Signature-Based Network Intrusion Detection System Using SNORT And WINPCAP

Sagar N. Shah*
M.E. (Computer Science & Engineering),
Parul Institute of Engineering & Technology,
Vadodara, Gujarat, India

Ms. Purnima Singh
Assistant Professor, Computer Science & Engineering,
Parul Institute of Engineering & Technology,
Vadodara, Gujarat, India

***Abstract*- Nowadays, organizations discover that it is essential to protect their valuable information and internal resources from unauthorized access like deploying firewall. As the use of internet is growing rapidly the possibility of attack is also increasing in that ratio. Signature is the pattern that you look for inside a data packet. Signatures may be present in different parts of a data packet depending upon the nature of the attack. Intrusion detection system's main role in a network is to help computer systems to prepare and deal with the network attacks. Intrusion detection systems (IDS) have become a key component in ensuring the safety of systems and networks. These systems enforce a security policy by inspecting arriving packets for known signatures (patterns). Snort is mostly used signature based IDS because of it is Lightweight and open source software. Basic analysis and security engine (BASE) is also used to see the alerts generated by Snort. In this paper we have implemented the signature-based Network intrusion detection using Snort and WinPcap.**

Keywords*- *Network Intrusion Detection System, Snort, Signature-based, WinPcap, BASE

I. INTRODUCTION

As the use of technology is increases, risk associated with technology is also increases. Network security is the big challenge among the researchers. People are working in the field of network security from 1987 when Dorothy Denning published an intrusion detection model [1]. But till now we did not get any perfect solution. While the availability of continuous communication has created many new opportunities, it has also brought new possibilities for malicious users. The Importance of network Security is therefore growing; one of the ways of malicious activity detection on a network is by using Intrusion Detection System. Intrusion detection system's main role in a network is to help computer systems to prepare and deal with the network attacks.

Intrusion detection functions include [2]:

- Analysis of abnormal activity patterns
- Analyzing system configurations and vulnerabilities
- Ability to recognize patterns typical of attacks
- Monitoring and analyzing both user and system activities
- Assessing system and file integrity

Intrusion Detection Systems (IDS) inspect arriving packets for malicious content (signatures) as defined by a security policy. Unfortunately, comparing packet headers and payloads against a policy can be complex and time-consuming. For example, it has been found that content matching (scanning for signatures) accounts for more than 70% of the packet processing time [3],[4].

This paper focuses on analyzing the abnormal activity that has been detected by our Intrusion Detection System using Snort and WinPcap. Snort is a popular NIDS that is used to audit network packets and compare those packets with the database of known attack signature and this attack signature database must be updated time by time.

II. SIGNATURE-BASED NETWORK IDS

A signature-based NIDS examines ongoing traffic, activity, transactions, or behaviour for matches with known patterns of events specific to known attacks. As with antivirus software, a signature-based NIDS requires access to a current database of attack signatures and some way to actively compare and match current behaviour against a large collection of signatures.

Signature based detection system (also called misuse based), this type of detection is very effective against known attacks [5]. It implies that misuse detection requires specific knowledge of given intrusive behaviour. An example of Signature based Intrusion Detection System is SNORT.

Advantages [6]:

- Signature definitions are modeled on known intrusive activity. So, the user can examine the signature database, and quickly determine which intrusive activity the misuse detection system is programmed to alert on.
- Misuse detection system begins protecting your network immediately upon installation.
- There are low false positives as long as attacks are clearly defined in advance.
- When an alarm fires, the user can relate this directly to a specific type of activity occurring on the network.

Disadvantages [6]:

- One of the biggest problem for Signature based NIDS is how to keep up with large volume of incoming traffic when each packet needs to be compared with every signature in the database. So, processing the whole traffic is so time-consuming and will slow down the throughput of the system.
- Misuse detection system must have a signature defined for all of the possible attacks that an attacker may launch against your network. This leads to the necessity for frequent signature updates to keep the signature database of your misuse detection system up-to-date.
- Misuse detection has a well-known problem of raising alerts regardless of the outcome. For example a window worm trying to attack a Linux system, the misuse IDS will send so many alerts for unsuccessful attacks which may be hard to manage.
- Someone may set up the misuse detection system in their lab and intentionally try to find ways to launch attacks that bypass detection by the misuse detection system.
- The knowledge about attacks is very dependent on the operating system, version and application hence tied to specific environments.

III. Component of Snort

Snort is basically the combination of multiple components. All the component work together to find a particular attack and then take the corresponding action that is required for that particular attack. Basically it consists of following major components as shown in figure 1 [7]:

1. Packet Decoder
2. Preprocessor
3. Detection Engine
4. Logging and Alerting System
5. Output Modules

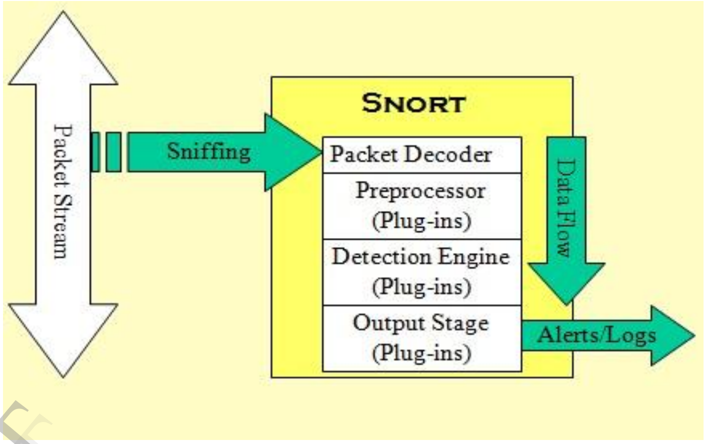


Figure 1: Component of Snort [7]

Packet comes from internet and enters into packet decoder and it goes through several phases, required action is taken by snort at every phase like if detection engine found any miscellaneous content in packet then it drop that packet and in the way towards output module packet is logged in or alert is generated.

1. Packet Decoder:

The packet decoder collects packet from different network interfaces and then send to be preprocessor or sent to the detection engine. Network interface might be Ethernet, SLIP, PPP and so on.

2. Preprocessor:

It works with snort to modify or arrange the packet before detection engine to apply some operation on packet if packet is corrupted. Sometimes they also generate alert if any anomalies found in the packet. Basically it matches the pattern of whole string. so, by changing the sequence or by adding some extra value intruder can fool the IDS but preprocessor re-arranges the string and IDS can detect the string. Preprocessor does one very important task i.e. defragmentation. Because sometimes intruder break the signature into two parts and send them in two packets. So, before checking the signature both packet should be defragmented and only then signature can be found and this is done by preprocessor.

3. Detection Engine:

Its main work is to find out intrusion activity exits in packet with the help of snort rules and if found then apply appropriate rule otherwise it drops the packet. It takes different time to respond different packet and also depends upon the power of machine and number of rules defines in the system.

4. Logging and Alerting System:

This system is responsible from the generation of alerts and logging of packets and messages. Depending upon what the detection engine finds inside a packet, the packet may be used to log the activity or generate an alert. All of the log files are stored under a preconfigured location by default. This location can be configured using command line options. There are many command line options to modify the type and detail of information that is logged by the logging and alerting system. All log files are kept by default under C:\Snort\log folder and by using –l command line option, location can be changed.

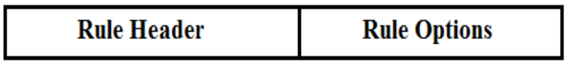
5. Output Modules:

Output modules or plug-ins save output generated by the logging and alerting system of Snort depending on how user wants for different operation. Mainly it controls the different output due to logging and alerting system. Depending on the configuration, output modules can send output messages a number of other destinations. Commonly used output modules are:

- The database module is used to store Snort output data in databases, such as MySQL, MSSQL or Oracle,
- The SNMP module can be used to send Snort alerts in the form of traps to a management server,
- The Sending Server Message Block (SMB) alerts module can send alerts to Microsoft Windows machines in the form of pop-up SMB alert windows,
- The syslog module logs messages to the syslog utility (using this module you can log messages to a centralized logging server.)

IV. Rule structure of snort

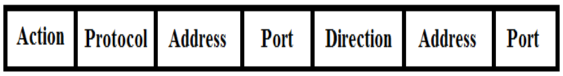
All IDS rules have two logical parts: **rule *header*** and **rule *option*** [8]. This is shown in Figure 2.



Rule Header	Rule Options

Figure 2: Basic Structure of IDS Rules

The rule header contains information about what action a rule takes. It also contains criteria for matching a rule against data packets. The options part usually contains an alert message and information about which part of the packet should be used to generate the alert message. The options part contains additional criteria for matching a rule against data packets. A rule may detect one type or multiple types of intrusion activity. Intelligent rules should be able to apply to multiple intrusion signatures.



Action	Protocol	Address	Port	Direction	Address	Port

Figure 3: Structure of IDS rule header

The ***action*** part of the rule determines the type of action taken when criteria are met and rule is exactly matched against a data packet. Typical actions are generating an alert or log message or invoking another rule.

i. **Pass –** This action tells Snort to ignore the packet. This action plays an important role in speeding up Snort operation in cases where you don't want to apply checks on certain packets. For example, if you have a vulnerability assessment host on your own network that you use to find possible security holes in your network, you may want Snort to ignore any attacks from that host. The pass rule plays an important part in such a case.

ii. **Log –** The log action is used to log a packet. Packets can be logged in different ways, as discussed later in this book. For example, a message can be logged to log files or in database. Packets can be logged with different levels of detail depending on the command line arguments and configuration file.

iii. **Alert –** The alert action is used to send an alert message when rule conditions are true for a particular packet. An alert can be sent in multiple ways. For example, you can send an alert to a file or to a console. The functional difference between Log and Alert actions is that Alert actions send an alert message and then log the packet. The Log action only logs the packet.

The ***protocol*** part is used to apply the rule on packets for a particular protocol only. This is the first criterion mentioned in the rule. Some examples of protocols used are IP, ICMP, UDP and etc.

The ***address*** part define source and destination address. Address may be a single host, multiple host or network address. The researcher can also use these parts to exclude some address from a complete network. Source and destination address are determined based on direction field. As an example, if the direction field is “->”, the ***address*** on the left side is source and the ***address*** at the right side is destination.

In case of TCP or UDP protocol, the ***port*** parts determine the source and destination ports of a packet on which the rule is applied. In case of network layer protocols like IP and ICMP, port numbers have no significance.

The ***direction*** part of the rule actually determines which address and port number is used as source and which as destination. Snort utilizes a pattern matching model for detection of network attack signatures using identifiers such as TCP fields, IP addresses, TCP/UDP port numbers, ICMP type/code, and strings contained in the packet payload. For example, Snort may have a rule such as the following:

Alert tcp $HOME_NET 12345 -> $EXTERNAL_NET any (msg:”IDS80-BACKDOOR ACTIVITY- Possible Netbus/GabanBus”; flags: SA)

This is the pattern-matching rule for the Netbus Trojan. Let us break this rule down to understand how the Snort packet engine recognizes signatures.

Alert : this is an alert message

Tcp : snort will be focused on the IP protocol

$HOME_NET : HOME_NET is a variable set to an organization’s IP address range (for example 10.0.0.0/16)

12345 : destination TCP port number of original SYN packet from $EXTERNAL_NET. This represents the SYN/ACK portion of the TCP handshake.

-> **:** Indicate that traffic will be matched for source IP of HOME_NET and destination IP of EXTERNAL_NET.

$EXTERNAL_NET :EXTERNAL_NET is a variable set to an IP address range to be matched. For instance, this might be set to 0.0.0.0 if the IDS is placed at an Internet connection.

Any : the “any” keyword refers to TCP source port number for the originator of the connection

Msg “…” : this message is printed to the snort.alert log file.

Flags : SYN and ACK flags are set. Other flags such as PSH, FIN, RST, and URG could also be specified as part of a signature.

V. Snort NIDS Topology

From the figures referred from [7] given below concept of signature based IDS can easily understand. It is clear that when any person sends data inside the network so first of all it goes to Default gateway and check rule and if found malicious then it discards the packet otherwise send to destination system.

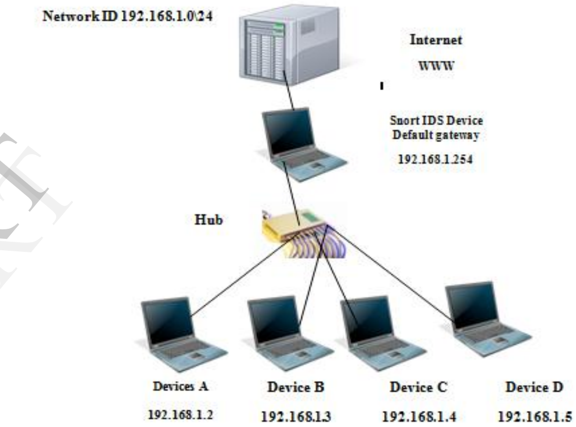


Figure 4: Snort NIDS Topology [7]

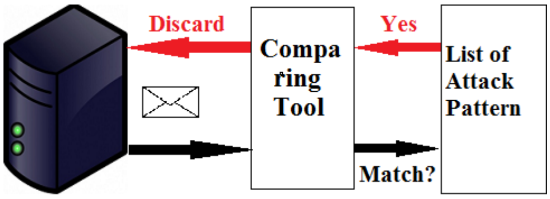


Figure 5: Snort Signature Database [7]

In figure 4 Snort IDS computer is connected through the internet. Networks send packets to snort IDS devices. Before reaching the packet to destination, default gateway checks that packet and if packet is malicious then snort IDS device discards the packet otherwise send packet to devices and if in figure 5 working of IDS device is clearly mention that how device checks the packets. So, when a packet comes to device then it use comparing tool to check that packet from the database of signature stored in IDS device and if it get result that packet is matched from the database then IDS system discard the packet otherwise it sends the packet to destination system.

VI. TOOLS USED IN SIGNATURE-BASED NIDS SYSTEM

To implement signature-based network Intrusion detection System; we need to install some tools, such as Snort, BASE and WinPcap.

Snort [9]

Snort is an open source network intrusion detection and prevention system [9] (available at http:// www.snort.org/snort-downloads?). It can analyze real-time traffic analysis and data flow in network. It is able to check protocol analysis and can detect different type of attack. In NIDS snort basically checks packet against rule written by user. Snort rules can be written in any language, its structure is also good and it can be easily read and rules can be modify also. In buffer overflow attack, snort can detect the attack by matching the previous pattern of attacks and then will take appropriate action to prevent from attack. In signature based IDS system if pattern matches then attack can be easily found but when a new attack comes then system fails but snort overcome this limitation by analyzing the real-time traffic. Whenever any packet comes into network then snort checks the behaviour of network if performance degrades of network then snort stop the processing of packet, discards the packet and stores its detail in the signature database [10].

WinPcap

WinPcap is an open source library for packet capture and network analysis [11] for the Win32 platforms.

The purpose of WinPcap is to give this kind of access to Win32 applications; it provides facilities to:

- capture raw packets, both the ones destined to the machine where it's running and the ones exchanged by other hosts (on shared media)
- Filter the packets according to user-specified rules before dispatching them to the application.
- Transmit raw packets to the network.
- Gather statistical information on the network traffic.

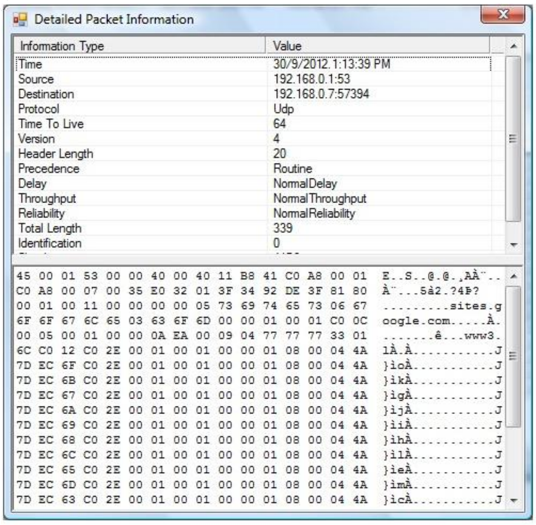
Basic Analysis and Security Engine (BASE) [12]

BASE is a web interface to perform analysis of intrusions that snort has detected on the network. This application provides a web front-end to query and analyze the alerts coming from a SNORT IDS system. It uses a user authentication and role-base system; so that you as the security admin can decide what and how much information each user can see. It also has a simple to use, web-based setup program for people not comfortable with editing files directly [12].

VII. IMPLEMENTATION DETAILS

WinPcap provide the packet-capture and filtering engines of many open source and commercial network tools, including protocol analyzers (packet sniffers), network monitors, network intrusion detection systems, traffic-generators and network-testers. It also support saving captured packets to a file [13], and reading files containing saved packets; applications can be written, using WinPcap, to be able to capture network traffic and analyze it, or to read a saved capture and analyze it, using the same analysis code. A capture file saved in the format that WinPcap use can be read by applications that understand that format, such as tcpdump, Wireshark, CA NetMaster.

Snapshot



Detailed Packet Information

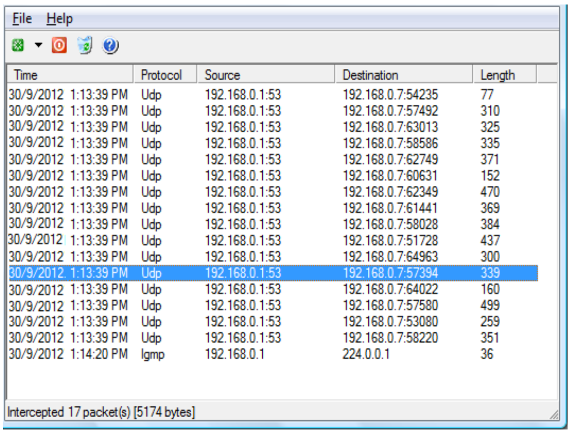
Information Type	Value
Time	30/9/2012 1:13:39 PM
Source	192.168.0.1:53
Destination	192.168.0.7:57394
Protocol	Udp
Time To Live	64
Version	4
Header Length	20
Precedence	Routine
Delay	NormalDelay
Throughput	NormalThroughput
Reliability	NormalReliability
Total Length	339
Identification	0

```
45 00 01 53 00 00 40 00 40 11 B8 41 C0 A8 00 01  E..S..@.@.¸AÀ¨..
C0 A8 00 07 00 35 E0 32 01 3F 34 92 DE 3F 81 80  À¨...5à2.?4Þ?
00 01 00 11 00 00 00 00 05 73 69 74 65 73 06 67  .........sites.g
6F 6F 67 6C 65 03 63 6F 6D 00 00 01 00 01 C0 0C  oogle.com.....À.
00 05 00 01 00 00 0A EA 00 09 04 77 77 77 33 01  .......ê...www3.
6C C0 12 C0 2E 00 01 00 01 00 00 01 08 00 04 4A  lÀ.À...........J
7D EC 6F C0 2E 00 01 00 01 00 00 01 08 00 04 4A  }ìoÀ...........J
7D EC 6B C0 2E 00 01 00 01 00 00 01 08 00 04 4A  }ìkÀ...........J
7D EC 67 C0 2E 00 01 00 01 00 00 01 08 00 04 4A  }ìgÀ...........J
7D EC 6A C0 2E 00 01 00 01 00 00 01 08 00 04 4A  }ìjÀ...........J
7D EC 69 C0 2E 00 01 00 01 00 00 01 08 00 04 4A  }ìiÀ...........J
7D EC 68 C0 2E 00 01 00 01 00 00 01 08 00 04 4A  }ìhÀ...........J
7D EC 6C C0 2E 00 01 00 01 00 00 01 08 00 04 4A  }ìlÀ...........J
7D EC 65 C0 2E 00 01 00 01 00 00 01 08 00 04 4A  }ìeÀ...........J
7D EC 6D C0 2E 00 01 00 01 00 00 01 08 00 04 4A  }ìmÀ...........J
7D EC 63 C0 2E 00 01 00 01 00 00 01 08 00 04 4A  }ìcÀ...........J
```

Figure 6: Packet Details

As soon as we start the internet, the host systems on which we access this module start capturing the packets. It shows the data in the decimal format. The details of the captured packets are shown in the snapshot. The Default Gateway used to capture and monitor the packet is as follows: Getting IP address to Keep watch / monitor.

m_Monitor = new Socket (AddressFamily.InterNetwork, SocketType.Raw, ProtocolType.IP);



Time	Protocol	Source	Destination	Length
30/9/2012 1:13:39 PM	Udp	192.168.0.1:53	192.168.0.7:54235	77
30/9/2012 1:13:39 PM	Udp	192.168.0.1:53	192.168.0.7:57492	310
30/9/2012 1:13:39 PM	Udp	192.168.0.1:53	192.168.0.7:63013	325
30/9/2012 1:13:39 PM	Udp	192.168.0.1:53	192.168.0.7:58586	335
30/9/2012 1:13:39 PM	Udp	192.168.0.1:53	192.168.0.7:62749	371
30/9/2012 1:13:39 PM	Udp	192.168.0.1:53	192.168.0.7:60631	152
30/9/2012 1:13:39 PM	Udp	192.168.0.1:53	192.168.0.7:62349	470
30/9/2012 1:13:39 PM	Udp	192.168.0.1:53	192.168.0.7:61441	369
30/9/2012 1:13:39 PM	Udp	192.168.0.1:53	192.168.0.7:58028	384
30/9/2012 1:13:39 PM	Udp	192.168.0.1:53	192.168.0.7:51728	437
30/9/2012 1:13:39 PM	Udp	192.168.0.1:53	192.168.0.7:64963	300
30/9/2012 1:13:39 PM	Udp	192.168.0.1:53	192.168.0.7:57394	339
30/9/2012 1:13:39 PM	Udp	192.168.0.1:53	192.168.0.7:64022	160
30/9/2012 1:13:39 PM	Udp	192.168.0.1:53	192.168.0.7:57580	499
30/9/2012 1:13:39 PM	Udp	192.168.0.1:53	192.168.0.7:53080	259
30/9/2012 1:13:39 PM	Udp	192.168.0.1:53	192.168.0.7:58220	351
30/9/2012 1:14:20 PM	Igmp	192.168.0.1	224.0.0.1	36

Intercepted 17 packet(s) [5174 bytes]

Figure 7: Packet Information and Hex Data

Once we select any packet by double click on it that is shown in the first snapshot, we are able to see the details of the packet i.e. the header field and the payload. The header part is consist of source IP address and destination IP address, name of the protocol, Time to live field, version of a protocol, Header length, various type of services and the total length field. The data of the header field is shown in the decimal form whereas the data of the payload is display in the hexadecimal form.

VIII. CONCLUSION AND FUTURE WORK

Security is a big issue for all networks in today's enterprise environment. Hackers and intruders have made many successful attempts to bring down high-profile company networks and web services. Snort is free and powerful software that capable of performing real-time traffic analysis and packet logging. It considered as the heart of Intrusion Detection System. Once the Snort will identify any intrusion then it will send alert to security person and security person will take required action immediately.

However, snort is a strong Intrusion Detection System; the problem is that snort system is not familiar with Windows Operating System. In this paper, Signature-based Network Intrusion Detection System with snort has been implemented and configured with windows-based environment. The results show that it is possible to configure snort IDS with Windows and it can be configured as a firewall.

The future work is to develop a parallel technique (parallelization) for improving the performance of signature-based network intrusion detection system and reduce the processing time of the traffic.

REFERENCES

[1] D. E. Denning. "An Intrusion-Detection Model". IEEE transactions on software engineering, Volume : 13 Issue: 2, February 1987.

[2] Harley Kozushko, "Intrusion Detection: Host-Based and Network-Based Intrusion Detection Systems", on September 11, 2003.

[3] S. Antonatos K.G. Anagnostakis and E. P. Markats. Generating realistic workloads for network intrusion detection systems. In *Proceedings ACM Workshop on Software and Performance.*, 2004.

[4] Mike Fisk and George Varghese. Fast content-based packet handling for intrusion detection. Technical report, University of California at San Diego, 2001.

[5] D. E. Denning, "An intrusion-detection model." IEEE Transactions on Software Engineering, Vol.SE-13(No.2):222-232, Feb. 1987.

[6] Guan Xin and Li Yun-jie, "A new Intrusion Prevention Attack System Model based on Immune Principle", International Conference on e-Business and Information System Security (EBISS), in IEEE, pp. 1-4, 2010.

[7] Vinod Kumar, Vinay Pathak, Dr. Om Prakash Sangwan, "Evaluation of Buffer Overflow and NIDPS", International Journal on Computer Science and Emerging Trends (IJCSET), August issue, 2012.

[8] Rafeeq A. (2003). *Intrusion Detection Systems with Snort advance IDS technique Using Snort, Apache, MySQL, PHP, and ACID.* Publication Pearson Education. Upper Saddle River, New Jersey.

[9] Caswell, Brian. "Snort - The Open Source Network IDS: More info about Snort" URL: http://www.snort.org

[10] Intrusion Detection with SNORT: Advanced IDS Techniques Using SNORT, Apache, MySQL, PHP, and ACID by Rafeeq Ur Rehman.

[11] The industry-standard windows packet capturelibrary,"Winpcap,"2010.[Online]. Available: www.winpcap.org

[12] Basic Analysis and Security Engine (BASE) project (2012). Available: http://base.secureideas.net/about.php.

[13] Jiekun Zhang, Dafang Zhang and Kun Huang, ― A Regular Expression Matching Algorithm using Transition Merging‖ IEEE, 2009.

AUTHOR'S PROFILE

Passport Size Latest Color Photo	**Author's Name** (Font Size – 10, Times New Roman, Bold) personal profile which contains their education details, their publications, research work, membership, achievements, with photo that will be maximum 200-400 words. (Font Size – 8, Times New Roman)

Passport Size Latest Color Photo	**Author's Name** ((Font Size – 10, Times New Roman, Bold) personal profile which contains their education details, their publications, research work, membership, achievements, with photo that will be maximum 200-400 words. (Font Size – 8, Times New Roman)