deter 2). Evaluation approach followed with creation of topology on Deter Emulab. 3). Configuration of Snort, Mysql on the node of the Deter Emulab which is running on the Ubuntu operation system.

II. BRIEF INTRODUCTION TO SNORT AND DETER

A. Snort:

Snort is lightweight, open source network intrusion detection system capable of performing real time traffic analysis and packet logging on Internet Protocol (IP)[1] Networks. Snort can do protocol analysis and content searching and it can be used to detect different kind of attacks such a Common Gateway Interface(CGI) attacks, Server Message Block(SMB) probes, stealth port scans and much more.

a. Why Snort?:

Many times it's too easy for hacker community to scan your network for the open loop holes that could be running or ports that are available for attack. With this being a important fact there isn't an excuse t ignore security when deploying intrusion detection in is so easy to do. Having snort to watch your network is important because many of the security problems actually come from within the network. Best of this entire tool is free and available on most platforms.

b. Other aspect of choosing snort[8] is:

a) Snort is configurable.
b) Snort is free.
c) Snort is widely used
d) Snort runs on multiple platforms.
e) Snort is constantly updated.

c. When we can use snort?:

Snort can be used anytime you want to have basic security measures in place that allow you to sniff, log and analyze the traffic on your network.

d. Modes in which snort works:

Snort can be configuring in three main modes: sniffer, packet logger, and network intrusion detection. Sniffer mode simply reads the packets off the network and displays them in a continuous stream on the console[7]. Packet logger mode logs the packets to the disk. Network intrusion detection mode is the most complex and configurable, allowing Snort to analyze network traffic for matches against a user-defined rule set and to perform one of several actions, based on what it sees.

B. Deter Testbed:

The DETER testbed [9] aims to facilitate network security experimentation by providing an environment for researchers to perform experiments within, in a secure, isolated fashion. DETER runs a tailored configuration of the Emulab software developed at Utah [10]. It allows a security researcher to obtain exclusive use of a subset of the testbed machines, configure them into a specified topology, and access them through a firewall from across the Internet.

Rapid advances are urgently needed to defend against network attacks such as distributed denial of service, worms, and viruses. These cyber-security problems include some of strategic importance, like the protection of critical infrastructure. Rapid advances require an improvement in the state of the art of experimental evaluation of network security mechanisms.

Such efforts require the development of large-scale security testbeds combined with new frameworks and standards for testing and benchmarking to make the testbeds truly useful. Current impediments to evaluating network security mechanisms include lack of scientific rigor, lack of relevant and representative network data [11]; inadequate models of defense mechanisms; and inadequate models of the network, background, and attack traffic data [12]. The latter is challenging because of the complexity of interactions among traffic, topology, and protocols [12, 13].

Cyber-defense research has been severely limited by the lack of a public experimental infrastructure for testing new theories and new technologies in realistic scenarios. It is both unclear and unproven that technologies tested on small subnet-sized topologies modeled by a few machines will scale to realistic Internet environments.

To meet this challenge, the cyber-DEfense Technology Experimental Research (DETER) testbed [14] has been developed. The DETER testbed is intended to provide an experimental infrastructure to support the development and demonstration of next-generation information security technologies. DETER provides a medium-scale facility for safe, repeatable security-related experimentation, to validate theory and simulation.

The DETER testbed is implemented as an Emulab [15] cluster, using the comprehensive and powerful cluster testbed control package developed by Jay Lepreau and his colleagues at the University of Utah.

With a current design point of several hundred experimental nodes, the DETER testbed provides an intermediate point between small-scale and Internet-scale experiments.

Since it is chartered to support scientific investigation, the testbed is designed with experimental repeatability as a fundamental requirement. Repeatability allows experimenters, to deeply investigate, validate, and find alternative explanations for their research results and to build upon the results of others.

In addition to the harneeded to conduct experiments, the DETER testbed provides tools that aid the experimenters, many of which are being developed by experimenters themselves.

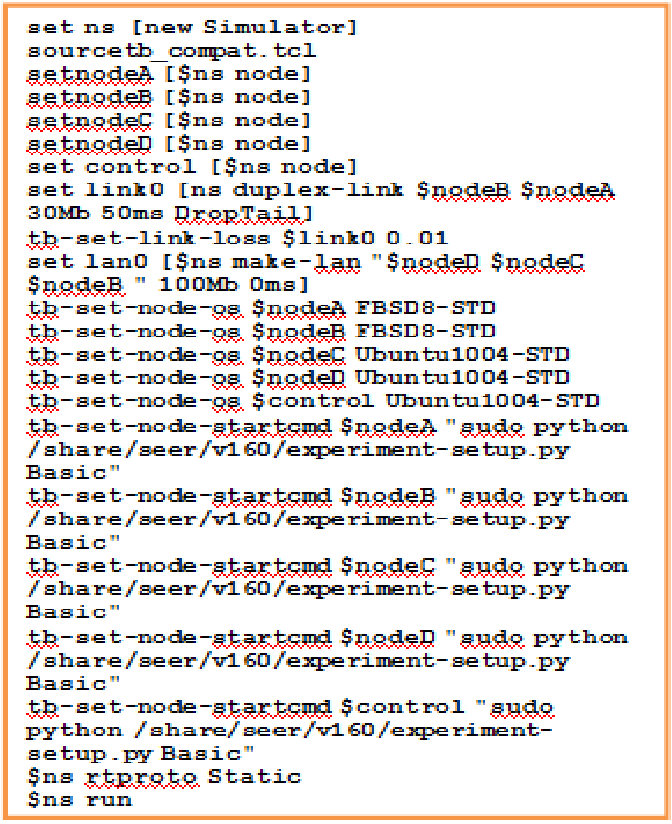
III. EVALUATION APPROACH FOLLOWED

A testing approach can be a theoretical model, simulation, emulation or deployment in an operational network. No approach is inherently good or bad, and each can answer some set of research questions. Knowing the limitations and implicit assumptions of each approach can help to choose an appropriate testing tool for their hypothesis. Inappropriate tools lead to incorrect results.

This paper focuses on report emulation approach. We evaluated snort with experiments on the DETER testbed using SEER GUI BETA6 environment (Benzel et at., 2006) (MIrkovic et al, 2007). The test bed is located at the USC Information Sciences Institute and UC Berkeley, and allows security researchers to evaluate attacks and defenses in a controlled environment.

A. *Experiment Topology:*

Figure 1 show our experiment topology definition and Figure 2 shows the experimental topology.



```
set ns [new Simulator]
sourcetb_compat.tcl
setnodeA [$ns node]
setnodeB [$ns node]
setnodeC [$ns node]
setnodeD [$ns node]
set control [$ns node]
set link0 [ns duplex-link $nodeB $nodeA
30Mb 50ms DropTail]
tb-set-link-loss $link0 0.01
set lan0 [$ns make-lan "$nodeD $nodeC
$nodeB " 100Mb 0ms]
tb-set-node-os $nodeA FBSD8-STD
tb-set-node-os $nodeB FBSD8-STD
tb-set-node-os $nodeC Ubuntu1004-STD
tb-set-node-os $nodeD Ubuntu1004-STD
tb-set-node-os $control Ubuntu1004-STD
tb-set-node-startcmd $nodeA "sudo python
/share/seer/v160/experiment-setup.py
Basic"
tb-set-node-startcmd $nodeB "sudo python
/share/seer/v160/experiment-setup.py
Basic"
tb-set-node-startcmd $nodeC "sudo python
/share/seer/v160/experiment-setup.py
Basic"
tb-set-node-startcmd $nodeD "sudo python
/share/seer/v160/experiment-setup.py
Basic"
tb-set-node-startcmd $control "sudo
python /share/seer/v160/experiment-
setup.py Basic"
$ns rtproto Static
$ns run
```

Figure 1. Experiment topology definition.

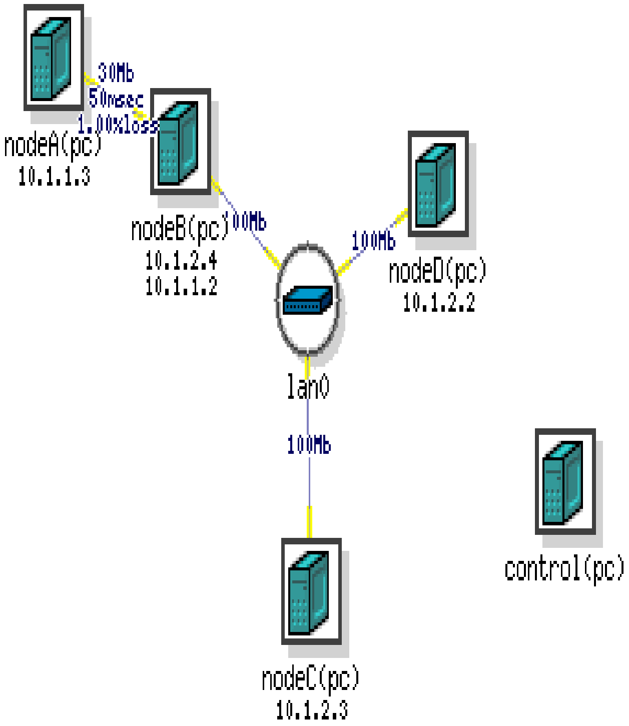


Figure 2. Experiment topology

The snort will be installed and experimented on node C of the Deter Testbed.

IV. RESULTS AND DISCUSSIONS

After generating topology in Deter testbed the following steps are used to access the node of Deter testbed.

A. *Accessing the node using SSH:*

a. The node is accessed using Putty by entering Hostname users.isi.deterlab.net and port no 22.

b. Enter :-
Login as: - snort123
Password: - *******

c. Then Type command
ssh<name of the node> Example :- ssh pc106

After accessing the particular node C of the deter testbed then snort is installed on node C by following steps:-

B. *Installation of Snort:*

a. sudo apt-get install snort-mysql

b. Then do the necessary changes in snort-conf file.

With the completion of installation of snort on node C then mysql is installed and configure for logging the alerts. The below written steps are followed for installation and configuration of mysql on node C:-

C. *Installation of mysql:*

a. Sudo apt-get install mysql-server

b. Enter the password:-******

D. *Creation of database and tables:*

a) Mysql –u root –p

b) Enter Password:- ******

E. *Creation of Database:*

a. Mysql> CREATE DATABASE snort;

F. *Creation of User:*

a. Mysql>CREATE user acid@localhost identified by ‘luvpreet’;

G. *Granting Access rights to user acid:*

b. Mysql> Grant USAGE on *.* to acid@localhost identified by “insert-password-here”;

c. Mysql>GrantINSERT,UPDATE,DELETE,ALTER,CREATE,SELECT on snort.* to acid@localhost;

d. Mysql>Flush privileges;

e. Mysql>Exit;

H. *Creation of Table:*

a. Cd /usr/share/doc/snort.conf

b. Zcat create_mysql.gz | mysql –u root –h localhost –p snort

I. *For checking table is created or not:*

a. use snort

b. show tables

c. exit

At this step we will check whether our snort is working or not. For this follow the following steps:-

J. Testing Snort:

f. sudo snort –T -c /etc/snort/snort.conf

When we run this command some files of snort required changes the change that required is to open the particular files and delete those rules which are not correct. When all the files of snort are okay then the snort will successfully installed on Node C.

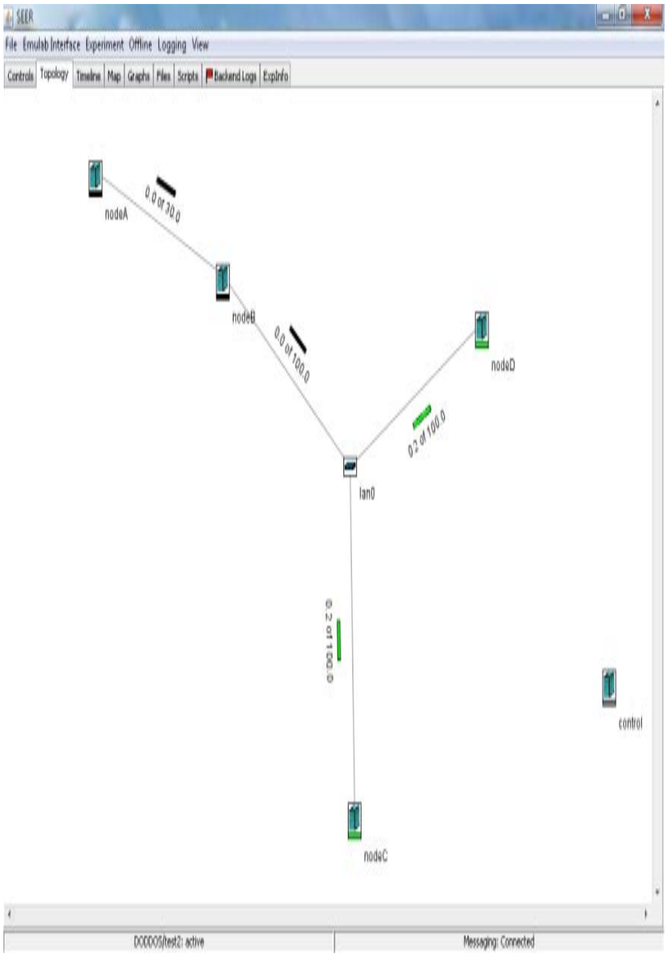
At this step we had taken results by sending the traffic from node d to node c using seer. The Results are shown in Figure 3 to Figure 7.

K. Start snort:

a. sudo service snort start
b. sudo –vd -c /etc/snort/snort.conf

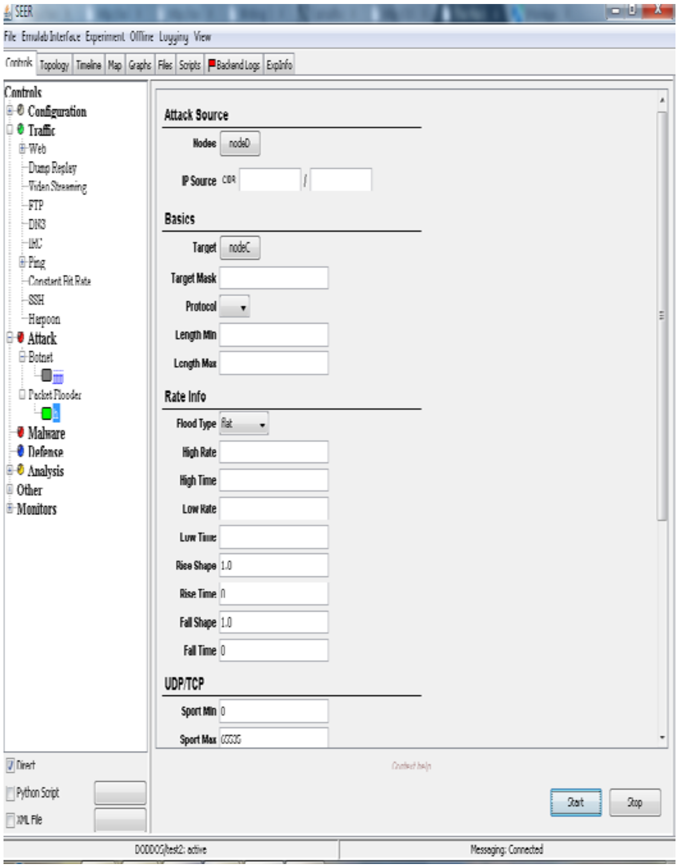
The results are taken first packet flooder attack is launched from Node D to Node C via the help of seer .The topology is generated in seer is shown in Figure 3 with the traffic moving from Node D to Node C in green colour.

Figure 4 shows hwo attack is launched from seer and Figure 5 shows the graph generated in the seer at the time of launch of packet flooder attack from Node D to Node C in seer The last Two figures 6 and 7 shows what output is generated at Node C in seer.


SEER
File Emulab Interface Experiment Offline Logging View
Controls Topology Timeline Map Graphs Files Scripts Backend Logs ExpInfo
nodeA
0.0 of 10.0
nodeB
0.0 of 100.0
nodeD
0.2 of 100.0
lan0
0.2 of 100.0
control
nodeC

DODDOS/test2: active
Messaging: Connected

Figure 3. Experimental Topology in seer showing packet flooding attack from Node D to Node C.



Attack Source
Nodes: nodeD
IP Source CIDR: /

Basics
Target: nodeC
Target Mask:
Protocol:
Length Min:
Length Max:

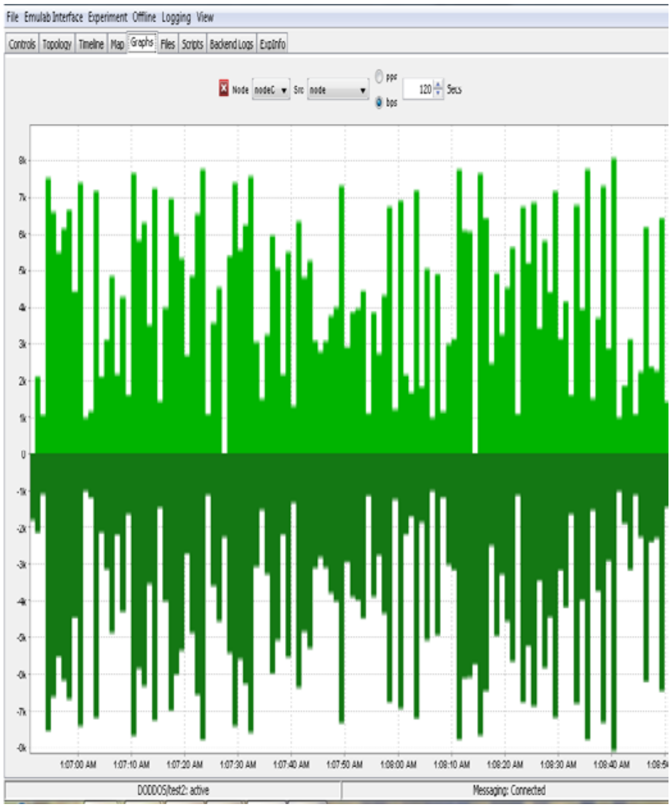
Rate Info
Flood Type: flat
High Rate:
High Time:
Low Rate:
Low Time:
Rise Shape: 1.0
Rise Time: 0
Fall Shape: 1.0
Fall Time: 0

UDP/TCP
Sport Min: 0
Sport Max: 65535

☑ Direct
☐ Python Script
☐ XML File

DODDOS/test2: active
Messaging: Connected

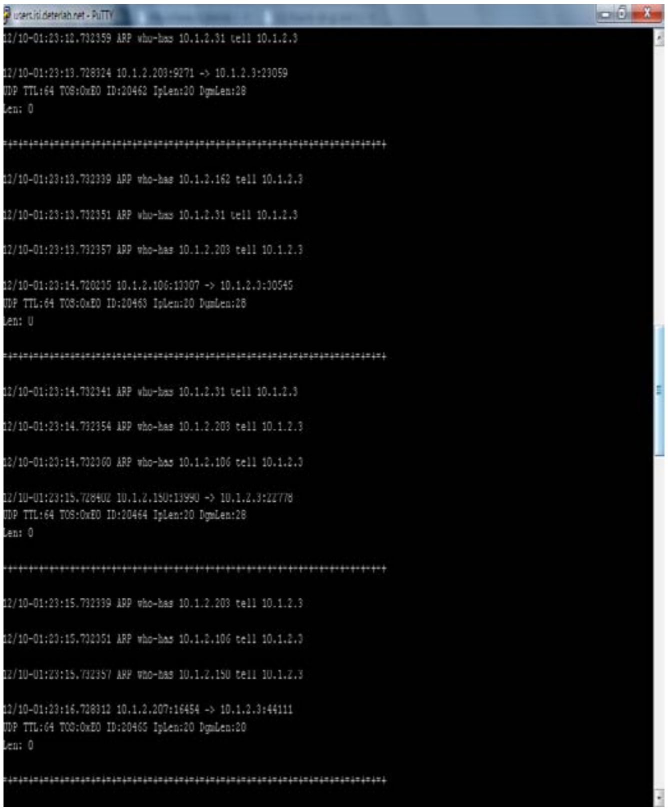
Figure 4. Showing how packet flooding attack is launched from Node D to Node C.



Node: nodeC Src: node ● bps 120 Secs

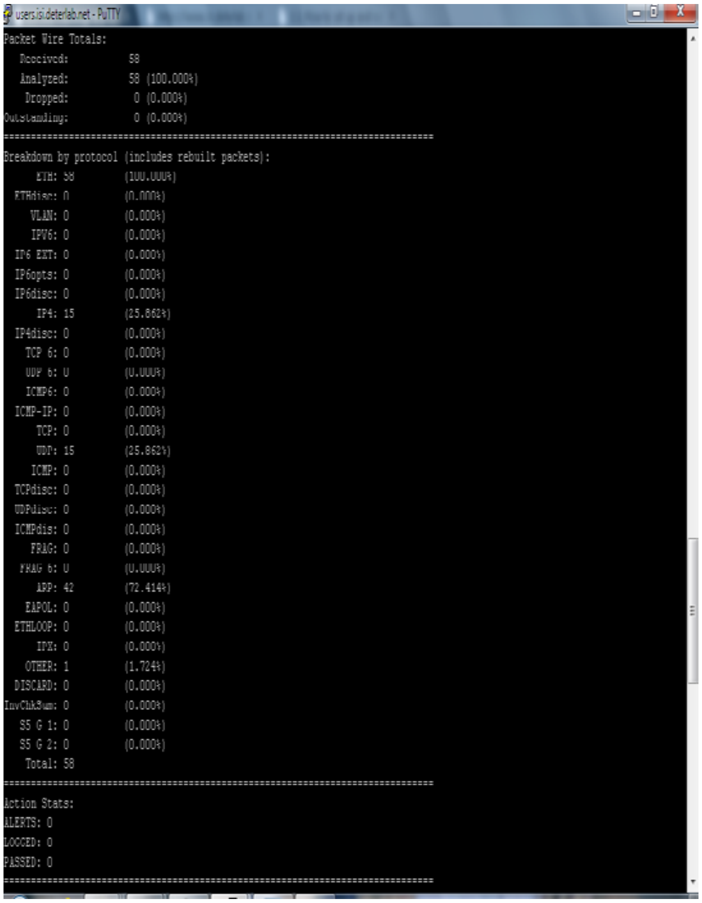
DODDOS/test2: active
Messaging: Connected

Figure 5 Graph when packet flooding attack is launched from Node D to Node C.



```
12/10-01:23:12.732359 ARP who-has 10.1.2.31 tell 10.1.2.3

12/10-01:23:13.728324 10.1.2.203:9271 -> 10.1.2.3:23059
UDP TTL:64 TOS:0xE0 ID:20462 IpLen:20 DgmLen:28
Len: 0

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

12/10-01:23:13.732339 ARP who-has 10.1.2.162 tell 10.1.2.3

12/10-01:23:13.732351 ARP who-has 10.1.2.31 tell 10.1.2.3

12/10-01:23:13.732357 ARP who-has 10.1.2.203 tell 10.1.2.3

12/10-01:23:14.720235 10.1.2.106:13307 -> 10.1.2.3:30545
UDP TTL:64 TOS:0xE0 ID:20463 IpLen:20 DgmLen:28
Len: 0

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

12/10-01:23:14.732341 ARP who-has 10.1.2.31 tell 10.1.2.3

12/10-01:23:14.732354 ARP who-has 10.1.2.203 tell 10.1.2.3

12/10-01:23:14.732360 ARP who-has 10.1.2.106 tell 10.1.2.3

12/10-01:23:15.728402 10.1.2.150:13990 -> 10.1.2.3:22778
UDP TTL:64 TOS:0xE0 ID:20464 IpLen:20 DgmLen:28
Len: 0

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

12/10-01:23:15.732339 ARP who-has 10.1.2.203 tell 10.1.2.3

12/10-01:23:15.732351 ARP who-has 10.1.2.106 tell 10.1.2.3

12/10-01:23:15.732357 ARP who-has 10.1.2.150 tell 10.1.2.3

12/10-01:23:16.728312 10.1.2.207:16454 -> 10.1.2.3:44111
UDP TTL:64 TOS:0xE0 ID:20465 IpLen:20 DgmLen:20
Len: 0

+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Figure 6 Output at Node C when Packet Flooder attack is launched from Node D to Node C.



```
Packet Wire Totals:
   Received:          58
   Analyzed:          58 (100.000%)
    Dropped:           0 (0.000%)
Outstanding:           0 (0.000%)
===============================================================================
Breakdown by protocol (includes rebuilt packets):
      ETH: 58        (100.000%)
  ETHdisc: 0         (0.000%)
     VLAN: 0         (0.000%)
     IPV6: 0         (0.000%)
  IP6 EXT: 0         (0.000%)
  IP6opts: 0         (0.000%)
  IP6disc: 0         (0.000%)
      IP4: 15        (25.862%)
  IP4disc: 0         (0.000%)
    TCP 6: 0         (0.000%)
    UDP 6: 0         (0.000%)
    ICMP6: 0         (0.000%)
  ICMP-IP: 0         (0.000%)
      TCP: 0         (0.000%)
      UDP: 15        (25.862%)
     ICMP: 0         (0.000%)
  TCPdisc: 0         (0.000%)
  UDPdisc: 0         (0.000%)
  ICMPdis: 0         (0.000%)
     FRAG: 0         (0.000%)
   FRAG 6: 0         (0.000%)
      ARP: 42        (72.414%)
    EAPOL: 0         (0.000%)
  ETHLOOP: 0         (0.000%)
      IPX: 0         (0.000%)
    OTHER: 1         (1.724%)
  DISCARD: 0         (0.000%)
InvChkSum: 0         (0.000%)
   S5 G 1: 0         (0.000%)
   S5 G 2: 0         (0.000%)
    Total: 58
===============================================================================
Action Stats:
ALERTS: 0
LOGGED: 0
PASSED: 0
===============================================================================
```

Figure 7 Report Generated by snort when packet flooder Using the Template

So from the results it is shown that the snort is detecting the attacks which is launched with the help of seer.

Now we have a clear image that Snort An Intrusion Detection system which is helpful in detecting the attacks which are generated at our loop holes. By this we can protect our system from the attacks.

V. CONCLUSION

In this paper we provided only simple how to configure the snort on node of deters and results are taken by performing the packet flooder attacks from seer. But at present we do not have any mechanism to show whether these attacks are logged in the mysql database or not. What kind of attacks are logged so in future we are planning to check the log files and more experiments or attacks will be launched and results will be taken.

VI. ACKNOWLEDGEMENT

We would like to express our gratitude to all those who gave us the possibility to complete this experimental work. We are extremely thankful to all colleagues and faculty members for their constructive criticism and guidelines.

VII. REFERENCES

[1] M John, C Alan, A Julie (2002) 'Role of Intrusion Detection System', CERT Coordination center.

[2] J Anderson, 'Computer Security Threat Monitoring and Surveillance',(1980) tech. report, James P. Anderson Co., Fort Washington, Pa.

[3] D.E. Denning, "An Intrusion Detection Model," (2, Feb. 1987) IEEE Trans. Software Eng., Vol. SE- 13, No., pp. 222-232.

[4] J. Allen et al., State of the Practice of Intrusion Detection Technologies, (2000.) Tech Report CMU/SEI-99-TR-028, Carnegie Mellon Univ., Software Engineering Inst., Pittsburgh,

[5] G Meera, S.K.K (2010) 'Detecting and Preventing attack using network intrusion detection system' IJCSS,Vol 2, Issue 3.

[6] CGI Group (2002) 'Intrusion Detection Systems (IDS) Introduction and overview', ppt

[7] http://i.i.com.com/cnwk.1d/i/tr/downloads/home/1597491705_chapter_4.pdf.

[8] S Charlie, W Paul, H Bert 'Snort for dummies' book.

[9] T. Benzel, R. Braden, D. Kim, C. Neuman, A. Joseph, K. Sklower, R. Ostrenga, and S. Schwab. Experience with DETER: A Testbed for Security Research. In Proceedings of the 2nd IEEE Conference on Testbeds and Research Infrastructures for the Development of Networks and Communities (TridentCom 2006), Barcelona, SPAIN, March 2006.

[10] B. White, J. Lepreau, L. Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar. 'An

Integratede Experimental Environment for Distributed Systems and Networks'. In Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (ODSI), pp. 255-270, Boston, MA, Dec. 2007.

[11] J McHugh,. Testing intrusion detection systems: A critique of the 1998 and 1999 DARPA intrusion detection system valuations as performed by Lincoln Laboratory. ACM Transactions on Information and System Security 3, 4 (Nov. 2000), 262-294.

[12] S Floyd. and E Kohler,. Internet research needs better models. Hotnets-I (Oct. 2002).

[13] S Floyd,. and Paxson, V. Difficulties in simulating the Internet. IEEE/ACM Transactions on Networking 9, 4 (Aug 2001), 392-403.

[14] Members of the Deter and EMIST Team, Cyber Defense Technology Networking and Evaluation, Communications of the ACM 47(3), March 2004, pp 58-61.

[15] B White,., J. Lepreau, L.Stoller, R. Ricci, S. Guruprasad, M. Newbold, M. Hibler, C. Barb, and A. Joglekar.'An Integrated experimental environment for distributed systems and networks'. In Proceedings of the Fifth Symposium on Operating Systems Design and Implementation (OSDI02), (Dec. 2002). Pp 255-270.

.