Intrusion Detection in Software Defined Networking Using Snort and Mirroring

Nithya Sampath, Jayakumar Sadhasivam, Senthil Jayavel,
N. Swetha Chindarmony and Sakshi Sharma

Abstract--- *Software-Defined Networking is a rising concept that aims to replace conventional networks by breaking up vertical integration. The control logic of network is separated from the underlying routers and switches, by logically centralized network control, and to program the network. An intrusion detection system is a software application that keeps track of a system or network for occurrence of any policy violations or malicious activity. Reports are sent to the network administrator or collected centrally using a security information and event management system when there is an occurrence of a malicious activity or policy violations. The aim of this paper is to create an Intrusion Detection System using Snort which is an open-source, free and lightweight application. The concept of the paper is to build an efficient and simplified Intrusion Detection System. First, setup a simple network topology with four virtual machines where three of them are hosts and fourth one is designed to run Snort.*

Keywords--- *Software Defined Networking (SDN), OpenFlow Protocol, Open vSwitch, Snort, Mirroring, Intrusion Detection System (IDS), Denial of Service (DoS) attacks, Distributed Denial of Service (DDoS)attacks, Switched Port Analyser (SPAN).*

I. INTRODUCTION

Software Defined Networking is a new technology for computer networking which gives the network administrators the provision for centralized network management dynamically. Software Defined Networking is associated mainly with the OpenFlow protocol. Software Defined Networking provides dynamic, centrally manageable and cost efficient network. This network infrastructure encompasses programmed, centrally managed network. A software system application called controller is provided with the management functions which is decoupled from the hardware unlike conventional networking. This permits the network administrator to efficiently handle the business desires. The software defined network traffic is formed by programming the management at the control plane without interrupting the switches at data plane. The forwarding rules in the switch to route traffic from the sender to the destination is set by the control plane which includes a centralized controller. For handling cloud service challenges like, dynamic load traffic, additional information measure demand and security problems this technique of SDN is used.

The OpenFlow protocol is mostly used by Software Defined Networking for balancing traffic load, directing the traffic and executes policies to scale the network. For preserving network and data security, an efficient security tool

Nithya Sampath, School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India. E-mail: nithya.s@vit.ac.in

Jayakumar Sadhasivam, School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India. E-mail: jayakumarsvit@gmail.com

Senthil Jayavel, Computer Science and Engineering, Nandha Engineering College, Erode, India. E-mail: senthil.j.vit@gmail.com

N. Swetha Chindarmony, School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India.

Sakshi Sharma, School of Information Technology and Engineering, Vellore Institute of Technology, Vellore, India.

Received: 19 Sept 2019 | Revised: 20 Oct 2019 | Accepted: 20 Nov 2019

is required. People with malicious intentions may cause serious threat to the organization's confidential information. If there is no proper security for protecting the data, a loophole occurs through which the attacker can intrude into the network and access the confidential information. For detecting and monitoring abnormal data packets travelling through the network, software called Intrusion Detection System is being used. When a data packet with an attacking pattern is detected, the IDS generates alert informing intrusion occurrence in the network. Snort is a popular open source utility application used for Intrusion Detection System and Intrusion Protection System. Snort based IDS generates alerts in real-time for protecting the system's risk from intruders. Snort based IDS usually have their rules in the form of a single line, which can be easily understood and modified. The data packet from the traffic network is matched between rules which are defined using Snort-IDS.

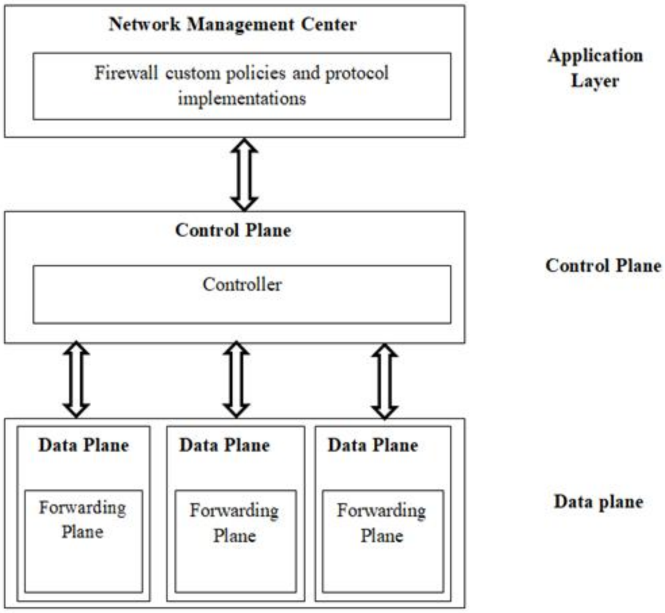


Figure 1: Software defined networking architecture

II. Background

Some background details about specification based detection and the software defined networking is being introduced in this section.

Detection based on Specifications

Specification based intrusion detection system and intrusion detection framework was put forward by Berthier et al in the year 2010, 2011. The expected outcomes activities functionalities and security rules to be considered are defined in specifications. Therefore, any anomaly from the actual specifications will be considered as a violation of security. Specifications rules for control systems, VoIP protocols, routing protocols have been recently defined for efficiently implementing security. Nick McKeown et al put forth the concept of Software Defined Networking initially. This concept separates the data and control plane for each node in a network. Each node still has the data plane but the control plane exists logically as a centralized controller. The controller centrally manages the flow entries for each packet in the network. OpenFlow is the protocol and commonly used architecture of Software Defined Networking.

III. Integration Of Software Defined Networking And Snort

Snort is a utility tool which is an open source and light-weight signature based Intrusion Detection System. The conventional Snort deployment mirrors network traffic to the Intrusion Detection System. Snort contains a pre-defined set of rules which it matches against all the packets in the network traffic, which when matched will alert the Intrusion Detection System resulting in blocking of the malicious activity by firewall.

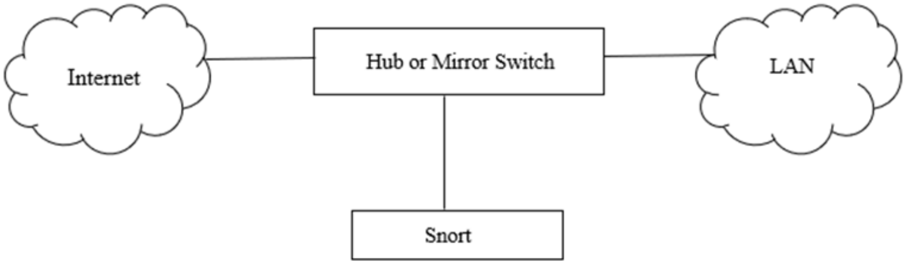


Figure 2. **Conventional Snort Deployment**

In Software Defined Networking environment the Snort service is deployed by implementing mirroring on Open VSwitch as shown in figure 2. The OpenDayLight Controller sets two output ports for each flow entry in the switch to implement mirroring. One output port of the switch is for regular forwarding of packets and the second one is for forwarding it to Snort for anomaly detection from the specified rules. The OpenDayLight controller can force stop specified traffic by commanding the Open VSwitch, if any suspicious network traffic is found.

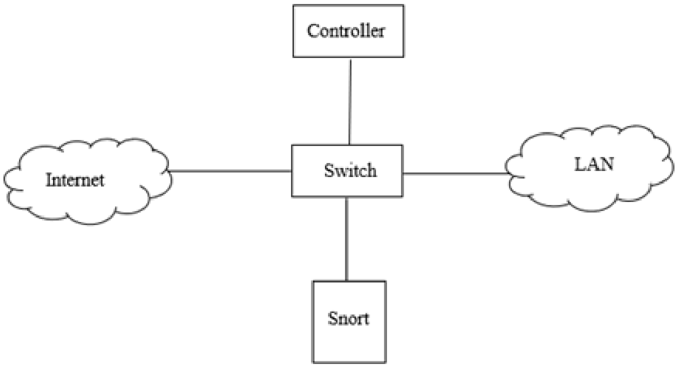


Figure 3. Snort with Software Defined Networking

IV. Related Work

The concept being discussed in**[1]** is about building up an intrusion or threat detection system in Advanced Metering Infrastructure systems using Software Defined Networking. The protocol being used is OpenFlow protocol which is mostly used in most of the systems using Software Defined Networking. The Software Defined Networking technology is integrated with the Intrusion Detection System in the Advanced Metering Infrastructure. Rules are set using Snort which is an open-source utility application. Snort architecture in such a way that all network is

Received: 19 Sept 2019 | Revised: 20 Oct 2019 | Accepted: 20 Nov 2019

forwarded to it by the process of mirroring. By using Software Defined Networking with Snort-based Intrusion Detection System, Advanced Metering Infrastructure system can provide an effectively efficient defence mechanism against intrusion and other threats to the system.

The concept in **[2]** focuses on deploying Software Defined Networking based Intrusion Detection System to cloud platform to enhance the security of cloud computing. A new Intrusion Detection and Prevention system is proposed which uses Snort-based IDS and Open vSwitch. The proposed system is compared, analysed and evaluated with the conventional approach. POX controller is used to enhance the security using Network Reconfiguration features. This approach is then directly implemented on cloud to enhance security systems in cloud.

The concept in**[3]**approaches with a system for detecting Intrusion for Software Defined Networking environment to identify and report malicious activities happening in the network to network administrators. The proposed approach is of a genetic algorithm which is to be deployed to prevent attacks in the network. The data in the network is monitored, and it provides information of data flowing through the network. Rules are made and the data flowing through the network is checked against these rules to check the presence of malicious activities in the network. Based on the type of traffic in the network, the genetic algorithm can be modified which increases the flexibility and reliability of the algorithm which prevents attacks and provides enhanced security.

The concept in**[4]**is that it proposes OpenSec which is a framework for Open-Flow based network security permits operators for security across the network. It provides abstraction so that operators can specify easy and human-readable security policies. OpenSec has a layer that operates on the top of controller for giving security services. It allows operators to define the security polices with description and a list of services to deal with malicious contents. When it found some malicious contents, it generates alerts only. It depends upon the on the processing units for scanning incoming traffic. This framework forwards the flows to security units and automatically responses towards the event generated by middleboxes. It uses the control layer that allows users to forward only traffic part to security units. OpenSec automatically deals with alerts without involving network administrator.

The concept discussed in **[5]** proposes the idea that it is possible to differentiate if network traffic flows represent malicious attacks or normal operation. The approach is using machine learning methods and self-organized maps for detecting unauthorized activities. The concept is based on the of SDN flow classification mechanism. The Kohonen algorithm has been used as self-organized maps learning method. The self-organized maps allow creating a type of structure that represents input vectors which is the collected data in measuring module with labels to specify the first packet in the flows. The clients request the system. At the same time, unauthorized activities are performed by malicious host using attack tools to the servers. The generic traffic is probed by measuring module. The servers are on separate virtual machines and clients are virtualized on the mininet OS level.

The idea proposed in **[6]** focuses on issue of threat detection as well as defence of Software Defined Networking defence. To represent security roles of Software Defined Networking it uses the matter-elements and a configuration point OpenFlow controller and OpenFlow switch. For threat of Software Defined networking, the paper formally represents the security roles and dependent function based on extension theory formalizes the NETCONF operation. The extension approach uses basic-elements from a formal viewpoint.

The concept being discussed in **[7]** focuses on FLOWGUARD framework for accurately detecting and effective resolution of firewall policy in OpenFlow based networks. When network states are updated it checks for firewall policy violation. It uses innovative resolution techniques for diverting network update conditions. For firewall Packet filtering traditional techniques has been used to handle packet violation. FLOWGUARD has been used on the top of Floodlight with three components, flow tracking, violation detection and violation resolution. The solution centrally enforces rules to eliminate flow packet violation and the flow policy that requires corresponding firewall rules.

V. ARCHITECTURE COMPONENT

Using Snort is of the few methods that Intrusion Detection Systems use to detect the occurrence of intrusion in the network.

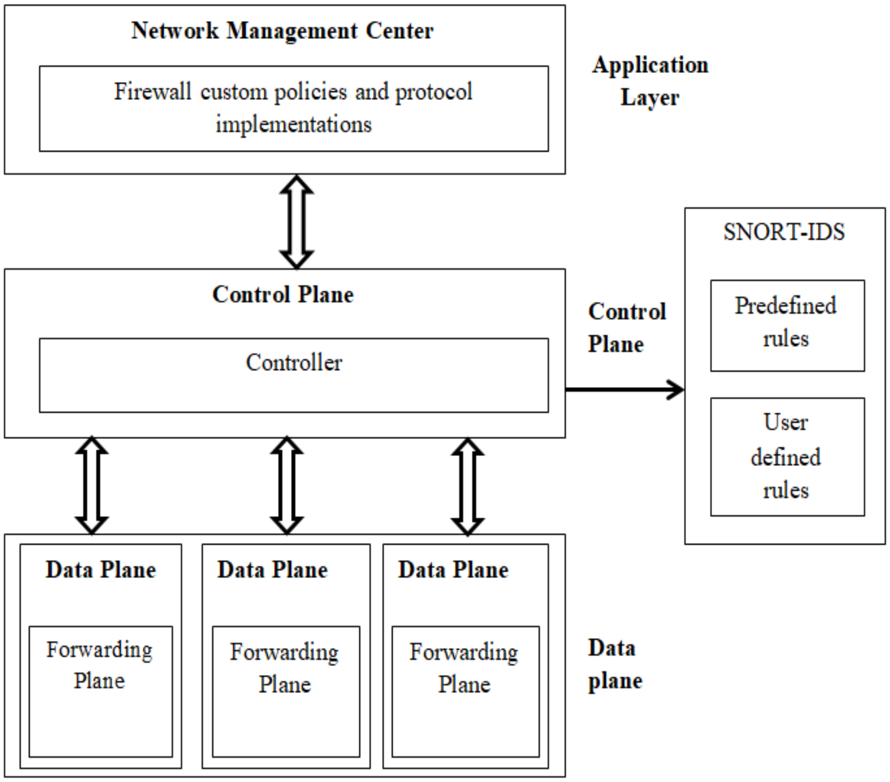


Figure.4 Architecture of Software Defined Networking with IDS

The proposed system is to create an Intrusion Detection System in Software Defined Networking. The Software Defined Networking controller, Virtual machine running IDS and network attack detecting software all are connected with each other through OpenFlow based software controlled switches.

Received: 19 Sept 2019 | Revised: 20 Oct 2019 | Accepted: 20 Nov 2019

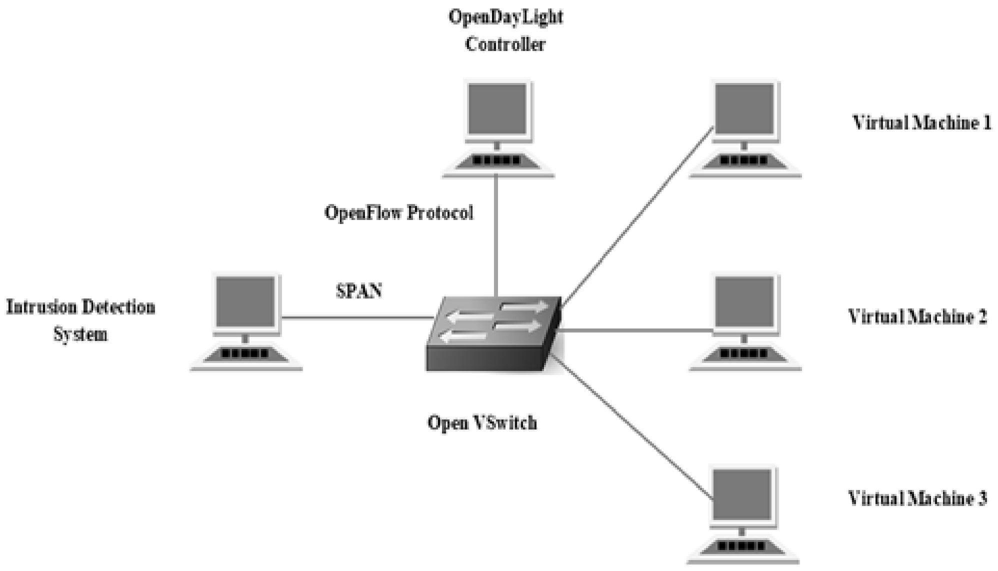


Figure 5. Network topology of proposed system

A simple network topology is set up on a Virtual Box virtualization environment consisting of four virtual machines that are connected by the virtual switch called Open VSwitch. The Open VSwitch is controlled by a logically centralized controller called the OpenDayLight Controller. Figure 5 depicts the network topology of the proposed system. There are four Virtual Machines out of which three are hosts and one is used to run Snort. Metasploit framework is an open-source tool which helps to create exploitation on a particular chosen remote machine by entering it through a bug of the target machine. To simulate attack scenarios, Metasploit is installed on one of the three pure hosts.

The SDN Controller used is OpenDayLight which is the brain of the architecture also called the Control Plane, which is one of the layers of the SDN architecture. OpenDayLight is used to program the Software Defined Networking Controller. OpenDayLight is the controller used widely and is an open-source controller. It can automate as well as customize computer networks of variable size and scale. Switched Port Analyser or Port Mirroring is used to monitor traffic in a network. By enabling SPAN, a switch can send a copy of all packets arriving at a particular port to another port wherein the packets can be used to detect malicious contents.

SPAN Mirroring is configured on the Open VSwitch which has two output ports, one to divert the traffic destined to any host, and the second to the Virtual Machine that is running Snort which can examine the same. The incoming traffic reaches on OpenFlow enabled switch which is equipped with a rule in the forwarding table. The proposed idea is to create an Intrusion Detection System on Software Defined Networking environment using virtual machines.

Received: 19 Sept 2019 | Revised: 20 Oct 2019 | Accepted: 20 Nov 2019

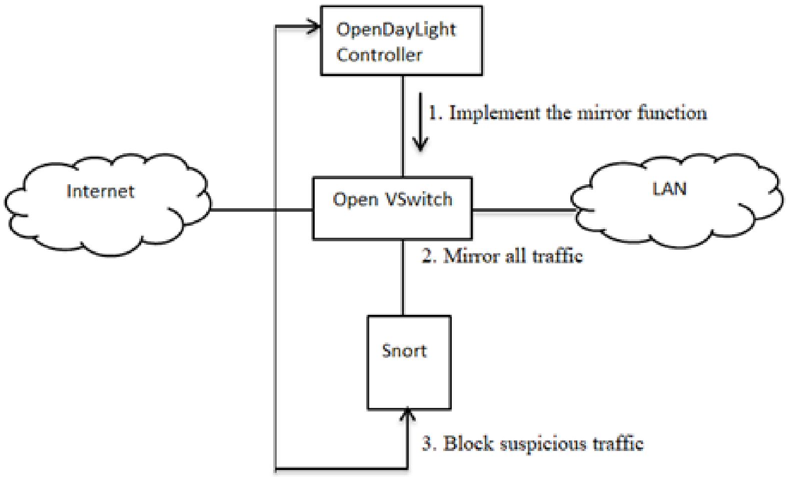


Figure 6. Software Defined Networking with Snort in detail

Algorithm

1. Set rules for Snort.
2. Configure the network with the hosts, detection system and controller.
3. Receive the mirrored packets from switch.
4. If suspicious packet flow found,

Then check for anomaly from usual flow.

Send reports of flow of suspicious packets to Intrusion Detection System.

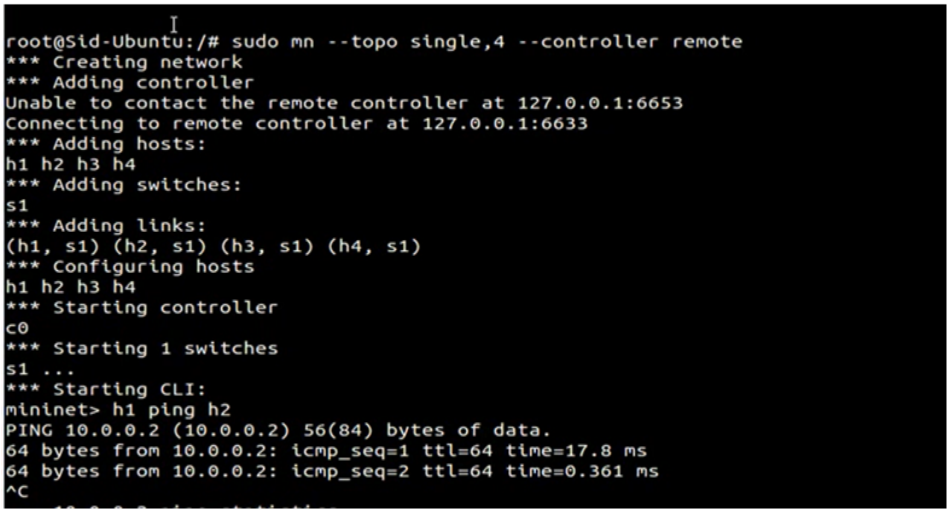
Identify the type of rule violation by checking with predefined rules.

Block the suspicious flow(attack)

Else

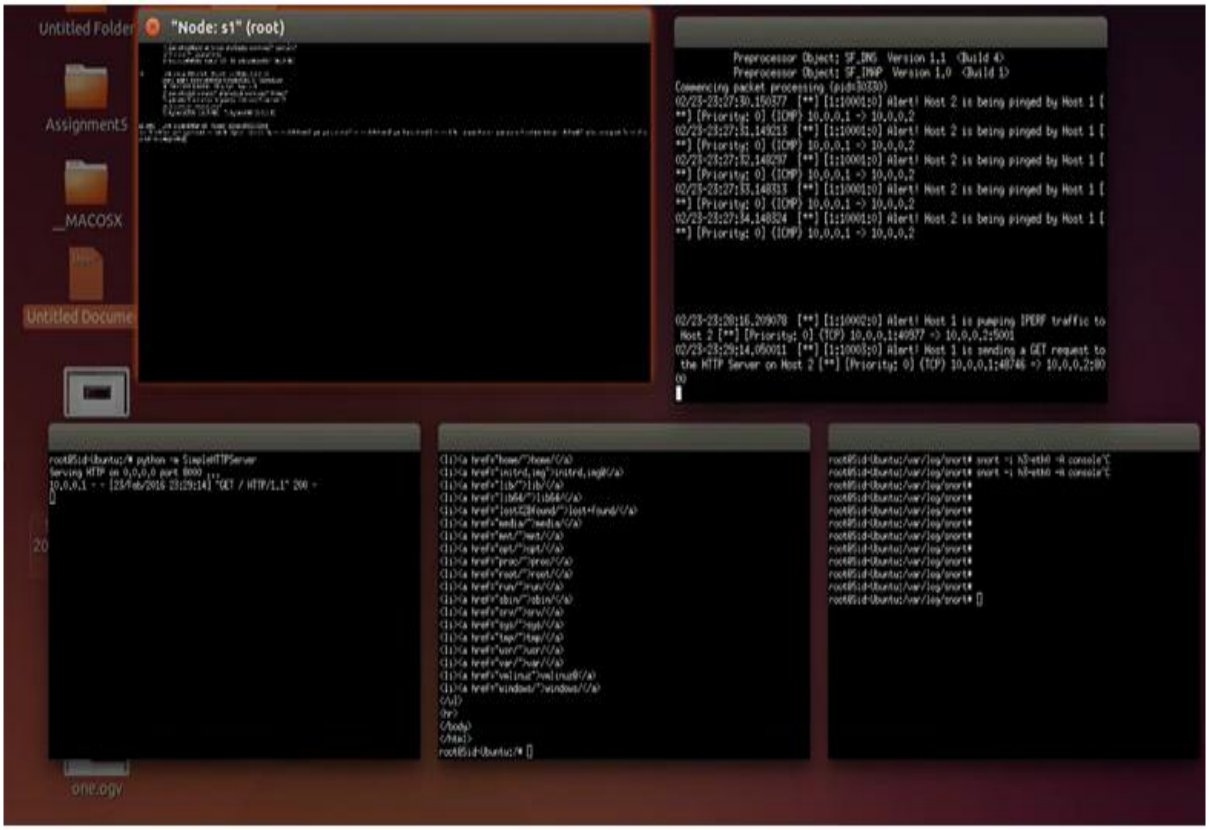
Go to step 3, proceed with usual monitoring of packet flow.

Implementation Results



```
root@Sid-Ubuntu:/# sudo mn --topo single,4 --controller remote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Connecting to remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1
*** Adding links:
(h1, s1) (h2, s1) (h3, s1) (h4, s1)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 1 switches
s1 ...
*** Starting CLI:
mininet> h1 ping h2
PING 10.0.0.2 (10.0.0.2) 56(84) bytes of data.
64 bytes from 10.0.0.2: icmp_seq=1 ttl=64 time=17.8 ms
64 bytes from 10.0.0.2: icmp_seq=2 ttl=64 time=0.361 ms
^C
```

Figure 7. Configuring the network topology



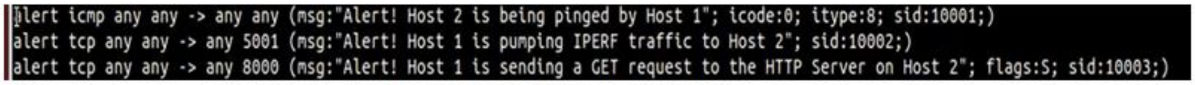
```
        Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
        Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
Commencing packet processing (pid=30330)
02/23-23:27:30.150377  [**] [1:10001:0] Alert! Host 2 is being pinged by Host 1 [
**] [Priority: 0] {ICMP} 10.0.0.1 -> 10.0.0.2
02/23-23:27:31.149213  [**] [1:10001:0] Alert! Host 2 is being pinged by Host 1 [
**] [Priority: 0] {ICMP} 10.0.0.1 -> 10.0.0.2
02/23-23:27:32.148297  [**] [1:10001:0] Alert! Host 2 is being pinged by Host 1 [
**] [Priority: 0] {ICMP} 10.0.0.1 -> 10.0.0.2
02/23-23:27:33.148313  [**] [1:10001:0] Alert! Host 2 is being pinged by Host 1 [
**] [Priority: 0] {ICMP} 10.0.0.1 -> 10.0.0.2
02/23-23:27:34.148324  [**] [1:10001:0] Alert! Host 2 is being pinged by Host 1 [
**] [Priority: 0] {ICMP} 10.0.0.1 -> 10.0.0.2




02/23-23:28:16.209078  [**] [1:10002:0] Alert! Host 1 is pumping IPERF traffic to
 Host 2 [**] [Priority: 0] {TCP} 10.0.0.1:40977 -> 10.0.0.2:5001
02/23-23:29:14.050011  [**] [1:10003:0] Alert! Host 1 is sending a GET request to
 the HTTP Server on Host 2 [**] [Priority: 0] {TCP} 10.0.0.1:48746 -> 10.0.0.2:80
00
```

```
root@Sid-Ubuntu:/# python -m SimpleHTTPServer
Serving HTTP on 0.0.0.0 port 8000 ...
10.0.0.1 - - [23/Feb/2016 23:29:14] "GET / HTTP/1.1" 200 -
```

```
<li><a href="home/">home/</a>
<li><a href="initrd.img">initrd.img@</a>
<li><a href="lib/">lib/</a>
<li><a href="lib64/">lib64/</a>
<li><a href="lost%2Bfound/">lost+found/</a>
<li><a href="media/">media/</a>
<li><a href="mnt/">mnt/</a>
<li><a href="opt/">opt/</a>
<li><a href="proc/">proc/</a>
<li><a href="root/">root/</a>
<li><a href="run/">run/</a>
<li><a href="sbin/">sbin/</a>
<li><a href="srv/">srv/</a>
<li><a href="sys/">sys/</a>
<li><a href="tmp/">tmp/</a>
<li><a href="usr/">usr/</a>
<li><a href="var/">var/</a>
<li><a href="vmlinuz">vmlinuz@</a>
<li><a href="windows/">windows/</a>
</ul>
<hr>
</body>
</html>
root@Sid-Ubuntu:/# 
```

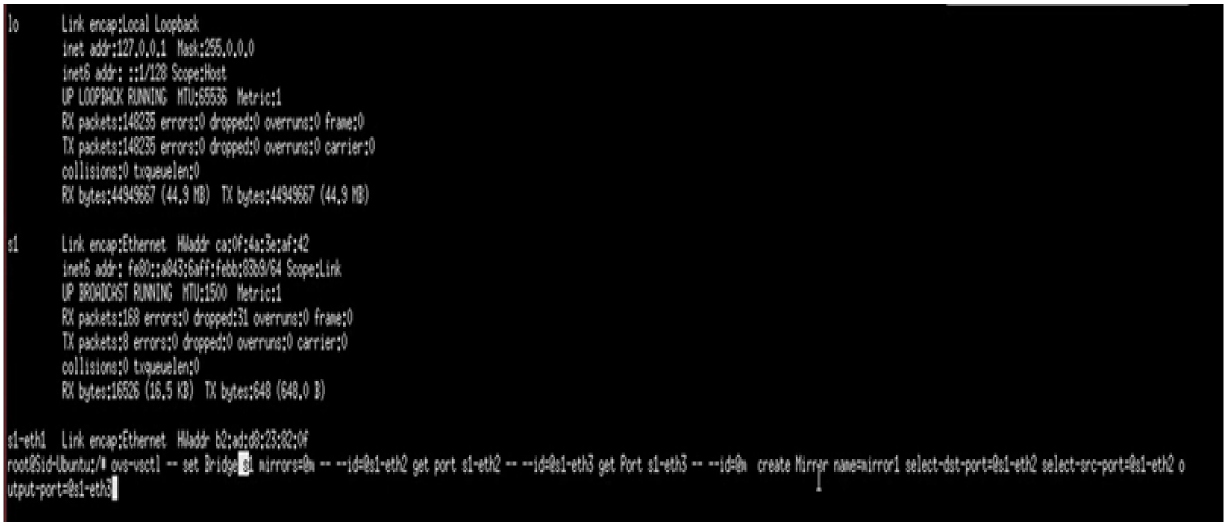
```
root@Sid-Ubuntu:/var/log/snort# snort -i h3-eth0 -A console^C
root@Sid-Ubuntu:/var/log/snort# snort -i h3-eth0 -A console^C
root@Sid-Ubuntu:/var/log/snort#
root@Sid-Ubuntu:/var/log/snort#
root@Sid-Ubuntu:/var/log/snort#
root@Sid-Ubuntu:/var/log/snort#
root@Sid-Ubuntu:/var/log/snort#
root@Sid-Ubuntu:/var/log/snort#
root@Sid-Ubuntu:/var/log/snort#
root@Sid-Ubuntu:/var/log/snort#
root@Sid-Ubuntu:/var/log/snort#
root@Sid-Ubuntu:/var/log/snort# 
```

Figure 8 Terminal windows for the OpenVSwitch and hosts



```
alert icmp any any -> any any (msg:"Alert! Host 2 is being pinged by Host 1"; icode:0; itype:8; sid:10001;)
alert tcp any any -> any 5001 (msg:"Alert! Host 1 is pumping IPERF traffic to Host 2"; sid:10002;)
alert tcp any any -> any 8000 (msg:"Alert! Host 1 is sending a GET request to the HTTP Server on Host 2"; flags:S; sid:10003;)
```

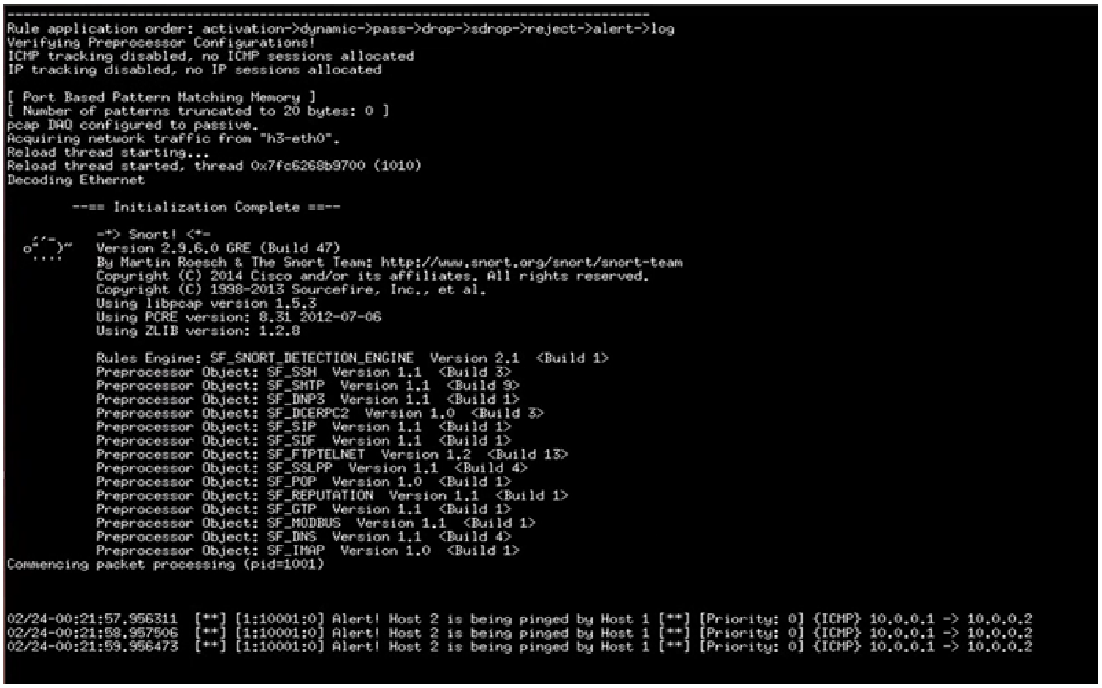
Figure 9 Setting sample snort rules



```
lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:148235 errors:0 dropped:0 overruns:0 frame:0
          TX packets:148235 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:44949667 (44.9 MB)  TX bytes:44949667 (44.9 MB)

s1        Link encap:Ethernet  HWaddr ca:0f:4a:3e:af:42
          inet6 addr: fe80::a843:6aff:febb:83b9/64 Scope:Link
          UP BROADCAST RUNNING  MTU:1500  Metric:1
          RX packets:168 errors:0 dropped:31 overruns:0 frame:0
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:16526 (16.5 KB)  TX bytes:648 (648.0 B)

s1-eth1   Link encap:Ethernet  HWaddr b2:ad:d8:23:82:0f
root@Sid-Ubuntu:/# ovs-vsctl -- set Bridge s1 mirrors=@m -- --id=@s1-eth2 get port s1-eth2 -- --id=@s1-eth3 get Port s1-eth3 -- --id=@m  create Mirror name=mirror1 select-dst-port=@s1-eth2 select-src-port=@s1-eth2 o
utput-port=@s1-eth3
```

Figure 10 Mirroring from OpenVSwitch

Received: 19 Sept 2019 | Revised: 20 Oct 2019 | Accepted: 20 Nov 2019



```
--------------------------------------------------------------------------------
Rule application order: activation->dynamic->pass->drop->sdrop->reject->alert->log
Verifying Preprocessor Configurations!
ICMP tracking disabled, no ICMP sessions allocated
IP tracking disabled, no IP sessions allocated

[ Port Based Pattern Matching Memory ]
[ Number of patterns truncated to 20 bytes: 0 ]
pcap DAQ configured to passive.
Acquiring network traffic from "h3-eth0".
Reload thread starting...
Reload thread started, thread 0x7fc6268b9700 (1010)
Decoding Ethernet

        --== Initialization Complete ==--

   ,,_     -*> Snort! <*-
  o"  )~   Version 2.9.6.0 GRE (Build 47)
   ''''    By Martin Roesch & The Snort Team: http://www.snort.org/snort/snort-team
           Copyright (C) 2014 Cisco and/or its affiliates. All rights reserved.
           Copyright (C) 1998-2013 Sourcefire, Inc., et al.
           Using libpcap version 1.5.3
           Using PCRE version: 8.31 2012-07-06
           Using ZLIB version: 1.2.8

           Rules Engine: SF_SNORT_DETECTION_ENGINE  Version 2.1  <Build 1>
           Preprocessor Object: SF_SSH  Version 1.1  <Build 3>
           Preprocessor Object: SF_SMTP  Version 1.1  <Build 9>
           Preprocessor Object: SF_DNP3  Version 1.1  <Build 1>
           Preprocessor Object: SF_DCERPC2  Version 1.0  <Build 3>
           Preprocessor Object: SF_SIP  Version 1.1  <Build 1>
           Preprocessor Object: SF_SDF  Version 1.1  <Build 1>
           Preprocessor Object: SF_FTPTELNET  Version 1.2  <Build 13>
           Preprocessor Object: SF_SSLPP  Version 1.1  <Build 4>
           Preprocessor Object: SF_POP  Version 1.0  <Build 1>
           Preprocessor Object: SF_REPUTATION  Version 1.1  <Build 1>
           Preprocessor Object: SF_GTP  Version 1.1  <Build 1>
           Preprocessor Object: SF_MODBUS  Version 1.1  <Build 1>
           Preprocessor Object: SF_DNS  Version 1.1  <Build 4>
           Preprocessor Object: SF_IMAP  Version 1.0  <Build 1>
Commencing packet processing (pid=1001)


02/24-00:21:57.956311  [**] [1:10001:0] Alert! Host 2 is being pinged by Host 1 [**] [Priority: 0] {ICMP} 10.0.0.1 -> 10.0.0.2
02/24-00:21:58.957506  [**] [1:10001:0] Alert! Host 2 is being pinged by Host 1 [**] [Priority: 0] {ICMP} 10.0.0.1 -> 10.0.0.2
02/24-00:21:59.956473  [**] [1:10001:0] Alert! Host 2 is being pinged by Host 1 [**] [Priority: 0] {ICMP} 10.0.0.1 -> 10.0.0.2
```

Figure 11 Alert by Snort in accordance with the specified rules

The implementation of the proposed concept is shown in figure 7, figure 8, figure 9, figure 10 and figure 11. One of the rules set in Snort is to alert when Host 1 pings Host 2. So when a ping command is given to ping host 2, the alert message is being displayed in the Snort console or terminal. Similarly, it can alert when the other rules are violated. So, this concept can be implemented to protect the Software Defined Networking systems from various intruders and their attacks.

VI. CONCLUSION

The idea being proposed in this paper is to develop an Intrusion Detection System in Software Defined Networking using Snort tool and Mirroring concept to effectively provide a defense against malicious activities taking place in a network. According to recent researches, the most occurring attacks are Denial of Service attacks and Distributed Denial of Service attacks. As these attacks bound to increase day by day, an effective method is to be implemented to provide security from these attacks. So, the concept proposed by this paper tends to provide security by checking the deviations of the flow from the usual flows in the network with some predefined set of rules specified in the Snort principles. Detection of any malicious activities will result in the OpenDayLight controller block the current activity taking place in the network to avoid occurrence of attack. Also, the rules in Snort application can be altered according to the needs of the network by the authorized network administrator.

REFERENCES

[1] P.W. Chi, C.T Kuo, H.M. Ruan, S.J. Chen, and C.L. Lei, 2014, November. An AMI threat detection mechanism based on SDN networks. *In Proc. SECURWARE* Vol 1 pp. 208-211.

[2] T. Xing, Z. Xiong, D. Huang, and D. Medhi, 2014, November. SDNIPS: Enabling Software-Defined Networking based intrusion prevention system in clouds. *In Network and Service Management (CNSM), 2014 10th International Conference on* pp. 308-311 IEEE.

[3] R. Shreshthi, P. Kapre, K. Shekatkar, M. Kalgane, Y. Hande, 2014, Intrusion Detection System based Software Defined Networking. *International Journal of Advance Engineering and Research Development,* 2014 January Volume 4, Issue 1.

[4] Lara, and B. Ramamurthy. "OpenSec: Policy-based security using software-defined networking." *IEEE Transactions on Network and Service Management* 13, Vol no. 1 (2016): pg. 30-42.

[5] D. Jankowski, and M. Amanowicz. "Intrusion detection in software defined networks with self-organized maps." *Journal of Telecommunications and Information Technology* (2015).

[6] H. Xu , C. Wang and H. Chen. "An Extension Approach for Threat Detection and Defense of Software-Defined Networking." *International Journal of Security and Its Applications* 10, Vol no. 2 (2016): 365-374.

[7] H. Hu, W. Han, G. Ahn, and Z. Zhao. "FLOWGUARD: building robust firewalls for software-defined networks." *In Proceedings of the third workshop on Hot topics in software defined networking,* pp. 97-102. ACM, 2014.

[8] J. Sadhasivam, M. Kubendiran, P. Tomy, B. Jeyakumar, M. Sathish Kumar, and R. Anusha, “Review of Gaming and Its Evolution Over Networks,” *Int. J. Civ. Eng. Technol.,* vol. 8, no. 11, pp. 61–68, 2017.

[9] S. Nithya, M. Asha Jerlin, R. Charanya, S. Jayakumar, and R. Rathi, “Self Restorative Cluster Head Selection In Heterogeneous Network,” *Glob. J. Pure Appl. Math.,* vol. 11, no. 3, pp. 1655–1662, 2015.

[10] R. C, asha J. M, J. Sadhasivam, N. S, and R. Rohit, “A Case Study on Attack Models And Privacy Models In Mining Medical Datasets,” *Int. J. Mech. Eng. Technol.,* vol. 8, no. 11, pp. 964–976, 2017.

[11] J. Sadhasivam, S. Jayavel, B. Jeyakumar, and S. Merchant, “HOCS : Host Os Communication Service Layer,” *Int. J. Civ. Eng. Technol.,* vol. 8, no. 11, pp. 35–41, 2017.

[12] S. Jayakumar, S. Jayavel, and M. Senthilkumar, “Network Security – MAC Address Block,” *Int. Conf. Netw. Commun. Comput.,* pp. 419–422, 2011.

[13] S. Jayakumar, S. Jayavel, and N. S, “Automatic Campus Network Management using GPS,” *Int. J. Comput. Sci.* Issues, vol. 9, no. 3, pp. 468–472, 2012.

[14] J. Sadhasivam, ashaJerlin M, and N. S, “Intelligent Interior Mapping using Wall Following Behaviour,” *Int. J. Trend Res. Dev.,* vol. 3, no. 6, p. 112, 2016.

[15] J. Sadhasivam, R. Charanya, S. Harish Kumar, and A. Srinivasan, “Identifying images of handwritten digits using deep learning in H2O,” *IOP Conf. Ser. Mater. Sci. Eng.,* vol. 263, p. 042033, 2017.

[16] J. Sadhasivam, M. Alamelu, R. Radhika, S. Ramya, K. Dharani, and S. Jayavel, “Enhanced way of securing automated teller machine to track the misusers using secure monitor tracking analysis,” *IOP Conf. Ser. Mater. Sci. Eng.,* vol. 263, p. 042032, 2017.

[17] Aarthi, S.,& Vijay, N. (2014). Sophisticated Data Entry Application using Matchmaking Algorithm through Scanned Images. *International Journal of System Design and Information Processing,* 2(1), 27-29.

[18] Patidar, H.P., & Sharma,N. (2016). Adaptive Approach of DSR and OLSR Routing Protocols Using Optimal Probabilistic Logical Key Hierarchy in MANET. *Bonfring International Journal of Networking Technologies and Applications,* 3(2), 13-20.

[19] Venkateswara Rao, B., and Nagesh Kumar, G.V. (2014). Voltage Collapse Proximity Indicator based Placement and Sizing of Static VAR Compensator using BAT Algorithm to Improve Power System Performance. *Bonfring International Journal of Power Systems and Integrated Circuits,* 4(3), 31-38.

[20] Neenu Preetam, I., & Gupta, H. (2014). Cardless Cash Access using Biometric ATM Security System. *International Scientific Journal on Science Engineering & Technology,* 17(10), 893-897.

[21] Revathi, M., Prakash, K., &Suguna, R. (2018). A Systematic Study on Cyber Physical System. *Bonfring International Journal of Research in Communication Engineering,* 8(1), 1-4.

[22] Prabhu, N., Agilan, S., Muthukumarasamy, N., &Senthil, T. S. (2014). Thermal Investigations of Aluminum Doped WO3 Nanoparticles by Solvo Thermal Cum Chemical Method. *Journal of Ovonic Research* Vol. 10(5), 167-174.

[23] Reka M., & Shanthi N. (2014). An efficient semantic-link classifier for web document clustering. *International Journal of Applied Engineering Research* 9(23):22997-23012.

[24] Ramkumar R.P., &Arumugam S. (2014). Improved iris segmentation algorithm without iris localization phase. International Journal of Applied Engineering Research. 9(23):22291-22300.

[25] Mohankumar G.B., Balachandran M., &Raj Kumar S. (2014). *International Journal of Applied Engineering Research.* 9(23):20211-20224.

[26] Vanathi D., &Sengottuvelan P. (2014). Graph based anonymization technique for privacy preserving data publication using attribute probability matrix and semantic ontology. *International Journal of Applied Engineering Research* 9(23):20833-20841.

[27] Praveenkumar B.J., &Pradeepkumar K. (2014). Natural fiber composite using epoxy reinforcement: A review. *International Journal of Applied Engineering Research* 9(23):22757-22762.

[28] Satheesh A., &Jeyageetha V. (2014). Improving power system stability with facts controller using certain intelligent techniques. *International Journal of Applied Engineering Research,* 9(23):21893-21910.

[29] Muthukumar, M., Karthikeyan, P., Vairavel, M., Loganathan, C., Praveenkumar, S., & Kumar, A. S. (2014). Numerical studies on PEM fuel cell with different landing to channel width of flow channel. *Procedia Engineering,* 97, 1534-1542

[30] Umamaheswari R., Shanthi N.(2015). An efficient hybrid information retrieval approach for unstructured document classification. *International Journal of Applied Engineering Research,* 10(24).

[31] RAJAN C., SHANTHI N.(2015). Genetic based optimization for multicast routing algorithm for MANET. *Sadhana - Academy Proceedings in Engineering Sciences,* 40(8).

[32] Prabhadevi S., Javavel S., Kapoor R.(2015). Algorithm of sentiment analysis for computing machines. *Journal of Scientific and Industrial Research,* 74(12).

[33] Palanisamy T., Krishnasamy K.N.(2015). Bayes node energy polynomial distribution to improve routing in wireless sensor network. *PLoS ONE,* 10(10).

[34] Prasath A., Satheesh A.(2015). Implementation of real time data acquisition system with ARM. *ICIIECS 2015 - 2015 IEEE International Conference on Innovations in Information, Embedded and Communication Systems.*

[35] Balakrishnan M., Gowthaman S., Jaya Kumaran S.P., Sabhapathy G.R.(2015). A smart spy robot charged and controlled by wireless systems. *ICIIECS 2015 - 2015 IEEE International Conference on Innovations in Information, Embedded and Communication Systems.*

[36] Sasikala E., Rengarajan N.(2015). An Intelligent Technique to Detect Jamming Attack in Wireless Sensor Networks (WSNs). *International Journal of Fuzzy Systems,* 17(1).

[37] Gopinath B., Shanthi N.(2015). Development of an automated medical diagnosis system for classifying thyroid tumor cells using multiple classifier fusion. *Technology in Cancer Research and Treatment,* 14(5).

[38] Bennet A., Sankaranarayanan, Kandasamy R., Aruna, Kavitha, Thamizhoviya.(2016). Performance evaluation of enhancing the capacity of spectrum sharing cognitive radio networks. *IIOAB Journal,* 7(9).

[39] Kumar P.R., Santhakumar K., Palani S. (2016). An intelligent approach for optimizing Energy consumption and Schedule length of Embedded multiprocessors <sup>1</sup>. *Journal of Intelligent and Fuzzy Systems,* 31(1).

[40] Arthy G., Marimuthu C.N. (2016). Immune RBF neural network algorithm for DSTATCOM. 2016 *International Conference on Computer Communication and Informatics, ICCCI* 2016.

[41] Somasundaram K., Saritha S., Ramesh K. (2016). Enhancement of network lifetime by improving the leach protocol for large scale WSN. *Indian Journal of Science and Technology,* 9(16).

[42] Sukumar P., Gnanamurthy R.K. (2016). Computer aided detection of cervical cancer using pap smear images based on adaptive neuro fuzzy inference system classifier. *Journal of Medical Imaging and Health Informatics,* 6(2).

[43] Thangavelu S.K., Kasthuri N., Sundaram V., Aravind N., Bilakanti N. (2016). A stand-alone EEG monitoring system for remote diagnosis. *Telemedicine and e-Health,* 22(4).

[44] Sengottaian S., Natesan S., Mathivanan S. (2017). Weighted delta factor cluster ensemble algorithm for categorical data clustering in data mining. *International Arab Journal of Information Technology,* 14(3).

[45] Upendra Roy B.P., Rengarajan N. (2017). Feasibility Study of an Energy Storage System for Distributed Generation System in Islanding Mode. *Journal of Energy Resources Technology,* 139(1).

[46] Arthy G., Marimuthu C.N. (2018). A novel center-tapped transformer based multilevel inverter with common DC source. *Journal of Vibro engineering,* 20(8).

[47] Jagan K., Sivasankaran S., Bhuvaneswari M., Rajan S. (2018). Effect of second order slip and non-linear thermal radiation on mixed convection flow of MHD Jeffrey nanofluid with double stratification under convective boundary condition. *IOP Conference Series: Materials Science and Engineering,* 390(1).

[48] Khed V.C., Mohammed B.S., Liew M.S., Alaloul W.S., Adamu M., Al-Fakih A., Karthikeyan J. (2018). Experimental investigation on pullout strength of hybrid reinforcement of fibre in rubberized cementitious composites. *International Journal of Civil Engineering and Technology,* 9(7).

[49] Deepa A., Marimuthu C.N. (2018). Modified RS encoder architecture with reduced critical path delay for high speed data communication. *Proceedings of the International Conference on Intelligent Sustainable Systems, ICISS* 2017.

[50] Vijayakumar, J., &Arumugam, D. S. (2012). Study of betelvine plants diseases and methods of disease identification using digital image processing. *European Journal of Scientific Research*, *70*(2), 240-244.

[51] Satheesh, A., &Manigandan, T. (2012). Improving power system stability using PSO and NN with the aid of FACTS controller. *European Journal of Scientific Research*, *71*(2), 255-264.

[52] Rameshkumar, A., &Arumugam, S. (2012, April). PI Control of Quasi-resonant Buck Converter. In *International Conference on Advances in Information Technology and Mobile Communication* (pp. 477-485). Springer, Berlin, Heidelberg.

[53] Suresh, Y., Arumugam, S., &Bhagyaveni, M. A. (2012). A Forager Bee's Intelligence Inspired Dynamic Queue Scheduling for the Internet Traffic. *Journal of Computer Science*, *8*(5), 665.

[54] Ramesh, K., &Somasundaram, K. (2012, January). Optimized FZ-LEACH Using Exponential Weighted Moving Average for Wireless Sensor Network. In *International Conference on Computer Science and Information Technology* (pp. 473-481). Springer, Berlin, Heidelberg.

[55] Satheesh,A., &ManigandanT. (2013). Maintaining Power System Stability with Facts Controller Using Bees Algorithm and NN.Journal of Theoretical and Applied Information Technology, 49 (1), 38-47.

[56] Vijayalakshmi M., &Subramanian P.Synthesis, spectral characterization, biological activity and DNA cleavage studies of Cu(II), Ni(II) and Zn(II) schiff base complexes derived from 2,4-dihydroxy benzaldehyde and P-aminophenol. *International Journal of Pharmacy and Technology,* 5(1):5144-5155.