Published by :
http://www.ijert.org

International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Volume 13, Issue 03 March 2024

Developing a Packet Sniffing Tool in Adverse Environment

R.Suhasini, Reddyvari Venkateswara Reddy, Sridhara Srija, Miriyala Sai Teja, Yeturi Sri Priya,
Assistant Professor, Department of CSE (Cyber Security), CMR College of Engineering & Technology, Hyderabad, Telangana, India
Associate Professor, Department of CSE (Cyber Security), CMR College of Engineering & Technology, Hyderabad, Telangana, India
B.Tech Student, Department of CSE (Cyber Security), CMR College of Engineering & Technology Hyderabad, Telangana, India

***Abstract*— Packet sniffers are network analyzers, play a pivotal role in monitoring and validating internet traffic within organizations. These tools allow admins to control, observe, and record data packets as they traverse a network. However, their use extends beyond legitimate purposes; malicious actors also employ sniffers to obtain the contents of sensitive information. In this research, we introduce, a novel packet sniffing tool designed for robust performance even in adverse environments. our primary goals include Protocol Analysis: it decodes and analyzes captured packets, providing the benefits of network operations. Encoded Packet Database: The tool constructs an encoded packet database, enhancing efficiency and scalability. User-Friendly Interface: front end presents relevant data for the user requests. By addressing challenges related to stealthy packet sniffing, it contributes to network security, management, and troubleshooting.**

***Keywords*:**
Network, monitoring, analysis, ports, IP Address, protocols, network traffic,.

I. INTRODUCTION

In the present-day highly interconnected world, understanding network traffic is of utmost importance in many domains, including both research and development on network security and more generally, such as software architecture. Packet sniffing tools can reveal a substantial amount of network behavior, but it is crucial to ensure their ethical, transparent use, and comply with regulations. This introduction focuses on the development of tools within these ethical boundaries, with emphasis on: 1. Elucidate the intended utilizing the instrument and justify its compliance with ethical and legal principles. Provide a clear explanation of this purpose. Point out its role in generating benefits, like security assessments or network research. 'transparency and consent: Get the right permissions, clearly tell everyone affected how you use this tool. Assess the influence on privacy and data protection rights. Trim the edge: Develop the software to gather only the information that is strictly necessary for its desired outcome. Refrain from taking pictures of personal information that is not within the bounds of your target. secure and protect data: Establish robust security on unauthorized access or misuse of captured data. Comply using the information protection regulations and ideal procedures. Responsible Reporting and Disclosure: Recommend outcomes in a responsible and ethical manner, while avoiding activities which harms individuals or systems. Consider the possible outcomes of your disclosures. The packet sniffing tools may harm to positive outcomes and promote ethical and responsible practices based on the above principles. The approach fosters trust, transparency and ultimately creates a safer and more beneficial online environment for everyone involved.
Network tools which involves to capture the network and to access the network packets which is used by offensive and also by defensive.

The various kinds of packet sniffers which are found in open- source are:

1. MAC Sniffers: These focus on data related to MAC address filtering.
2. Protocol Sniffers: They examine network protocols.
3. LAN Sniffers: Mainly used in systems or networks they check an assortment of IP addresses.
4. IP Sniffers: Gather data pertaining to IP filters for analysis and troubleshooting.
5. ARP Sniffers: Map IP addresses to MAC addresses enabling packet spoofing and other attacks.
6. Password Sniffers: Extract information, from network traffic to collect passwords.

The softwares which are available for the capturing packets are :

1. Wireshark: A popular free and open-source graphical user interface software that enables examination of network traffic.
2. TCP Dump: A tool used via command line for capturing and studying packets.
Additionally there are custom tools available.

II. LITERATURE REVIEW

In this study to obtain optimal output research, literature review conducted related previous studies. In research, it serves as a reference. Numerous studies have been conducted that carries out by previous researchers, like [3], they discussed packet analysis and network traffic monitoring over TCP protocol used Wire Shark packet sniffer. The researchers examined graphs showing TCP time sequences, TCP Throughput and TCP round trip times to analyze network traffic data. They put forward suggestions, for managing network traffic based on their analysis. Similarly, [5] proposed a new method of monitoring systems. It can provide detailed information based on traffic behavior methods and a history of connected traffic. While monitoring internet traffic data, for analysis, Qadeer and colleagues (2017) created a sniffer tool on the Linux platform to enhance Intrusion Detection capabilities. The primary objective of this study is to examine network bottleneck situations and subsequently identify and manage software presence efficiently. Additionally, Lizarti et al. (2018) delve into traffic analysis within Virtual Private Networks (VPNs) using (SNMP). Their research focuses on real time traffic reporting by network traffic applications based on TCP and UDP ports accessible privately through VPN technology. This enables network administrators to monitor and troubleshoot network issues effectively from, within the system. Armaan (2021) illustrated and tested the precision of various models. Every digital application relies on data to function properly. Safeguarding data is crucial, due to cyber threats. Developing models involves challenging feature selection. Machine learning offers a cutting edge solution for predictions. This approach requires flexibility to handle data types effectively. IT security professionals utilize malware analysis tools to identify patterns and assess the severity of software. These technologies play a role, in enhancing cybersecurity by monitoring security alerts and preventing malware attacks. Prompt removal of malware is essential to prevent spread of infection. Malware analysis is gaining popularity because businesses mitigate the result of escalating malware threats and the evolving tactics used in cyberattacks.

Keeping an eye on a network can equip a admins with insights to proactively oversee the network and share network usage data with others. Link activity, error rates and link status are, among the factors that aid a admins in assessing the well being and utilization of a network. By gathering and verifying the data over time a network administrator can observe trends anticipate growth and potentially identify and replace a failing component before it causes issues. SNMP is widely used for collecting device details. Administrators use the SNMP to manage network devices such, as servers, workstations, routers, switches and security devices, within an IP network. It allows network administrators to monitor performance, troubleshoot issues and strategize for expansion.

SNMP is an protocol which provides the components and the communication between the server and the client. It consists among the subsequent components SNMP manager, SNMP agents (managed node) and Management Information Base (MIB)[9].

Packet sniffing is tools uses for as monitoring data packet when a packet crosses a network. There are packet sniffing available as both software and hardware, yet there are additionally hardware-based devices that are installed directly along the network. Sniffer manages the data that is directed to it. System administrators are allowed to use Sniffer on their networks to observe and address traffic issues. For instance, if communication glitch exists between two computers administrators can identify and monitor packets sent over a network to the other to identify the root of the problem. The components of a sniffer include;

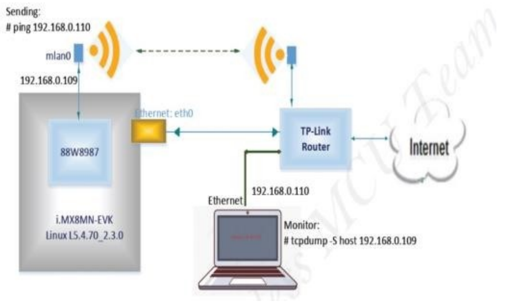
Hardware : standard network adapters .

Capture Filter : It captures the network traffic from the wire filters it for data. Then saves it in a storage unit. Buffers are employed to retain the captured frames by the Capture Filter. A real time analyzer, found within the packet sniffer software is utilized for examining traffic patterns and detecting intrusions. The decoder is responsible, for Protocol Analysis. Several tools used for packet sniffing include Wireshark, Kismet, Tcpdump, Cain and Abel, Ettercap, Dsniff, NetStumbler, Ntop, Ngrep, Etherape and KisMAC. Wireshark is a network packet analyzer. A network packet analysis will capture network packets and display data packets as detailed as possible. The user might view network analysis as a tool, for monitoring activities within a network cable to how an electrician uses a voltmeter to inspect an electrical cable. In times tools, like these were often costly limited in availability or even both. But with the appearance of the Wireshark all that has changed.

III. EXISTING SOLUTIONS:

1. TCP DUMP:

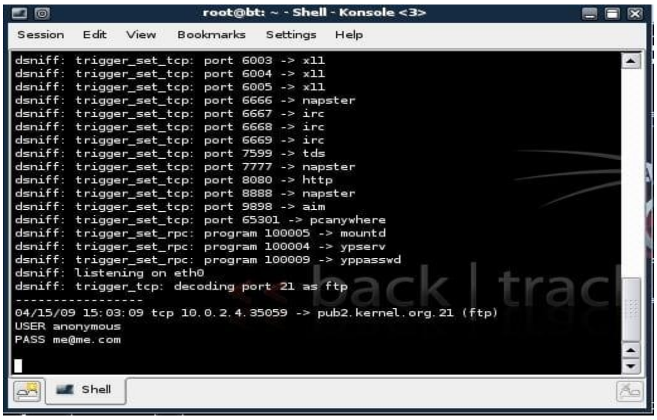
TCP Dump is a CLI analyzer that allows users to intercept and monitor TCP/IP and various packets due to the fact that they are sent over a network. It captures all network traffic on designated networks using LIBCAP. Displays it directly, on your screen.



```
# ping 192.168.0.110
# tcpdump -S host 192.168.0.109
```

Fig 1. TCP DUMP

2. DSNIFF :
DSNIFF is a tool which created to sniff packets distinguishing between protocols, aimed at intercepting and uncovering passwords. Additionally, the DSNIFF tool is tailored for Unix and Linux systems. Lacks a counterpart, on Windows platforms for assistance.



```
dsniff: trigger_set_tcp: port 6003 -> x11
dsniff: trigger_set_tcp: port 6004 -> x11
dsniff: trigger_set_tcp: port 6005 -> x11
dsniff: trigger_set_tcp: port 6666 -> napster
dsniff: trigger_set_tcp: port 6667 -> irc
dsniff: trigger_set_tcp: port 6668 -> irc
dsniff: trigger_set_tcp: port 6669 -> irc
dsniff: trigger_set_tcp: port 7599 -> tds
dsniff: trigger_set_tcp: port 7777 -> napster
dsniff: trigger_set_tcp: port 8080 -> http
dsniff: trigger_set_tcp: port 8888 -> napster
dsniff: trigger_set_tcp: port 9898 -> aim
dsniff: trigger_set_tcp: port 65301 -> pcanywhere
dsniff: trigger_set_rpc: program 100005 -> mountd
dsniff: trigger_set_rpc: program 100004 -> ypserv
dsniff: trigger_set_rpc: program 100009 -> yppasswd
dsniff: listening on eth0
dsniff: trigger_tcp: decoding port 21 as ftp
-----------------
04/15/09 15:03:09 tcp 10.0.2.4.35059 -> pub2.kernel.org.21 (ftp)
USER anonymous
PASS me@me.com
```

Fig 2. DSNIFF

IV. OBJECTIVE

The main purpose of this research is to develop and evaluate a packet sniffing tool which utilized at any location for offensive and defensive purposes like as we are the people who are capable of understanding the situations which are being created by the attackers. The attackers are using the knowledge of hacking to the illiterates and they're reaping the rewards. So we are developing a packet sniffing tool which can be able to sniff the data and monitor the traffic which is existing inside the computer.

V. PROBLEM DEFINITION

The problem addressed in this study is to create a basic packet sniffing tool which is used in adverse environment and may be able to capture the network traffic. To give awareness for a basic networking person on how the protocols and ports are getting opened and how the attacker is misusing them.

VI. PROPOSED METHODOLOGY

Our project is to implement a basic packet sniffing tool which will be able to produce the network traffic and may be able to sniff the data packets by using a powerful tool called scapy and is able to capture the life network traffic. It will become accustomed to the computer or system vulnerabilities by a user or in an organization. In organizations, the network access won't be given to any one of the user and able to be observed by the by the administrators. So, to know about the vulnerabilities as an cyber security person, you should be able to know the details of the ports which are opened and can't be accessed to any outsiders.

Nowadays, a large number of attackers are monitoring those weaknesses in the system and these are attacking those kind of systems. A large number of users doesn't know about the importance of cyber security and that is the reason a large number of people are trapped in the attackers.

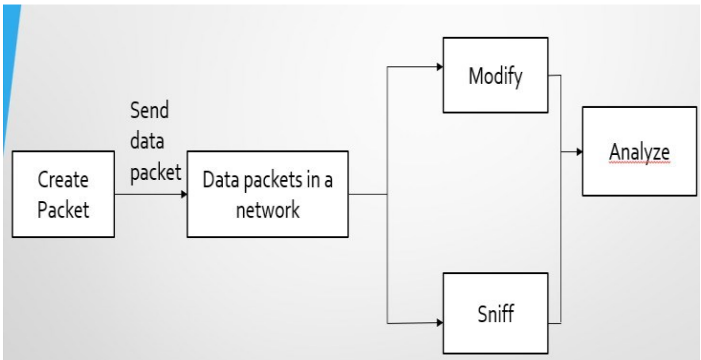
The process involved in our research was :

1. If you connects to a network, then you simply run the python file which was developed by us .

2. If the user runs the python file then it will produce the ports and the ip Address and it will demonstrate how the ports are opened and what are protocols which uses the attackers.

3. Every protocol will uses a port such that the port takes the responsibility for the functionality of the protocol.

Fig 3. Block Diagram



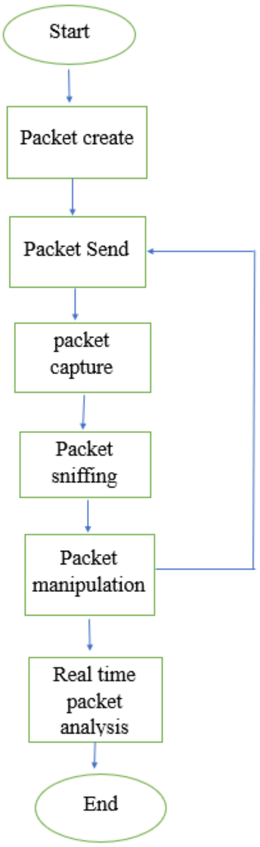


Fig.4 Flow Chart of Proposed System

VII. RESULTS:

Fig.5 Opening The Scappy Powerfull Tool

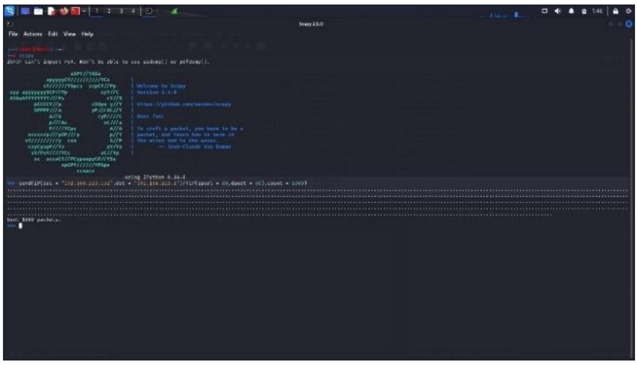


Fig 6 : Sending TCP/UDP Packets to the Network

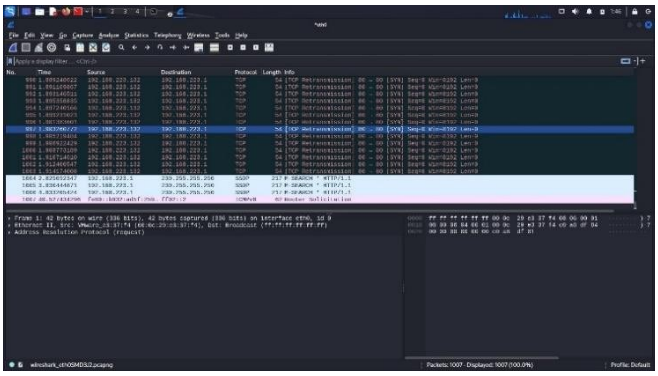


Fig.7 : Capturing the Network Traffic

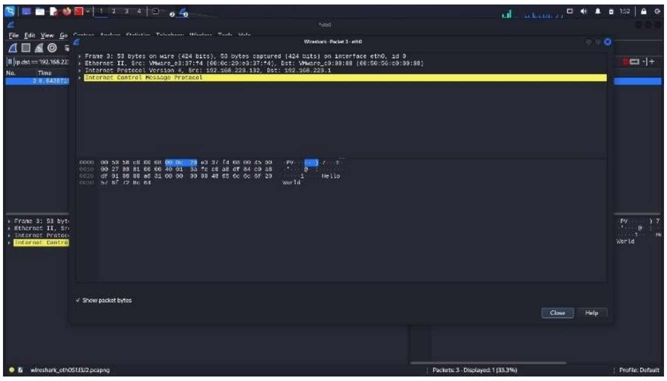
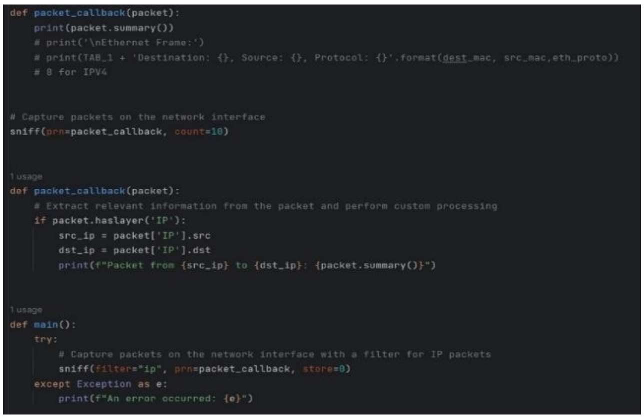


Fig.8: Captured the fake packet



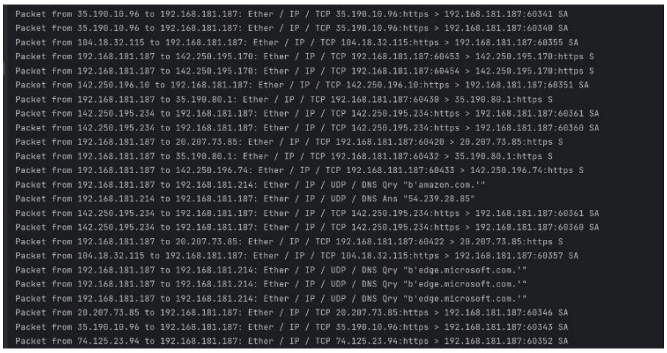
```
def packet_callback(packet):
    print(packet.summary())
    # print('\nEthernet Frame:')
    # print(TAB_1 + 'Destination: {}, Source: {}, Protocol: {}'.format(dest_mac, src_mac,eth_proto))
    # 8 for IPV4


# Capture packets on the network interface
sniff(prn=packet_callback, count=10)


1 usage
def packet_callback(packet):
    # Extract relevant information from the packet and perform custom processing
    if packet.haslayer('IP'):
        src_ip = packet['IP'].src
        dst_ip = packet['IP'].dst
        print(f"Packet from {src_ip} to {dst_ip}: {packet.summary()}")


1 usage
def main():
    try:
        # Capture packets on the network interface with a filter for IP packets
        sniff(filter="ip", prn=packet_callback, store=0)
    except Exception as e:
        print(f"An error occurred: {e}")
```

Fig. 9 : Sniffing the Packets.



```
Packet from 35.190.10.96 to 192.168.181.187: Ether / IP / TCP 35.190.10.96:https > 192.168.181.187:60341 SA
Packet from 35.190.10.96 to 192.168.181.187: Ether / IP / TCP 35.190.10.96:https > 192.168.181.187:60340 SA
Packet from 104.18.32.115 to 192.168.181.187: Ether / IP / TCP 104.18.32.115:https > 192.168.181.187:60355 SA
Packet from 192.168.181.187 to 142.250.195.170: Ether / IP / TCP 192.168.181.187:60453 > 142.250.195.170:https S
Packet from 192.168.181.187 to 142.250.195.170: Ether / IP / TCP 192.168.181.187:60454 > 142.250.195.170:https S
Packet from 142.250.196.10 to 192.168.181.187: Ether / IP / TCP 142.250.196.10:https > 192.168.181.187:60351 SA
Packet from 192.168.181.187 to 35.190.80.1: Ether / IP / TCP 192.168.181.187:60430 > 35.190.80.1:https S
Packet from 142.250.195.234 to 192.168.181.187: Ether / IP / TCP 142.250.195.234:https > 192.168.181.187:60361 SA
Packet from 142.250.195.234 to 192.168.181.187: Ether / IP / TCP 142.250.195.234:https > 192.168.181.187:60360 SA
Packet from 192.168.181.187 to 20.207.73.85: Ether / IP / TCP 192.168.181.187:60420 > 20.207.73.85:https S
Packet from 192.168.181.187 to 35.190.80.1: Ether / IP / TCP 192.168.181.187:60432 > 35.190.80.1:https S
Packet from 192.168.181.187 to 142.250.196.74: Ether / IP / TCP 192.168.181.187:60433 > 142.250.196.74:https S
Packet from 192.168.181.187 to 192.168.181.214: Ether / IP / UDP / DNS Qry "b'amazon.com.'"
Packet from 192.168.181.214 to 192.168.181.187: Ether / IP / UDP / DNS Ans "54.239.28.85"
Packet from 142.250.195.234 to 192.168.181.187: Ether / IP / TCP 142.250.195.234:https > 192.168.181.187:60361 SA
Packet from 142.250.195.234 to 192.168.181.187: Ether / IP / TCP 142.250.195.234:https > 192.168.181.187:60360 SA
Packet from 192.168.181.187 to 20.207.73.85: Ether / IP / TCP 192.168.181.187:60422 > 20.207.73.85:https S
Packet from 104.18.32.115 to 192.168.181.187: Ether / IP / TCP 104.18.32.115:https > 192.168.181.187:60357 SA
Packet from 192.168.181.187 to 192.168.181.214: Ether / IP / UDP / DNS Qry "b'edge.microsoft.com.'"
Packet from 192.168.181.187 to 192.168.181.214: Ether / IP / UDP / DNS Qry "b'edge.microsoft.com.'"
Packet from 192.168.181.187 to 192.168.181.214: Ether / IP / UDP / DNS Qry "b'edge.microsoft.com.'"
Packet from 20.207.73.85 to 192.168.181.187: Ether / IP / TCP 20.207.73.85:https > 192.168.181.187:60346 SA
Packet from 35.190.10.96 to 192.168.181.187: Ether / IP / TCP 35.190.10.96:https > 192.168.181.187:60343 SA
Packet from 74.125.23.94 to 192.168.181.187: Ether / IP / TCP 74.125.23.94:https > 192.168.181.187:60352 SA
```

Fig. 10: Protocols and the Packets.

VIII. CONCLUSION

In conclusion, networks become more complex and prone to hacking, packet sniffing evolves the tools to beginners as a method of understanding network operations to becoming an indispensable tool for network administrators, cybersecurity professionals, and ethical hackers. Their ability to provides the network behavior of the security, security weaknesses (such as intrusions or hijackers), and potential threats is widely recognized across varying scales, including both small and large-scale networks, with their importance in areas like enterprise environments and cyber attacks, underscoring vital systems, etc. over time. Through educating themselves on the inner workings of packet sniffers, taking the measures for detecting them, and adopting security mechanisms such as encryption and regular security audits, individuals and organizations can save the data being intercepted or accessed without authorization. A security-conscious approach is essential to maintaining the security networks and ensuring their protection against evolving threats. The development of continuous capture systems has transformed packet sniffers into potent tools for identifying and diagnosing any threat or network event. Teams that utilize packet capture can anticipate incidents faster and with greater precision, while also addressing cyber incidents, performance issues (such as failures, outages), and other network-related problems.

IX. REFERENCES :

[1] C.N.A. Program, Introduction to Networks Companion Guide: Pearson Education, 2013. .

[2] "Manajemen Bandwidth dan Monitoring Akses Data," by A. Siswanto and A. Tedyyana, (National Seminar on Information and Communication Technology, Medan, 2014), pp. 24-28.

[3] "Packet Sniffing and Network Traffic Analysis Using TCP—A New Approach," A. Bhandari, S. Gautam, T. K. Koirala, and M. R. Islam, in Advances in Communication, Electronics, and Computing, ed.: Springer, 2018, pp. 273-280.

[4] S. Ansari, S. Rajeev, and H. Chandrashekar, "Packet sniffing: a brief introduction," IEEE potentials, vol. 21,
pp. 17-19, 2002.

[5] "Detection of abnormal internet usage in LAN Islamic University of Riau, Indonesia" by S. L. Rosa and E. A. Kadir, in Proceedings of the International Conference on Intelligent Science and Technology, 2018, pp. 17-22.

[6] M. A. Qadeer, A. Iqbal, M. Zahid, and M. R. Siddiqui, "Network traffic analysis and intrusion detection using packet sniffer," Second International Conference on Communication Software and Networks, 2010. ICCSN'10. 2010, pp. 313-317.

[7] N. Lizarti and W. Agustin, "Aplikasi Network Traffic Monitoring Menggunakan Simple Network Management Protocol (SNMP) pada Jaringan Virtual Private Network (VPN)," SATIN-Sains and Information Technology vol. 1,
pp. 27-34, 2015.

[8] T. Lammle, Routing and Switching with CCNA Study Guide: Exams 100-101, 200-101, and 200-120: John Wiley & Sons, 2013.

[9] T. Lammle, Cisco Certified Network Associate, or CCNA Deluxe Study Guide: John Wiley & Sons, 2011.

[10] T. King, "Packet sniffing in a switched environment," SANS Institute, GESC practical, vol. 1, 2002.

[11] Analysis of different packet sniffing technologies for network monitoring and analysis by P. Asrodia and H. Patel analysis," International Journal of Electrical, Electronics and Computer Engineering, vol. 1, pp. 55-58, 2012