

Comparative Performance Analysis Of Cybersecurity Tools On A Wireless Network With WPA2 Encryption

Ekpedeme John¹

Department Of Electrical/Electronic and Computer Engineering
University of Uyo, Akwa Ibom State Nigeria

Constance Kalu²

Department Of Electrical/Electronic and Computer Engineering
University of Uyo, Akwa Ibom State Nigeria

Philip Asuquo³

Department Of Electrical/Electronic and Computer Engineering
University of Uyo, Akwa Ibom State Nigeria

Abstract— In this paper comparative performance analysis of cybersecurity tools on a wireless network with WPA2 encryption is presented. Specifically, experiments were conducted to test and compare the performance of four (4) open-source cybersecurity tools namely; Suricata, Snort, Aircrack-ng and Wireshark. The performance analysis of the four selected cybersecurity tools was carried out on a case study wireless network that consisted of MTN Router, 2 personal computers, and some mobile phones. The metric tested include packet capturing and Intrusion detection abilities, man-in-the-middle-attack, password cracking, ease of installation, and usage. The result show that all the tools tested had different degrees of packet capturing and intrusion detection ability. However, Wireshark performs better than the other tools in the aspect of packet capturing and analysis since it does not only show the source/destination IP/mac addresses but include information like frame check sequence, checksum, port number, protocol type. Also, the results showed that Snort was very efficient in the aspect of intrusion detection. None of the 4 tools was able to initiate Man-in-the-middle attack. Password cracking was implemented using Wireshark and Aircrack-ng. Also, except for Snort and Suricata, all the tools are easy to install and use. In all, the study found out that, no single tool is enough to initiate the performance analysis alone. For example, software applications like Aircrack-ng depended on packets captured by Wireshark to initiate password cracking. It was also discovered that some features in the tools that are not responding were due to a version of the operating system that was used for this study (that is. Windows 10).

Keywords— Cybersecurity, Wireshark Wireless Network, Suricata, Wpa2 Encryption, Packet Sniffing, Snort, Man in the Middle Attack, Aircrack-ng

1. Introduction

In the early years of telecommunication, wired and fiber optic communication were quite popular [1,2, 3,4, 5, 6,7, 8, 9, 10, 11], but the advancements in wireless communication technologies has brought about applications of wireless communication technologies in terrestrial and satellite communication, as well as in deep space communication [12,13,14, 15, 16, 17,18, 19, 20, 21,22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33]. As wireless communication technologies continues to dominate the telecommunication industry, researchers continue to identify and address numerous challenges that are associated with such technologies. Some of the notable challenges associated with classical wireless communication systems include pathloss, diffraction loss, multipath fading, rain fading, interference, limited bandwidth, among others [34,35, 36, 37, 38, 39,40, 41,42, 43,44, 45,46, 47,48, 49,50, 51,52, 53, 54, 55, 56, 57, 58, 59, 60, 61]. However, in this present era of Internet and associated Internet of Things technologies, wireless communication systems are increasingly faced with security challenges [62,63,64]. As, such increasingly, researchers are working tirelessly to develop cybersecurity tools and solution to address the security challenges associated with wireless communication networks. Notably, each cybersecurity tool can be effective in addressing some aspects of the security challenges prevalent in wireless networks. As such, proper cybersecurity tools analysis is required in order to identify the appropriate tool to use for different cybersecurity threats and attacks [65,66].

Basically, cybersecurity tools analysis methodologies are the various strategies or approaches used to check a security tool to certify that it performs the expected task. These

include a method of installation, selection of network adapter, Packet Sniffing, Man in the middle attack, etc [67,68]. For this paper, cybersecurity tools analysis will be defined as the procedure used to analyze the way and manner cyber security tools can be used to monitor traffic, capture and analyze packets, expose and crack the password and generally expose the vulnerabilities of a network [69,70]. The penetration testing is conducted on a small case study wireless network set up for this purpose using some of the available free cybersecurity tools.

The performance of cyber security tools is highly dependent on the diverse functionalities of the individual tool [71,72]. For example, some of the tools are more capable of packet capturing/analysis and intrusion detection than others while some are more efficient in password cracking and man in the Middle attack than others. Some tools are designed for a single case scenario while some are able to perform two or more types of functions. For this study, the performance of 4 cybersecurity tools was investigated on a case study wireless network with WPA2 encryption [73,74].

Specifically, experiments were conducted to test and compare the performance of four (4) open-source cybersecurity tools namely; Suricata [75,76,77], Snort [77,78,79], Aircrack-ng [80,81] and Wireshark [82,83]. The performance analysis of the four selected cybersecurity tools was carried out on a case study wireless network that consisted of MTN Router, 2 personal computers, and some mobile phones. The metric tested include packet capturing and intrusion detection abilities, man-in-the-middle-attack, password cracking, ease of installation, and usage [84,85,86,87]. The results of the experimental study and the discussion of the results are presented.

2.0 Methodology

In this paper, experiments were conducted to test and compare the performance of four (4) open-source cybersecurity tools namely; Suricata, Snort, Aircrack-ng and Wireshark. A small case study lab was set up which consist of MTN Router, 2 personal computers, and some mobile phones. The function of the MTN MF253v router (Figure 1) is to serve as a source of the internet. It has a operates at maximum frequency of 1900 MHz, maximum transmission rate of 150mbps and supports Edge, Gprs, and HSDPA communication protocol. It is also compatible with Windows 7, 8, and 10.

The personal computer (PC) shown in Figure 2 is HP 250 G3 Notebook with an installed Windows 10 professional operating system. The PC has a hard disk capacity of 500 GB, a Quad-core processor, and an installed Ram of 4Gb. The PC serves as a threat source in this project i.e., Network attack/ Analysis software such as Cain and Abel and Wireshark will be installed and launched from this PC.

Another personal computer shown in Figure 3 is a Dell Latitude E5510 with an installed windows 10 professional operating system. The Dell PC has a hard disk capacity of 500Gb, a Dual-core processor, and an installed Ram of 4Gb. It serves as a threat victim in this project , that means the analysis software such as Suricata, Snort, and Wireshark are installed and launched from this Dell PC. It

will be used to monitor traffic and analyze packets to check their accuracy in detecting the threat.



Figure 1 MTN MF253v Router



Figure 2 Personal Computer (HP 250 G3 Notebook)



Figure 3 Personal Computer (Dell Latitude E5510)

2.1 The Software Tools

2.1.1 SURICATA

Suricata is an open-source IDS/IPS tool developed by the Open Security Foundation (OSF). It performs intrusion detection/prevention procedures and it has detection engine which is a very sophisticated and vigorous network threat detection engine. It is highly valued because it has ability to initiate capturing of data at layer 7 of the OSI model (i.e., application layer). Suricata tool can integrate with third-party tools such as Anavar, Squil, BASE, Kibana and

Snorby. It has an in-built scripting module, intelligent processing architecture and makes use of signature and anomaly-based methods. In order to use Suricata tool one has to download and install it in the computer system used for the vulnerability analysis. The procedure for the installation of Suricata on Windows operating system can be accessed at the following URLs;

- i. <https://blog.eldernode.com/install-suricata-on-windows-10/>
- ii. https://redmine.openinfosecfoundation.org/attachments/download/1166/SuricataWinInstallationGuide_v1.4.2.pdf

2.1.2 WIRESHARK

Wireshark is a free cross-platform and open-source packet analyzing tool mostly employed by network administrators and cyber security experts for things like network troubleshooting, packet analysis, application development and communications protocol development, and also education. It has ability to capture packets and its [promiscuous mode](#) feature allows administrators to view visible traffic on that interface. In addition, it can effectively be used on Microsoft Windows and Linux, as well as on other operating systems.

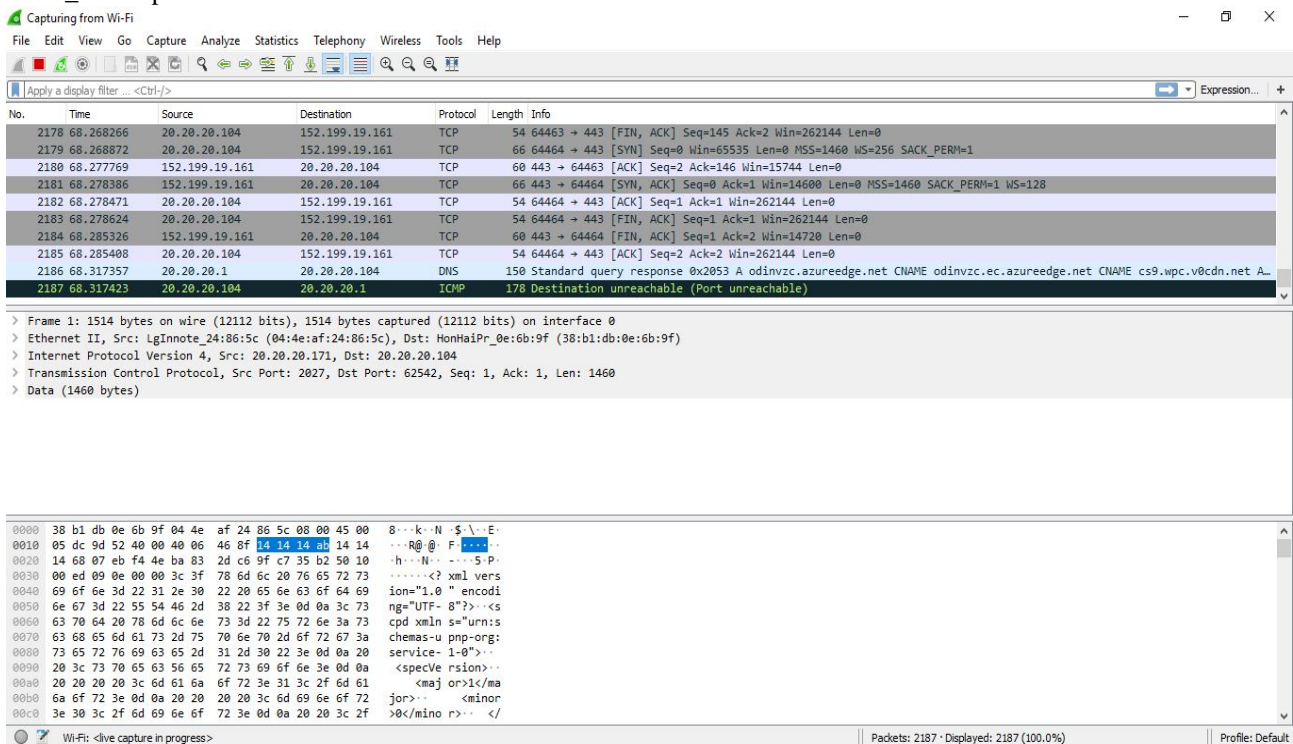


Figure 4 Wireshark Interface

A sample Wireshark interface is shown in Figure 4 and the installation procedure for Wireshark tool is as follows:

- i. Download and install Wireshark.exe from <https://www.wireshark.org/download.html>
- ii. Double-click on the file to open it. And follow subsequent instructions to complete the setup
- iii. Ensure to install WinPcap.
- iv. Select Finish completing the installation of Wireshark.
- v. Open the Wireshark and start monitoring network traffic

2.1.3 AIRCRACK-NG

Aircrack-ng is a cybersecurity tool that performs a variety of functions such as packet sniffing, cracking passwords, and analyzing packets. It is ready tool for wireless hacking and vulnerability testing. Also, it can use packet injection to initiate attacks like de-authentication and Cracking wireless protocol such as WEP and WPA1 and WPA2 through the use of dictionary /wordlist. A sample Aircrack-ng interface is shown in Figure 5. The installation procedure of Aircrack-ng on Windows operating system is quite lengthy but the procedure can be accessed from any of the following URLs;

- i. <https://windows-1.com/aircrack-ng-for-pc.html>
- ii. <https://www.systranbox.com/how-to-install-aircrack-ng-on-windows-10-kali-linux/>
- iii. https://www.aircrack-ng.org/doku.php?id=install_aircrack

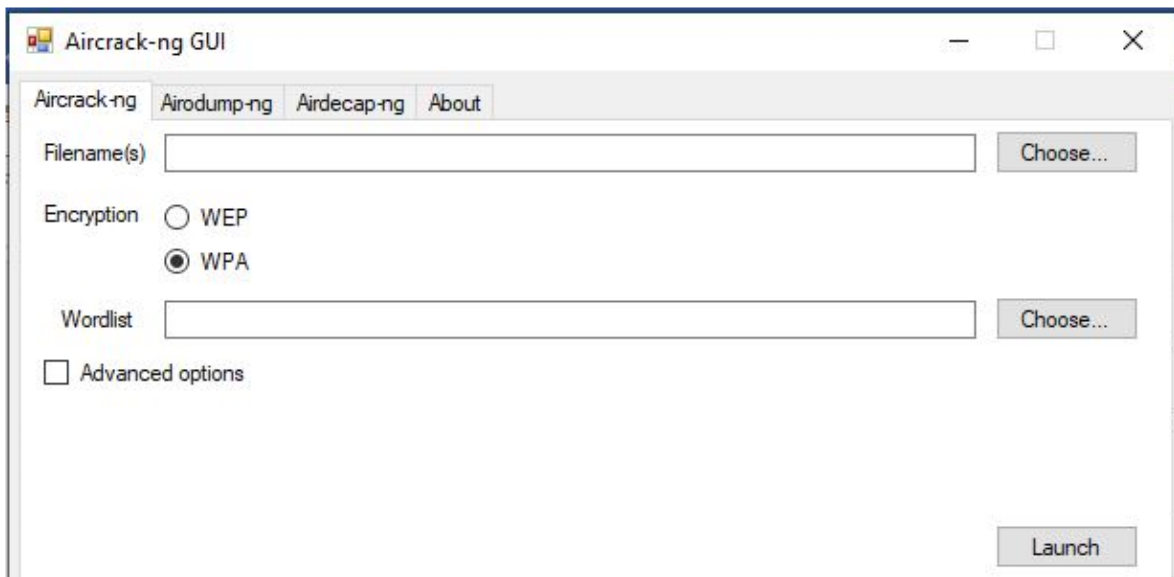


Figure 5 Aircrack-ng Interface

2.1.4 Snort

Snort, developed by Cisco, is one of the open-source network-based intrusion detection/prevention system (IDS/IPS) tools equipped with the ability to carry-out several functions such as real-time traffic analysis, protocol analysis, content searching and matching, and packet logging on Internet Protocol (IP) networks. It also functions as detector for probes or attacks, operating system fingerprinting attempts, semantic URL attacks, buffer overflows, server message block probes, and stealth port scans. However, although snort can carry out a lot of functions to ensure safety and security in a network, its most crucial function is intrusion detection using both Anomaly-based and Signature-based methods.

Notably, Snort can be configured in three main modes: 1. sniffer, 2. packet logger, and 3. network intrusion detection.

Sniffer Mode: The program will read network packets and display them on the console.

Packet Logger Mode: In packet logger mode, the program will log packets to the disk.

Network Intrusion Detection System Mode: In intrusion detection mode, the software will check and analyze packets and scrutinize it against a rule set defined by the user.

A sample **Snort** interface is shown in Figure 6. The installation procedure of Snort on Windows operating system is quite lengthy but the procedure can be accessed online at;

<https://zaeemjaved10.medium.com/installing-configuring-snort-2-9-17-on-windows-10-26f73e342780>.

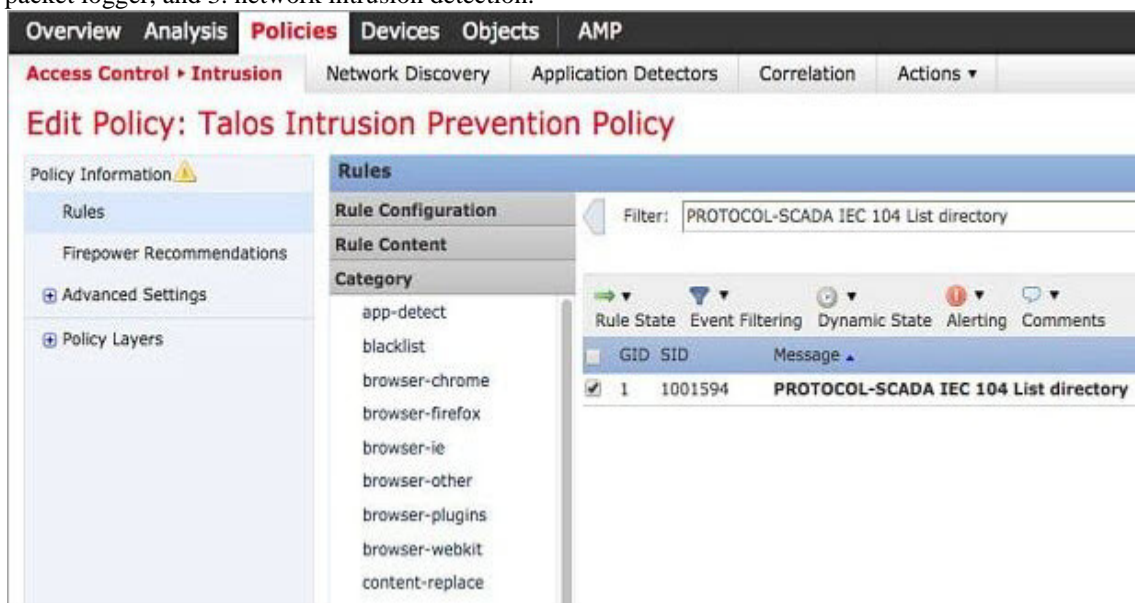


Figure 6 Snort GUI

2.2 The cybersecurity tools performance analysis

The performance analysis of the four selected cybersecurity tools was carried out on a case study wireless network that was set-up primarily for this purpose. The small case study wireless network that was set up consisted of the MTN Router, 2 personal computers, and some mobile phones. Once the computer system was booted up to the desktop, it was connected to the network and each cybersecurity tool was launched. Once all the tools were analyzed, the corresponding result was documented and discussed. The implementation flowchart for the cybersecurity tools performance analysis is shown in Figure 7. Also, the picture of the two PCs that are wirelessly connected to the MTN Router is shown in Figure 8 while the schematic diagram of the network set up is shown in Figure 9.

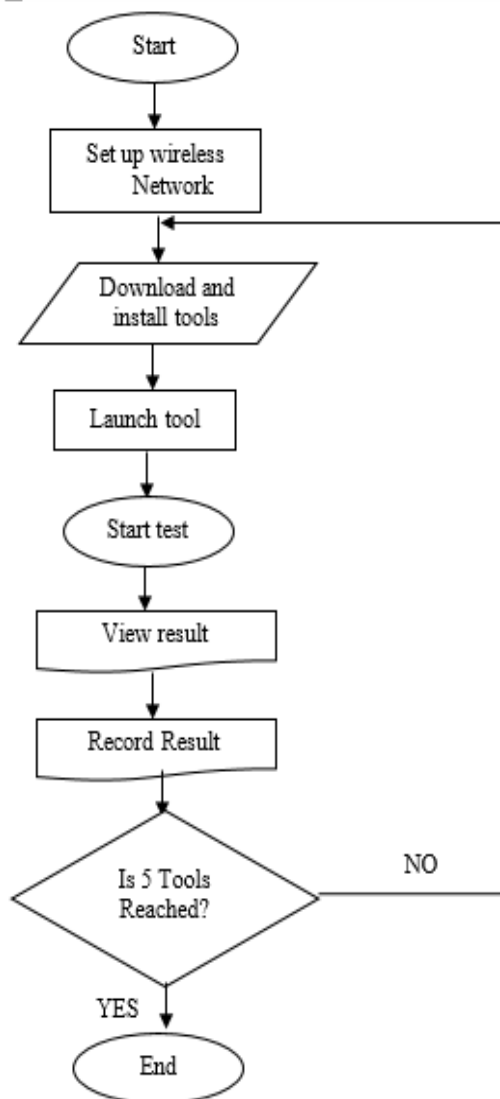


Figure 7 Implementation flowchart for the cybersecurity tools performance analysis



Figure 8 Picture of the two PCs that are wirelessly connected to the MTN Router

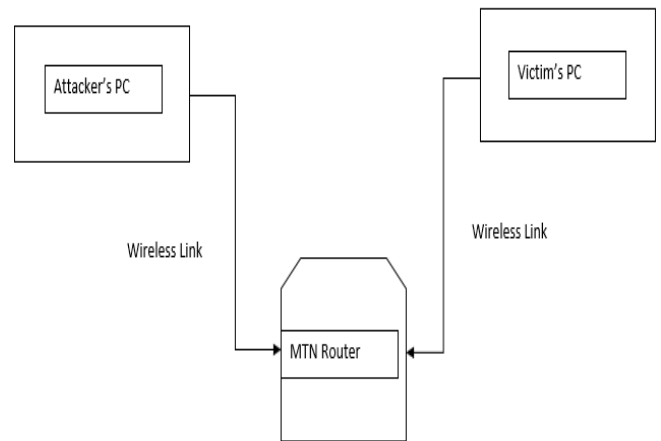


Figure 9 The schematic diagram of the network set up

3.0 Results and Discussion

In this paper, the performance of 4 cybersecurity tools were analysed for a wireless network with WPA-2 encryption. The results are presented for each of the tools. Next, a comparative analysis of the tools are made based on the results obtained for each of the tools.

3.1 Suricata

i. Packet capturing/Intrusion detection

In order to capture packet in Suricata, the command “PS C:\suricata> C:\suricata\suricata.exe -c C:\suricata\suricata.yaml -i 193.55.100.145” was used, as shown in Figure 10, where -c and -i signifies capture and interface respectively and 193.55.100.145 was the IP address of the network interface used in the performance analysis. “C:\suricata\log> ls” is the command used to show the statistics of captured packet.as shown in **Figure 11**.

```

PS C:\suricata> C:\suricata\suricata.exe -c C:\suricata\suricata.yaml -i 193.55.100.145
7/12/2021 -- 07:22:50 - <Info> - Running as service: no
7/12/2021 -- 07:22:50 - <Info> - translated 193.55.100.145 to pcap device \Device\NPF_{00D9825E-0120-4625-B489-82659F7FA9E9}
7/12/2021 -- 07:22:50 - <Notice> - This is Suricata version 6.0.3 RELEASE running in SYSTEM mode
7/12/2021 -- 07:22:50 - <Notice> - Protocol detector and parser disabled for SSH

```

Figure 10 Packet Capturing Begins in Suricata

```

PS C:\suricata> cd log
PS C:\suricata\log> ls

Directory: C:\suricata\log

Mode                LastWriteTime         Length Name
----                -
d-----           12/3/2021   4:25 PM             files
-a----           12/7/2021   7:53 AM      45836581 eve.json
-a----           12/3/2021   5:16 PM              0 fast.log
-a----           12/7/2021   7:53 AM      29683147 stats.log
-a----           12/7/2021   7:53 AM       36704 suricata.log

```

Figure 11 Packet captured statistics in Suricata

- ii. **Man-in-the-Middle Attack (ARP poison) and Password cracking** : These features are not available on Suricata

3.2 Wireshark

i. Packet capturing/Intrusion detection

In this paper, Wireshark was mainly used to sniff and capture packets. The targeted interface card was selected and several filters were put in the filter bar to help narrow

down the search result. For example, filters like “**UDP.port==53**” helps to ensure that only UDP packets are captured and monitored for intrusion. The screenshot showing some packets captured by the Wireshark tool is given in Figure 12.

In Figure 13 it is shown that the domain name of the website in the captured packet is made visible by wireshark. The captured packet is then saved and used for analysis by other cyber tools such as Aircrack-ng.

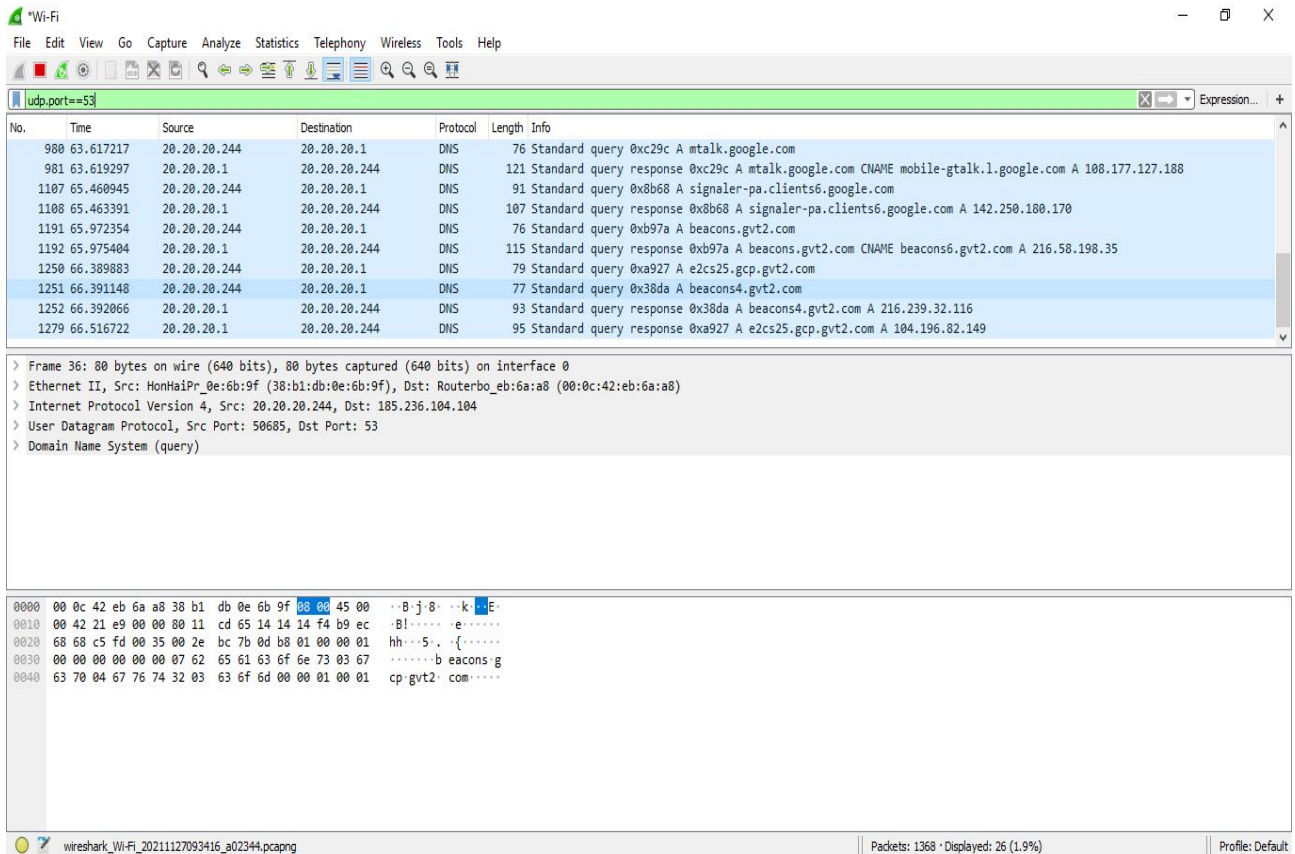


Figure 12 Captured DNS packets



Figure 13 Resolved DNS

ii. Password Decryption/Cracking

Wireshark was used to discover password over a WPA-2 encrypted packet. However, password can only be discovered if user tries to gain access to unsecured website, that is, websites that begins with “HTTP” alone. The following command “`http.request.method == post`” was used on Wireshark to filter such traffic. Wireshark was not able to crack or decrypt the packet to expose the password, however, the captured packet was decrypted by Aircrack-ng.

iii. Man-in-the-Middle Attack (ARP poison):
 Wireshark is not equipped with this feature

3.3 Aircrack-ng

i. Password Decryption/Cracking

In this paper, Aircrack-ng was used to initiate password cracking. To implement this procedure, Wireshark was used to capture packet that may likely contain the password as shown in Figure 14.

Protocol	Length	Info
DNS	139	Standard query response 0xf3ef A adservice.google.com.ng CNAME pagead46.l...
DNS	85	Standard query 0x0b28 A tpc.googlesyndication.com
DNS	101	Standard query response 0x0b28 A tpc.googlesyndication.com A 142.250.184.65
DNS	78	Standard query 0xf80f A www.googleapis.com
DNS	318	Standard query response 0xf80f A www.googleapis.com A 216.58.205.74 A 142...
DNS	96	Standard query 0x138b A passwordsleakcheck-pa.googleapis.com
DNS	112	Standard query response 0x138b A passwordsleakcheck-pa.googleapis.com A 14...

Figure 14 Password in Wireshark

The captured packet was saved in .Pcap format, and a wordlist known as "10_million_password_list_top_1000000.txt" was also downloaded from <https://weakpass.com/wordlist/50>. In this context, a wordlist is a list of passwords that are documented in an unencrypted format. It contains a list of frequently used passwords and possible combination of letters which can facilitate passwords cracking.

There are two methods of importing the captured packet and wordlist on Aircrack-ng are Graphic users interface (GUI) and Command Line Interface (CLI). The CLI mode is use by typing the following command in the command line interface "C:\Program Files\aircrack-ng-1.6-

```
win\bin>aircrack-ng "D:\wire shark.pcap" -w "D:\10_million_password_list_top_1000000.txt"
```

C: represent the path that contains the aircrack application while D: represent the path containing captured packet and the wordlist. The GUI method involves importing the captured packet and wordlist from the graphic user interface as shown Figure 15. The application immediately opened the command prompt containing some information about packet after clicking the launch button. The screenshot in Figure 16 is captured for the command prompt showing packet's information. Again, in Figure 17 it is shown that Aircrack-ng tool successfully decrypted the session password from the captured packet and wordlist that was imported. It is correctly displayed beside the KEY FOUND section.



Figure 15

imp Screenshot showing how to import the wordlist and captured packet


```
C:\Program Files\aircrack-ng-1.6-win\bin>aircrack-ng "D:\wire shark.pcap" -w D:\10_million_password_list_top_1000000.txt
Reading packets, please wait...
Opening D:\wire shark.pcap
Read 227 packets.

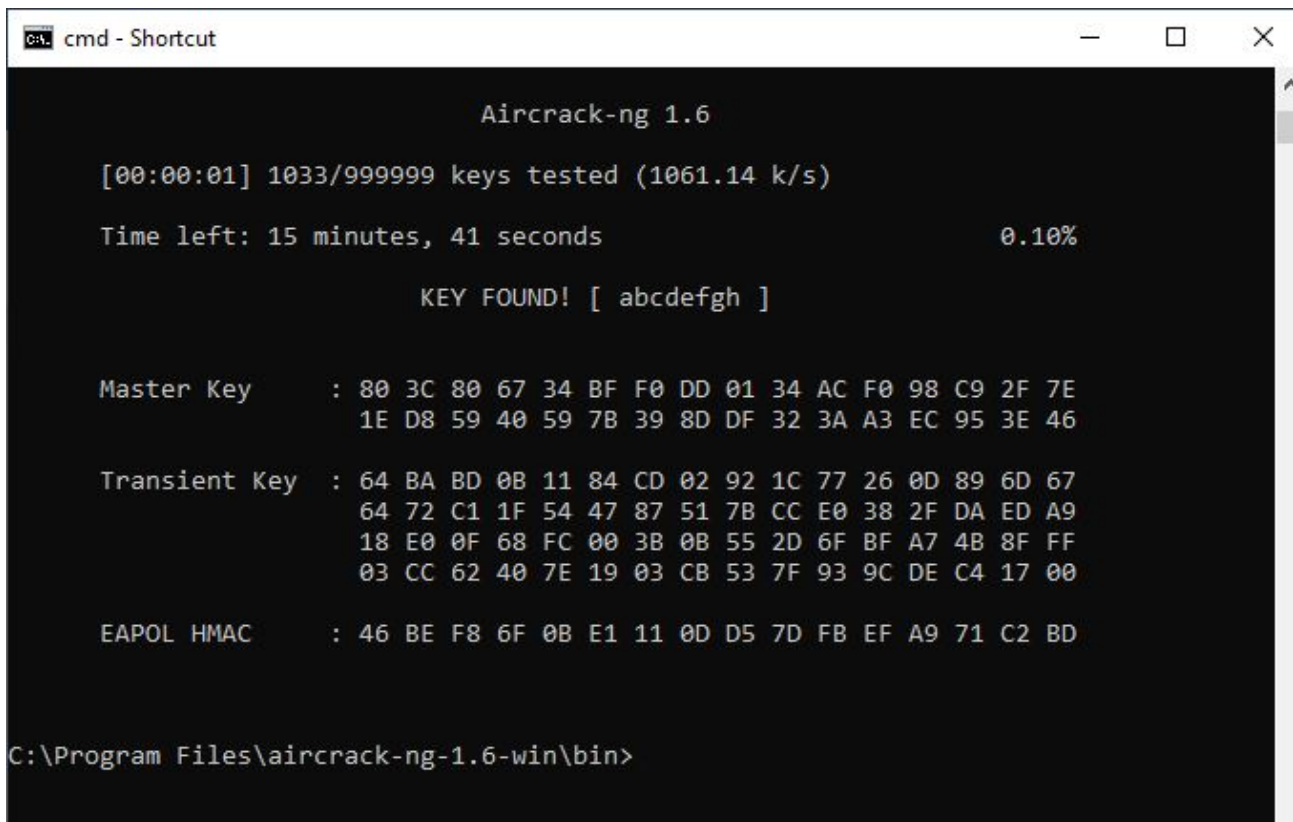
# BSSID          ESSID          Encryption
1 64:E5:99:7A:E9:64 test-ap        WPA (1 handshake)

Choosing first network as target.

Reading packets, please wait...
Opening D:\wire shark.pcap
Read 227 packets.

1 potential targets
```

Figure 16 The command prompt showing packet's information



```
cmd - Shortcut

Aircrack-ng 1.6

[00:00:01] 1033/999999 keys tested (1061.14 k/s)

Time left: 15 minutes, 41 seconds          0.10%

KEY FOUND! [ abcdefgh ]

Master Key      : 80 3C 80 67 34 BF F0 DD 01 34 AC F0 98 C9 2F 7E
                  1E D8 59 40 59 7B 39 8D DF 32 3A A3 EC 95 3E 46

Transient Key   : 64 BA BD 0B 11 84 CD 02 92 1C 77 26 0D 89 6D 67
                  64 72 C1 1F 54 47 87 51 7B CC E0 38 2F DA ED A9
                  18 E0 0F 68 FC 00 3B 0B 55 2D 6F BF A7 4B 8F FF
                  03 CC 62 40 7E 19 03 CB 53 7F 93 9C DE C4 17 00

EAPOL HMAC     : 46 BE F8 6F 0B E1 11 0D D5 7D FB EF A9 71 C2 BD

C:\Program Files\aircrack-ng-1.6-win\bin>
```

Figure 17 Aircrack-ng showing password from the captured packet

- ii. **Packet capturing/Intrusion detection:** the packet capturing feature was not loading due to operating system incompatibility
- iii. **Man-in-the-Middle Attack (ARP poison) :** Aircrack-Ng is not equipped with this feature

3.4 Snort

i Packet capturing/Intrusion detection

To implement packet capturing on Snort, some rules were set in the **C:\Snort\rules\local** path, this path automatically opened in notepad where one now has the privilege to put in some rules with regards to the type of packets you want to capture. As shown in Figure 18, ICMP, TCP, and UDP packets were set to be captured.

After setting the local rules, the command “snort -i4” was used in the Command prompt. This command was used to capture all the traffic as set in the local rules file. On implementing the command, snort was set to Packet Dump Mode and the system started capturing the traffic in the

network as seen in Figure 19. The information in the packet include date, packet type, source/destination addresses, IP-version etc, as shown in Figure 20. The capturing session was stopped by using “Ctrl-C”, this also brought out the breakdown of the entire packet captured.

```
#-----
# LOCAL RULES
#-----
alert icmp any any -> any any (msg:"TESTING ICMP"; sid:10000001;)
alert tcp any any -> any any (msg:"TESTING TCP"; sid:10000002;)
alert udp any any -> any any (msg:"TESTING UDP"; sid:10000003;)
```

Figure 18 Local rules folder for categories of packets to be captured

```
C:\Snort\bin>snort -i4
Running in packet dump mode

--== Initializing Snort ==--
Initializing Output Plugins!
pcap DAQ configured to passive.
The DAQ version does not support reload.
Acquiring network traffic from "\Device\NPF_{00D9825E-0120-4625-B4B9-82659F7FA9E9}".
Decoding Ethernet

--== Initialization Complete ==--
```

Figure 19 Snort in Packet dump mode

```
12/02-09:38:28.593606 [**] [1:10000003:0] TESTING UDP [**] [Priority: 0] {UDP} fe80:0000:0000:0000:e9d3:2bf3:c6c4:0d00:49664 -> ff02:0000:0000:0000:0000:0000:0001:0003:5355
12/02-09:38:28.594390 [**] [1:10000003:0] TESTING UDP [**] [Priority: 0] {UDP} 193.55.100.113:49664 -> 224.0.0.252:5355
12/02-09:38:28.597204 [**] [1:10000003:0] TESTING UDP [**] [Priority: 0] {UDP} 193.55.100.145:58832 -> 8.8.8.8:53
12/02-09:38:28.600219 [**] [1:10000003:0] TESTING UDP [**] [Priority: 0] {UDP} fe80:0000:0000:0000:e9d3:2bf3:c6c4:0d00:51351 -> ff02:0000:0000:0000:0000:0000:0001:0003:5355
12/02-09:38:28.600986 [**] [1:10000003:0] TESTING UDP [**] [Priority: 0] {UDP} 193.55.100.113:51351 -> 224.0.0.252:5355
12/02-09:38:28.611728 [**] [1:10000003:0] TESTING UDP [**] [Priority: 0] {UDP} fe80:0000:0000:0000:e9d3:2bf3:c6c4:0d00:59879 -> ff02:0000:0000:0000:0000:0000:0001:0003:5355
12/02-09:38:28.612495 [**] [1:10000003:0] TESTING UDP [**] [Priority: 0] {UDP} 193.55.100.113:59879 -> 224.0.0.252:5355
12/02-09:38:28.714509 [**] [1:10000003:0] TESTING UDP [**] [Priority: 0] {UDP} 8.8.8.8:53 -> 193.55.100.145:58832
12/02-09:38:28.716293 [**] [1:10000002:0] TESTING TCP [**] [Priority: 0] {TCP} 193.55.100.145:51913 -> 20.50.80.210:443
12/02-09:38:28.777669 [**] [1:10000003:0] TESTING UDP [**] [Priority: 0] {UDP} 193.55.100.113:52650 -> 239.255.255.250:1900
12/02-09:38:28.806961 [**] [1:10000001:0] TESTING ICMP [**] [Priority: 0] {IPV6-ICMP} fe80:0000:0000:0000:e9d3:2bf3:c6c4:0d00 -> ff02:0000:0000:0000:0000:0000:0000:0002
12/02-09:38:28.806961 [**] [1:10000001:0] TESTING ICMP [**] [Priority: 0] {IPV6-ICMP} fe80:0000:0000:0000:e9d3:2bf3:c6c4:0d00 -> ff02:0000:0000:0000:0000:0000:0000:0001:6
12/02-09:38:28.835961 [**] [1:10000002:0] TESTING TCP [**] [Priority: 0] {TCP} 20.50.80.210:443 -> 193.55.100.145:51913
12/02-09:38:28.836108 [**] [1:10000002:0] TESTING TCP [**] [Priority: 0] {TCP} 193.55.100.145:51913 -> 20.50.80.210:443
12/02-09:38:28.836901 [**] [1:10000002:0] TESTING TCP [**] [Priority: 0] {TCP} 193.55.100.145:51913 -> 20.50.80.210:443
12/02-09:38:28.958549 [**] [1:10000002:0] TESTING TCP [**] [Priority: 0] {TCP} 20.50.80.210:443 -> 193.55.100.145:51913
```

Figure 20 Captured packet in Command Prompt

```

=====
Run time for packet processing was 589.466000 seconds
Snort processed 50231 packets.
Snort ran for 0 days 0 hours 9 minutes 49 seconds
  Pkts/min:      5581
  Pkts/sec:      85
=====
Packet I/O Totals:
  Received:      50557
  Analyzed:      50231 ( 99.355%)
  Dropped:       323 ( 0.635%)
  Filtered:      0 ( 0.000%)
  Outstanding:   326 ( 0.645%)
  Injected:      0
=====
Breakdown by protocol (includes rebuilt packets):
  Eth:           50233 (100.000%)
  VLAN:          0 ( 0.000%)
  IP4:           47888 ( 95.332%)
  Frag:          0 ( 0.000%)
  ICMP:          0 ( 0.000%)
  UDP:           1087 ( 2.164%)
  TCP:           46431 ( 92.431%)
  IP6:           1104 ( 2.198%)
  IP6 Ext:       1484 ( 2.954%)
  IP6 Opts:      380 ( 0.756%)
  Frag6:         0 ( 0.000%)
  ICMP6:         477 ( 0.950%)
  UDP6:          627 ( 1.248%)
  TCP6:          0 ( 0.000%)
  Teredo:        0 ( 0.000%)
  ICMP-IP:       0 ( 0.000%)
  EAPOL:         0 ( 0.000%)
  IP4/IP4:       0 ( 0.000%)
  IP4/IP6:       0 ( 0.000%)
  IP6/IP4:       0 ( 0.000%)
  IP6/IP6:       0 ( 0.000%)

```

Figure 21 Breakdown of captured packet

The captured packet is saved or logged in the log folder of snort (i.e., C:\Snort\log), as shown in Figure 21 and can be read by using the command “C:\Snort\bin>snort -r C:\Snort\log\snort.log.1638433578”. -r initiate the read procedure while snort.log.1638433578 is the captured file name.

For Intrusion Detection there are three IP protocols that Snort currently analyzes for suspicious behavior. They are Transmission Control Protocol (tcp), User Datagram Protocol (udp), and Internet Control Message Protocol (icmp). As shown in Figure 22 and “snort -i 4 -c C:\Snort\etc\snort.conf -A console” was typed in C:\Snort\bin> directory and this automatically showed that snort is running in IDS mode

```

C:\Snort\bin>snort -i 4 -c C:\Snort\etc\snort.conf -A console
Running in IDS mode

  ---= Initializing Snort =---
  Initializing Output Plugins!
  Initializing Preprocessors!
  Initializing Plug-ins!
  Parsing Rules file "C:\Snort\etc\snort.conf"

```

Figure 23 Snort in IDS mode

- ii **Man-in-the-Middle Attack (ARP poison) and Password cracking** : These features are not available on for Snort

3.5 Discussion on the comparative analysis of the results of the 4 cybersecurity tools

i. Packet Capturing/Intrusion detection

From the result presented for each of the 4 cybersecurity tool, it can be deduced that all the tools tested had different degrees of packet capturing and intrusion detection ability. However, the results from the experiment showed that Wireshark performs better than other tools in the aspect of packet capturing and analysis since it does not only show the source/destination IP/mac addresses but include information like frame check sequence, checksum, port number, protocol type, etc. this information can prove valuable to network administrators and security expert when troubleshooting network vulnerability.

Notably, Snort was very efficient in the aspect of intrusion detection. This is due to the fact that Snort is designed to run in 2 modes which are Dump mode and IDS mode. Initiating Snort in Dump mode only Sniffed and captured the packet and then stored the captured packets in a selected folder but running Snort in IDS mode was able to identify bad traffic in real-time depending on the rules that was set in Snort.Conf file. Equally, Aircrack-ng is also equipped with packet capturing feature (Dump mode). However, in this study, the feature was unable to launch due to Operating System incompatibility.

Suricata was used in system mode to capture and analyses packet. However, it was not very efficient since it is very difficult to initiate filters and use the GUI on windows. Hence, the packet was only capture in Power shell CLI. Another downside of Suricata was that packet was captured in cluster instead of individually and one will only be notified of the captured traffic when it is already done and this makes it difficult to monitor the network in real-time.

It is worthy of note that packet sniffing and capturing only compromised the confidentiality of the data. However, the captured data can be used on other tools to cause more damages.

ii. Man-in-the-middle Attack

None of the 4 tools was able to initiate Man-in-the-middle attack. This attack compromised the integrity, and availability of the needed information.

iii. Password Cracking

Password cracking was implemented using Wireshark and Aircrack-ng. Notably, although Wireshark in itself cannot initiate password cracking except on an HTTP packet, it acted as a facilitator by capturing a packet that may contain a password, this packet was then acted upon by Aircrack-ng to crack the password. This attack compromised all the 3 aspects of the CIA-Triad

iv. Ease of Installation and Usage

Except for Snort and Suricata, all the tools are easy to install and use. Extra effort was needed to undergo training to understand how to correctly install and use Snort

and Suricata. Notably, Suricata and Snort's installation and usage require an appreciable degree of technical know-how.

The summary of the comparative analysis of the 4 cybersecurity tools is presented in Table 1. Also, Table 2 provides the classification of different cyber security tools based on their types of attack and other capacities. In addition, Table 3 shows the different cyber security tools and the layer through which they carried out their attack.

Table 1: Summary of the comparative analysis of the 4 cybersecurity tools

S/N	Tool	General Classification	Specific Uses
1	Wireshark	Packet Analyzer	used mainly for network packet capture/analysis, intrusion detection and wireless network connectivity testing/troubleshooting.
2	Aircrack-ng	Wireless Network Security tester	used to test the weaknesses of wireless networks by analyzing captured packets, passwords cracking and decryption, etc.
3	Snort	Intrusion Detection System	Used to scan the network in Dump and IDS mode to capture packets and give an alert when a questionable packet is detected based on the ruleset
4	Suricata	Packet Analyzer	Used to monitor traffic and give alerts based on the rule set

Table 2 Attacks/Tools

Metrics / Tools	Wireshark	Snort	Aircrack-ng	Suricata
Packet Capturing/Intrusion detection	Yes	Yes	No	Yes
Arp Poisoning (MITM)	No	No	No	No
Password Cracking	Only Http	No	Yes	No
Easy-To-Install	Yes	No	Yes	No
Easy-To-Use	No	No	Yes	No

Table 3: Attacks/Tools and their Layers of Attacks

Attacks / Tools	Wireshark	Snort	Aircrack-ng	Suricata
Packet Capturing/Intrusion detection	Datalink, Network and Transport Layer	Network/Transport Layer	Unable	Network layer
Arp Poisoning (MITM)	Unable	Unable	Unable	Unable
Password Cracking	Network Layer	Unable	Application Layer	Unable

4. Conclusion

Four (4) cybersecurity tools with different functionalities were employed in this study to capture, analyze, expose and crack packets. Particularly, experiments were conducted to test and compare the performance of four (4) open-source cybersecurity tools namely; Suricata, Snort, Aircrack-ng and Wireshark. The metric tested include packet capturing and intrusion detection abilities, man-in-the-middle-attack, password cracking, ease of installation, and usage.

The study found out that, no single tool is enough to initiate the performance analysis alone. For example, software applications like Aircrack-ng depended on packets captured by Wireshark to initiate password cracking. It was also discovered that some features in the tools that are not responding were due to a version of the operating system that was used for this study (that is. Windows 10).

References

- Bian, D., Kuzlu, M., Pipattanasomporn, M., Rahman, S., & Shi, D. (2019). Performance evaluation of communication technologies and network structure for smart grid applications. *IET Communications*, 13(8), 1025-1033.

- Ozuomba Simeon and Chukwudebe G. A.(2003) *An improved algorithm for channel capacity allocation in timer controlled token passing protocols, The Journal of Computer Science and its Applications (An international Journal of the Nigerian Computer Society (NCS)) Vol. 9, No. 1, June 2003, PP 116 124*
- Nwokonko Chisom . S and Ezenugu, I. A (2019)“Optimized Walficsh-Bertoni Path Loss Model For 1800GHZ Cellular Network Signal In Imo International Market Umusasa, Orlu Journal of Multidisciplinary Engineering Science Studies (JMEST) ISSN: 2458-925X Vol. 5 Issue 9, pp 3708 - 3711
- Kalu, S. Ozuomba, G. N. Onoh (2011) ANALYSIS OF TIMELY-TOKEN PROTOCOL WITH NON-UNIFORM HEAVY LOAD OF ASYNCHRONOUS TRAFFIC. *Electroscope Journal* Vol. 5 No. 5 (2011)
- Ezenugu, I. A, Nseobong. I. Okpura, Enyenihi Henry Johnson (2016) “Performance Evaluation of Empirical Rain Rate Models for Computing Rain Attenuation” *International Journal of Systems Science and Applied Mathematics*; Vol 1 No 4: pp86-90 <http://www.sciencepublishinggroup.com/j/ijssam>
- Ozuomba Simeon and Chukwudebe G. A.(2011) ; “Performance Analysis Of Timely-Token Protocol With Variable Load Of Synchronous Traffic” *NSE Technical Transactions , A Technical Journal of The Nigerian Society Of Engineers*, Vol. 46, No. 1 Jan – March 2011, PP 34 – 46.
- Ezenugu, I. A., Ikechukwu H. Ezech, Swinton C. Nwokonko (2017) “Determination of Single Knife Edge Equivalent Parameters for Triple Knife Edge Diffraction Loss by Giovanelli Method” *International Journal of Information and Communication Sciences* 2017; Vol 2 No1: pp10-14 <http://www.sciencepublishinggroup.com/j/ijics>
- Kalu, C., Ozuomba, Simeon., & Anthony, U. M. (2015). STATIC-THRESHOLD-LIMITED BuST PROTOCOL. *European Journal of Mathematics and Computer Science* Vol, 2(2).
- Ozuomba Simeon , Chukwudebe G. A. and Akaninyene B. Obot (2011); “Static-Threshold-Limited On-Demand Guaranteed Service For Asynchronous Traffic In Timely-Token Protocol “ *Nigerian Journal of Technology (NIJOTECH)* Vol. 30, No. 2 , June 2011 , PP 124 – 142
- Kalu C. , Ozuomba Simeon, Onoh G.N. (2013) Dynamic Threshold limited timed token (DTLTT) Protocol *Nigerian Journal of Technology (NIJOTECH)* Vol. 32. No. 1. March 2013, pp. 266-272.
- Ozuomba, Simeon, Amaefule, C. O., & Afolayan, J. J. (2013). Optimal Guaranteed Services Timed

- Token (OGSTT) Media Access Control (Mac) Protocol For Networks That Support Hard Real-Time And Non Real-Time Traffic. *Nigerian Journal of Technology (NIJOTECH)* 32(3), 470-477
12. Kalu C., Ozuomba S., and Mbocha C.C. (2013) Performance Analysis of Static- Threshold-Limited On-Demand Guaranteed Services Timed Token Media Access
 13. Simeon, Ozuomba. (2016). Evaluation Of Bit Error Rate Performance Of Multi-Level Differential Phase Shift Keying. Evaluation, 1(8). *International Multilingual Journal of Science and Technology (IMJST) Vol. 1 Issue 8, August – 2016*
 14. Ozuomba Simeon and Chukwudebe G. A. (2004) *A new priority scheme for the asynchronous traffic in timer-controlled token passing protocols, The Journal of Computer Science and its Applications (An international Journal of the Nigerian Computer Society (NCS)) Vol. 10, No. 2 , December 2004 , PP 17 -25*
 15. Routray, S. K. (2014). The changing trends of optical communication. *IEEE Potentials*, 33(1), 28-33.
 16. Anietie Basse, Simeon Ozumba & Kufre Udofia (2015). An Effective Adaptive Media Play-out Algorithm For Real-time Video Streaming Over Packet Networks. *European Journal of Basic and Applied Sciences Vol, 2(4)*.
 17. Idio, Uduak, Constance Kalu, Akaninyene Obot, and Simeon Ozuomba. (2013) "An improved scheme for minimizing handoff failure due to poor signal quality." In *2013 IEEE International Conference on Emerging & Sustainable Technologies for Power & ICT in a Developing Society (NIGERCON)*, pp. 38-43. IEEE, 2013.
 18. Kalu, C., Ozuomba, Simeon. & Udofia, K. (2015). Web-based map mashup application for participatory wireless network signal strength mapping and customer support services. *European Journal of Engineering and Technology*, 3 (8), 30-43.
 19. Samuel, W., Ozuomba, Simeon, & Constance, K. (2019). SELF-ORGANIZING MAP (SOM) CLUSTERING OF 868 MHZ WIRELESS SENSOR NETWORK NODES BASED ON EGLI PATHLOSS MODEL COMPUTED RECEIVED SIGNAL STRENGTH. *Journal of Multidisciplinary Engineering Science and Technology (JMEST) Vol. 6 Issue 12, December - 2019*
 20. Johnson, Enyenihi Henry, Simeon Ozuomba, and Ifio Okon Asuquo. (2019). Determination of Wireless Communication Links Optimal Transmission Range Using Improved Bisection Algorithm. *Universal Journal of Communications and Network*, 7(1), 9-20.
 21. Uduak Idio Akpan, Constance Kalu, Simeon Ozuomba, Akaninyene Obot (2013). Development of improved scheme for minimising handoff failure due to poor signal quality. *International Journal of Engineering Research & Technology (IJERT)*, 2(10), 2764-2771
 22. Njoku, Felix A., Ozuomba Simeon, and Fina Otosi Faithpraise (2019). Development Of Fuzzy Inference System (FIS) For Detection Of Outliers In Data Streams Of Wireless Sensor Networks. *International Multilingual Journal of Science and Technology (IMJST) Vol. 4 Issue 10, October - 2019*
 23. Ogbonna Chima Otumdi , Ozuomba Simeon, Philip M. Asuquo (2020) Device Hardware Capacity And Rssi-Based Self Organizing Map Clustering Of 928 Mhz Lorawan Nodes Located In Flat Terrain With Light Tree Densities *Science and Technology Publishing (SCI & TECH) Vol. 4 Issue 9, September - 2020*
 24. Simeon, Ozuomba. (2020). "APPLICATION OF KMEANS CLUSTERING ALGORITHM FOR SELECTION OF RELAY NODES IN WIRELESS SENSOR NETWORK." *International Multilingual Journal of Science and Technology (IMJST) Vol. 5 Issue 6, June - 2020*
 25. Samuel, Wali, Simeon Ozuomba, and Philip M. Asuquo (2019). EVALUATION OF WIRELESS SENSOR NETWORK CLUSTER HEAD SELECTION FOR DIFFERENT PROPAGATION ENVIRONMENTS BASED ON LEE PATH LOSS MODEL AND K-MEANS ALGORITHM. EVALUATION, 3(11). *Science and Technology Publishing (SCI & TECH) Vol. 3 Issue 11, November - 2019*
 26. Ezenugu, I. A (2014), "Comparison Of The Impact Of Rain On Receiver G/T For The Horizontal, Vertical And Circular Polarization" *Journal of Multidisciplinary Engineering Science and Technology (JMEST) ISSN: 2458-9403 Vol. 1 Issue 5, pp 12750-12754*
 27. Simeon, Ozuomba. (2020). "Analysis Of Effective Transmission Range Based On Hata Model For Wireless Sensor Networks In The C-Band And Ku-Band." *Journal of Multidisciplinary Engineering Science and Technology (JMEST) Vol. 7 Issue 12, December - 2020*
 28. Ogbonna Chima Otumdi , Ozuomba Simeon, Kalu Constance (2020). Clustering Of 2100 Mhz Cellular Network Devices With Som Algorithm Using Device Hardware Capacity And Rssi Parameters *Science and Technology Publishing (SCI & TECH) Vol. 4 Issue 2, February – 2020*
 29. Ezenugu, I. A. (2019). "Ericson Model-Based Characterization of Cellular Network Path Loss for A Road Lined With Delonixregia Trees"

- International Multilingual Journal of Science and Technology, (IMJST) Vol. 4 No 9, pp 724-728.
30. Atakpo, F. K., Simeon, O., & Utibe-Abasi, S. B. (2021) A COMPARATIVE ANALYSIS OF SELFORGANIZING MAP AND K-MEANS MODELS FOR SELECTION OF CLUSTER HEADS IN OUT-OF-BAND DEVICE-TO-DEVICE COMMUNICATION. *Journal of Multidisciplinary Engineering Science Studies (JMESS)*.
 31. Akpan, Ito J., Ozuomba Simeon, and Kalu Constance (2020). "Development Of A Guard Channel-Based Prioritized Handoff Scheme With Channel Borrowing Mechanism For Cellular Networks." *Journal of Multidisciplinary Engineering Science and Technology (JMEST) Vol. 7 Issue 2, February - 2020*
 32. Ewona, I. and Ekah, U. (2021). Influence of Tropospheric Variables on Signal Strengths of Mobile Networks in Calabar, Nigeria. *Journal of Scientific and Engineering Research*, 8(9): 137-45.
 33. Obi, E., Ekah, U. and Ewona, I. (2021). Real-Time Assessment of Cellular Network Signal Strengths in Calabar. *International Journal of Engineering Sciences & Research Technology*, 10(7): 47-57.
 34. Lam, C. F., Liu, H., Koley, B., Zhao, X., Kamalov, V., & Gill, V. (2010). Fiber optic communication technologies: What's needed for datacenter network operations. *IEEE Communications Magazine*, 48(7), 32-39.
 35. Aba, R. (2014). Path loss prediction for gsm mobile networks for urban region of Aba South-East Nigeria. *Int. J. Comput. Sci. Mobile Comput.*, 3, 267-281.
 36. Cheerla, S., Ratnam, D. V., & Borra, H. S. (2018). Neural network-based path loss model for cellular mobile networks at 800 and 1800 MHz bands. *AEU-International Journal of Electronics and Communications*, 94, 179-186.
 37. Eunice, Akinloye Bolanle, and Simeon Ozuomba (2016) "Evaluation of the Distribution of Terrain Roughness Index for Terrestrial Line of Site Microwave Links in Uyo Metropolis." *Mathematical and Software Engineering 2.1* (2016): 9-18
 38. Mbaocha C. C, Eze C. U, Ezenugu, I. A, Onwumere J. C. (2016) "Satellite Model for Yaw-Axis Determination and Control Using PID Compensator" *International Journal Of Scientific & Engineering Research*, Vol 7, No 7, pp 1623-1629 <http://www.ijser.org>
 39. Akaninyene B. Obot , Ozuomba Simeon and Afolanya J. Jimoh (2011); "Comparative Analysis Of Pathloss Prediction Models For Urban Macrocellular" *Nigerian Journal of Technology (NIJOTECH)* Vol. 30, No. 3 , October 2011 , PP 50 – 59
 40. Dialoke, Ikenna Calistus, Ozuomba Simeon, and Henry Akpan Jacob. (2020) "ANALYSIS OF SINGLE KNIFE EDGE DIFFRACTION LOSS FOR A FIXED TERRESTRIAL LINE-OF-SIGHT MICROWAVE COMMUNICATION LINK." *Journal of Multidisciplinary Engineering Science and Technology (JMEST) Vol. 7 Issue 2, February - 2020*
 41. Ezenugu, I. A. (2019) "Link Budget Analysis for Non-Line-Of-Sight Microwave Communication Link With Single Knife Edge Diffraction Obstruction" *Journal of Multidisciplinary Engineering Science and Technology (JMEST) Vol. 6 No 9, pp 10773 – 10777.*
 42. Ononiwu, Gordon, Simeon Ozuomba, and Constance Kalu. (2015). Determination of the dominant fading and the effective fading for the rain zones in the ITU-R P. 838-3 recommendation. *European Journal of Mathematics and Computer Science Vol, 2(2)*.
 43. Kalu, C., Ozuomba, Simeon. & Jonathan, O. A. (2015). Rain rate trend-line estimation models and web application for the global ITU rain zones. *European Journal of Engineering and Technology*, 3 (9), 14-29.
 44. Simeon, Ozuomba. (2016) "Comparative Analysis Of Rain Attenuation In Satellite Communication Link For Different Polarization Options." *Journal of Multidisciplinary Engineering Science and Technology (JMEST) Vol. 3 Issue 6, June - 2016*
 45. Njoku Chukwudi Aloziem, Ozuomba Simeon, Afolayan J. Jimoh (2017) Tuning and Cross Validation of Blomquist-Ladell Model for Pathloss Prediction in the GSM 900 Mhz Frequency Band , *International Journal of Theoretical and Applied Mathematics*
 46. Ozuomba, Simeon, Johnson, E. H., & Udoiwod, E. N. (2018). Application of Weissberger Model for Characterizing the Propagation Loss in a *Gliricidia sepium* Arboretum. *Universal Journal of Communications and Network*, 6(2), 18-23.
 47. Oloyede Adams Opeyemi, Ozuomba Simeon, Constance Kalu (2017) Shibuya Method for Computing Ten Knife Edge Diffraction Loss. *Software Engineering 2017*; 5(2): 38-43
 48. Ozuomba, Simeon, Enyenihi, J., & Rosemary, N. C. (2018). Characterisation of Propagation Loss for a 3G Cellular Network in a Crowded Market Area Using CCIR Model. *Review of Computer Engineering Research*, 5(2), 49-56.
 49. Constance, Kalu, Ozuomba Simeon, and Ezuruike Okafor SF. (2018). Evaluation of the Effect of Atmospheric Parameters on Radio Pathloss in Cellular Mobile Communication System. Evaluation, 5(11). *Journal of Multidisciplinary Engineering Science and*

- Technology (JMEST) Vol. 5 Issue 11, November - 2018
50. Kalu Constance, Ozuomba Simeon, Umana, Sylvester Isreal (2018). Evaluation of Walficsh-Bertoni Path Loss Model Tuning Methods for a Cellular Network in a Timber Market in Uyo. *Journal of Multidisciplinary Engineering Science Studies (JMESS)* Vol. 4 Issue 12, December - 2018
 51. Ozuomba, Simeon, Henry Johnson Enyenihi, and Constance Kalu (2018) "Program to Determine the Terrain Roughness Index using Path Profile Data Sampled at Different Moving Window Sizes." *International Journal of Computer Applications* 975: 8887.
 52. Ekah, U. J. and Emeruwa C. (2021). Guaging of Key Performance Indicators for 2G Mobile Networks in Calabar, Nigeria. *World Journal of Advanced Research and Reviews*, 12(2): 157-163.
 53. Egbe Jesam Nna, Ozuomba Simeon, Enyenihi Henry Johnson (2017) Modelling and Application of Vertical Refractivity Profile for Cross River State. *World Journal of Applied Physics* 2017; 2(1): 19-26
 54. Ozuomba, Simeon, Constant Kalu, and Henry Johnson Enyenihi. (2018) "Comparative Analysis of the Circle Fitting Empirical Method and the International Telecommunication Union Parabola Fitting Method for Determination of the Radius of Curvature for Rounded Edge Diffraction Obstruction." *Communications on Applied Electronics (CAE)* 7: 16-21.
 55. Ekah. U. J., Iloke, J., Ewona, I. & Obi, E. (2022). Measurement and Performance Analysis of Signal-to-Interference Ratio in Wireless Networks. *Asian Journal of Advanced Research and Reports*, 16(3): 22-31.
 56. Imoh-Etefia, Ubon Etefia, Ozuomba Simeon, and Stephen Bliss Utibe-Abasi. (2020). "Analysis Of Obstruction Shadowing In Bullington Double Knife Edge Diffraction Loss Computation." *Journal of Multidisciplinary Engineering Science Studies (JMESS)* Vol. 6 Issue 1, January – 2020
 57. Simeon, Ozuomba, Kalu Constance, and Ezuruike Okafor SF. (2018). "Analysis of Variation in the Vertical Profile Of Radio Refractivity Gradient and its impact on the Effective Earth Radius Factor." *International Multilingual Journal of Science and Technology (IMJST)* Vol. 3 Issue 11, November - 2018
 58. Ewona, I., Ekah. U. J., Ikoi, A.O. & Obi, E. (2022). Measurement and Performance Assessment of GSM Networks using Received Signal Level. *Journal of Contemporary Research*, 1(1): 88-98.
 59. Ozuomba, Simeon. (2019). EVALUATION OF OPTIMAL TRANSMISSION RANGE OF WIRELESS SIGNAL ON DIFFERENT TERRAINS BASED ON ERICSSON PATH LOSS MODEL. *Science and Technology Publishing (SCI & TECH)* Vol. 3 Issue 12, December - 2019
 60. Johnson, Enyenihi Henry, Simeon Ozuomba, and Kalu Constance. (2019). Development of model for estimation of radio refractivity from meteorological parameters. *Universal Journal of Engineering Science* 7(1), 20-26.
 61. Simeon, Ozuomba, Ezuruike Okafor SF, and Bankole Morakinyo Olumide (2018). Development of Mathematical Models and Algorithms for Exact Radius of Curvature Used in Rounded Edge Diffraction Loss Computation. *Development*, 5(12). *Journal of Multidisciplinary Engineering Science and Technology (JMEST)* Vol. 5 Issue 12, December – 2018
 62. Ono, M. N., Obot, A. B., & Ozuomba, Simeon. (2020). ENHANCED BISECTION ITERATION METHOD APPLIED IN FADE MARGIN-BASED OPTIMAL PATH LENGTH FOR FIXED POINT TERRESTRIAL MICROWAVE COMMUNICATION LINK WITH KNIFE EDGE DIFFRACTION LOSS. *International Multilingual Journal of Science and Technology (IMJST)* Vol. 5 Issue 6, June – 2020
 63. Zion, Idongesit, Simeon Ozuomba, and Philip Asuquo. (2020) "An Overview of Neural Network Architectures for Healthcare." *2020 International Conference in Mathematics, Computer Engineering and Computer Science (ICMCECS)*. IEEE, 2020
 64. Ogu R.E, Chukwudebe G. A and Ezenugu, I. A (2016), "An IoT Based Tamper Prevention System for Electricity Meter" *American Journal of Engineering Research*, Vol 5 No 10, pp 347-353
 65. Yang, N., Wang, L., Geraci, G., Elkashlan, M., Yuan, J., & Di Renzo, M. (2015). Safeguarding 5G wireless communication networks using physical layer security. *IEEE Communications Magazine*, 53(4), 20-27.
 66. Yan, Y., Qian, Y., Sharif, H., & Tipper, D. (2012). A survey on smart grid communication infrastructures: Motivations, requirements and challenges. *IEEE communications surveys & tutorials*, 15(1), 5-20.
 67. Tounsi, W., & Rais, H. (2018). A survey on technical threat intelligence in the age of sophisticated cyber attacks. *Computers & security*, 72, 212-233.
 68. Gunduz, M. Z., & Das, R. (2020). Cyber-security on smart grid: Threats and potential solutions. *Computer networks*, 169, 107094.
 69. Callegati, F., Cerroni, W., & Ramilli, M. (2009). Man-in-the-Middle Attack to the HTTPS Protocol. *IEEE Security & Privacy*, 7(1), 78-81.
 70. Conti, M., Dragoni, N., & Lesyk, V. (2016). A survey of man in the middle attacks. *IEEE*

- communications surveys & tutorials*, 18(3), 2027-2051.
71. Koscher, K., Czeskis, A., Roesner, F., Patel, S., Kohno, T., Checkoway, S., ... & Savage, S. (2010, May). Experimental security analysis of a modern automobile. In *2010 IEEE symposium on security and privacy* (pp. 447-462). IEEE.
 72. Bulgurcu, B., Cavusoglu, H., & Benbasat, I. (2010). Information security policy compliance: an empirical study of rationality-based beliefs and information security awareness. *MIS quarterly*, 523-548.
 73. Fischer, C. (2008). Feedback on household electricity consumption: a tool for saving energy?. *Energy efficiency*, 1(1), 79-104.
 74. Bhuyan, M. H., Bhattacharyya, D. K., & Kalita, J. K. (2013). Network anomaly detection: methods, systems and tools. *Ieee communications surveys & tutorials*, 16(1), 303-336.
 75. Oladeji, F. A., Akeredolu, E. I., Komolafe, O., & Oyetunji, M. O. An Enhanced Wireless Network Security Framework for Federal Road Safety Corps in Nigeria.
 76. Carranza, A., Mayorga, D., DeCusatis, C., & Rahemi, H. (2018, July). Comparison of wireless network penetration testing tools on desktops and raspberry Pi platforms. In *Proceedings of the LACCEI international Multi-conference for Engineering, Education and Technology, Boca Raton, FL, USA* (pp. 18-20).
 77. Fadhilah, D., & Marzuki, M. I. (2020, September). Performance analysis of ids snort and ids suricata with many-core processor in virtual machines against dos/ddos attacks. In *2020 2nd International Conference on Broadband Communications, Wireless Sensors and Powering (BCWSP)* (pp. 157-162). IEEE.
 78. Muhati, E., & Rawat, D. B. (2021). Hidden-Markov-model-enabled prediction and visualization of cyber agility in IoT era. *IEEE Internet of Things Journal*, 9(12), 9117-9127.
 79. Reyes, F., Fuertes, W., Tapia, F., Toulkeridis, T., Aules, H., & Pérez, E. (2019). A BI solution to identify vulnerabilities and detect real-time cyber-attacks for an academic CSIRT. In *Intelligent Computing: Proceedings of the 2018 Computing Conference, Volume 2* (pp. 1135-1153). Springer International Publishing.
 80. Kozik, R., & Choraś, M. (2014). Machine learning techniques for cyber attacks detection. In *Image Processing and Communications Challenges 5* (pp. 391-398). Springer International Publishing.
 81. VanSickle, R., Abegaz, T., & Payne, B. (2019). Effectiveness of tools in identifying rogue access points on a wireless network.
 82. Suciu, G., Anwar, M., & Istrate, C. (2019). Mobile Application and Wi-Fi Network Security for e-Learning Platforms. *eLearning & Software for Education*, 1.
 83. Salah, K. (2014, March). Harnessing the cloud for teaching cybersecurity. In *Proceedings of the 45th ACM technical symposium on Computer science education* (pp. 529-534).
 84. Pourmirza, Z., & Srivastava, A. (2020, August). Cybersecurity Analysis for the Communication Protocol in Smart Grids. In *2020 IEEE 8th International Conference on Smart Energy Grid Engineering (SEGE)* (pp. 58-63). IEEE.
 85. Natarajan, J. (2020). Cyber secure man-in-the-middle attack intrusion detection using machine learning algorithms. In *AI and Big Data's Potential for Disruptive Innovation* (pp. 291-316). IGI global.
 86. Yang, Y., McLaughlin, K., Littler, T., Sezer, S., Im, E. G., Yao, Z. Q., ... & Wang, H. F. (2012). Man-in-the-middle attack test-bed investigating cyber-security vulnerabilities in smart grid SCADA systems.
 87. Toutsop, O., Harvey, P., & Kornegay, K. (2020, October). Monitoring and detection time optimization of man in the middle attacks using machine learning. In *2020 IEEE Applied Imagery Pattern Recognition Workshop (AIPR)* (pp. 1-7). IEEE.