Performance Evaluation and Analysis of various Network Security tools

Sinchana K, Sinchana C, Gururaj H L, Vidyavardhaka College of Engineering, Mysore,
Sunil Kumar B R, Adichunchanagiri University, Mandya

Abstract— **With the developments in the internet and the rise of ecommerce applications and social networks, there is a lot of data generated daily. Thus, the transmission of this data safely through the web is an issue. Also, network security is gaining a greater importance as more and more people are moving towards the age of digital information. As the number of users increase, the cyber-attacks are becoming critical. There are various network security tools which deal with a lot of functions in contributing to the security of network. These network security tools help in finding the vulnerabilities in a website or a web application, to secure wi-fi networks, to crack passwords, to ensure message encryption etc. This paper describes some of such tools, which deal with vulnerability check in a website, wi-fi security, network scanning, network or protocol analyzing.**

Index Terms**—Network security, Zed Attack Proxy, Aircrack-ng, Wireshark, NMAP**

I. Introduction

THE fast developments in the internet has attracted a huge number of users, increasing the data production rate. As more and more people are moving towards the digitalized information, this information has to be safely transmitted. The safe transmission of such data is an issue. Along with this, the network security has got a greater importance as there is an increase in the number of users day by day. Due to increase in the number of users, cyber-attacks have become a major threat[1].

Thus, in order to ensure the safe transmission of data and network security, there are various network security tools available. These network security tools perform different functions. There are various types of tools which are used for different purposes[2]. These tools perform variety of operations to contribute to the network security.

Some of them help in sniffing the network or analyzing a protocol, some of them deal in securing a wi-fi network, some of them deal with password security, some check for the website security etc.

In this paper, we describe four such tools which perform certain tasks contributing to the network security[3]. We discuss about Zed Attack Proxy (ZAP) tool which deals in checking for the vulnerabilities in the websites or web application. How the threats in the websites are discovered and solutions are found using ZAP.

Then we discuss about Aircrack-ng tool which deals with password cracking of Wi-fi network. How this helps in discovering the effectiveness in the security of wi-fi network.

Next, we discuss about the NMAP tool that is Network mapper which is a network scanner that helps in finding out the hosts and servers in a computer network. It sends packets and analyses the responses.

The next tool is Wireshark tool, which is a packet analyzer. It is used for network troubleshooting and analyzing the network, development of software and communication protocols.

We discuss how these tools work and their important contributions to ensure network security.

II. Zed Attack Proxy

Zed Attack Proxy is an open source web application security scanner which is a free network security tool. It is an OWASP (Open Web Application Security Project) with the flagship status. It is completely free and open source[4]. ZAP is used for penetration testing of your website or a web application. It is ideal for the use of both beginners to the application security as well as professional penetration testers. It is becoming a framework for advanced testing.

A. ZAP PRINCIPLES

- It is free, open source which means anyone can access and use it.
- It's also cross platform that is it works on Linux, Windows and Macs.
- It is easy to use; any user be it beginner or a professional can use it easily.
- It is easy to install; it requires Java Runtime Environment but everything else is included in standard downloads.
- It is fully internationalized, i.e., it's been translated to many other languages.
- It is fully documented, set of useful documents and helpline files are included.
- It works well with other tools, so that we can use other tools in conjunction with zap if we need to.
- It supports the use of well-regarded components.

978-1-7281-1261-9/19/$31.00 ©2019 IEEE

B. ZAP MAIN FEATURES

- *Intercepting proxy*: we configure the browser to proxy through zap so that zap can see through all the requests and responses.
- *Active and passive scanners*: it provides both the active and passive scanners, so that passive scanners just examine through requests and responses and still recognise the types of problems just on that basis. But the active scanners are different and these can perform wide range of attacks only on applications that you have permission to test.
- *Spider:* it is used to crawl the application, say for example to find the missed-out pages or hidden pages.
- *Report generation*: zap can prepare reports based on the issues found including more advices and links on how to solve that particular problem.
- *Brute force component*: it can also find files even if there are no links to them using brute force component, which is based on owasp dirbuster tool.
- *Fuzzing*: it can also fuzz parameters and includes fuzzing libraries from the jbrofuzz and fuzz db tools. You can use fuzzing to find more subtle vulnerabilities that the automated scanners cannot find.
- *Extensibility*: zap can be easily extended

Any web application has to be tested and security testing plays a vital role in it. There are many security threats that a website or a web application may face, like SQL injection, Broken authentication and session management, cross sight scripting, broken access control, security misconfiguration, sensitive data exposure, insufficient data protection, cross site request forgery etc. Zap helps in detecting all such threats and vulnerabilities, and possibly give solutions to the problems[5].

C. WORKING OF ZAP

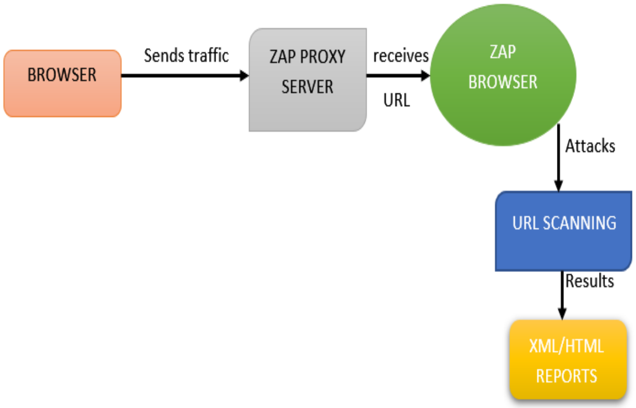


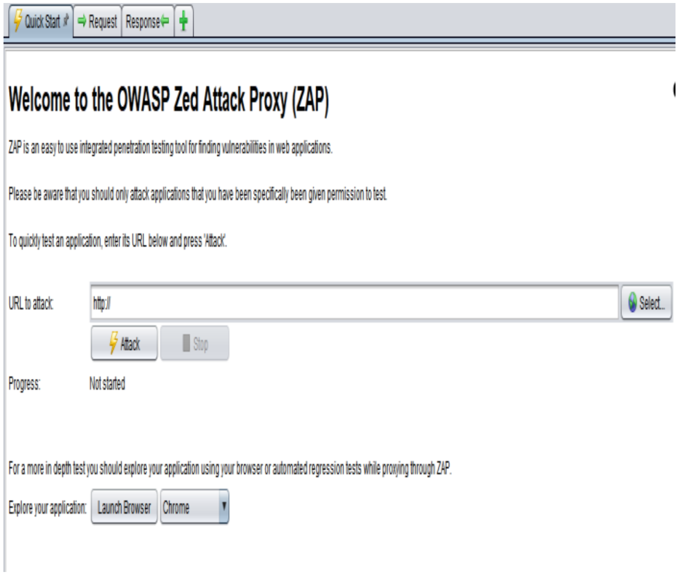
Fig. 1. Block diagram of working of ZAP

Initially, zap creates a proxy server. The browser sends the traffic to pass through the zap proxy server, then the zap browser receives the URL and the attacks on URL with help of auto scanners that helps to find out the vulnerabilities in a website. Finally, a report is generated with the respective problem, its reason and also the solution for it.

E. A SIMPLE PENETRATION TEST

Firstly, configure your browser to proxy through ZAP. Then, explore the application manually. Next, use spider to find 'hidden' content. See what issues the passive scanner has found. Finally, use the active scanner to find vulnerabilities

F. WORKING SCENARIO



Welcome to the OWASP Zed Attack Proxy (ZAP)

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

To quickly test an application, enter its URL below and press 'Attack'.

URL to attack: http://

Progress: Not started

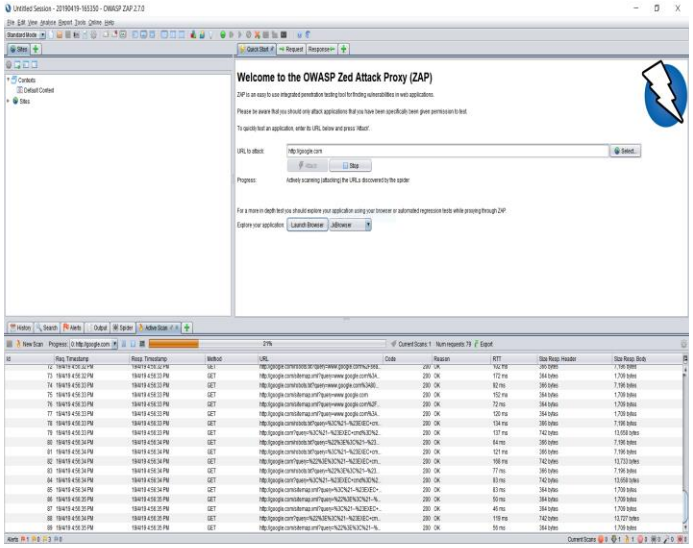
For a more in depth test you should explore your application using your browser or automated regression tests while proxying through ZAP.

Explore your application: Launch Browser Chrome

Fig. 2. Initial window of ZAP before entering URL to attack

Initially the zap window looks like this as shown in the above figure.

We give a URL to attack in order to check for the possible vulnerabilities and then the scanning process starts. It is as shown in the below figure.



Welcome to the OWASP Zed Attack Proxy (ZAP)

ZAP is an easy to use integrated penetration testing tool for finding vulnerabilities in web applications.

Please be aware that you should only attack applications that you have been specifically been given permission to test.

To quickly test an application, enter its URL below and press 'Attack'.

URL to attack: http://google.com

Progress: Actively scanning (attacking) the URLs discovered by the spider

For a more in depth test you should explore your application using your browser or automated regression tests while proxying through ZAP.

Explore your application: Launch Browser JxBrowser

Progress: 0: http://google.com 21% Current Scans: 1 Num requests: 79

Id	Req. Timestamp	Resp. Timestamp	Method	URL	Code	Reason	RTT	Size Resp. Header	Size Resp. Body
72	19/4/19 4:56:32 PM	19/4/19 4:56:32 PM	GET	http://google.com/robots.txt?query=www.google.com%2F568...	200	OK	102 ms	386 bytes	7,196 bytes
73	19/4/19 4:56:32 PM	19/4/19 4:56:33 PM	GET	http://google.com/sitemap.xml?query=www.google.com%3A...	200	OK	172 ms	384 bytes	1,709 bytes
74	19/4/19 4:56:33 PM	19/4/19 4:56:33 PM	GET	http://google.com/robots.txt?query=www.google.com%3A80...	200	OK	82 ms	386 bytes	7,196 bytes
75	19/4/19 4:56:33 PM	19/4/19 4:56:33 PM	GET	http://google.com/sitemap.xml?query=www.google.com	200	OK	152 ms	384 bytes	1,709 bytes
76	19/4/19 4:56:33 PM	19/4/19 4:56:33 PM	GET	http://google.com/sitemap.xml?query=www.google.com%2F...	200	OK	72 ms	384 bytes	1,709 bytes
77	19/4/19 4:56:33 PM	19/4/19 4:56:33 PM	GET	http://google.com/sitemap.xml?query=www.google.com%3A...	200	OK	120 ms	384 bytes	1,709 bytes
78	19/4/19 4:56:33 PM	19/4/19 4:56:33 PM	GET	http://google.com/robots.txt?query=%3C%21--%23EXEC+cm...	200	OK	134 ms	386 bytes	7,196 bytes
79	19/4/19 4:56:33 PM	19/4/19 4:56:33 PM	GET	http://google.com?query=%3C%21--%23EXEC+cmd%3D%2...	200	OK	137 ms	742 bytes	13,658 bytes
80	19/4/19 4:56:34 PM	19/4/19 4:56:34 PM	GET	http://google.com/robots.txt?query=%22%3E%3C%21--%23...	200	OK	84 ms	386 bytes	7,196 bytes
81	19/4/19 4:56:34 PM	19/4/19 4:56:34 PM	GET	http://google.com/robots.txt?query=%3C%21--%23EXEC+cm...	200	OK	121 ms	386 bytes	7,196 bytes
82	19/4/19 4:56:34 PM	19/4/19 4:56:34 PM	GET	http://google.com?query=%22%3E%3C%21--%23EXEC+cm...	200	OK	168 ms	742 bytes	13,733 bytes
83	19/4/19 4:56:34 PM	19/4/19 4:56:34 PM	GET	http://google.com/robots.txt?query=%22%3E%3C%21--%23...	200	OK	77 ms	386 bytes	7,196 bytes
84	19/4/19 4:56:34 PM	19/4/19 4:56:34 PM	GET	http://google.com?query=%3C%21--%23EXEC+cmd%3D%2...	200	OK	83 ms	742 bytes	13,658 bytes
85	19/4/19 4:56:34 PM	19/4/19 4:56:34 PM	GET	http://google.com/sitemap.xml?query=%3C%21--%23EXEC+...	200	OK	83 ms	384 bytes	1,709 bytes
86	19/4/19 4:56:35 PM	19/4/19 4:56:35 PM	GET	http://google.com/sitemap.xml?query=%22%3E%3C%21--%...	200	OK	50 ms	384 bytes	1,709 bytes
87	19/4/19 4:56:35 PM	19/4/19 4:56:35 PM	GET	http://google.com/sitemap.xml?query=%3C%21--%23EXEC+...	200	OK	46 ms	384 bytes	1,709 bytes
88	19/4/19 4:56:34 PM	19/4/19 4:56:35 PM	GET	http://google.com?query=%22%3E%3C%21--%23EXEC+cm...	200	OK	119 ms	742 bytes	13,727 bytes
89	19/4/19 4:56:35 PM	19/4/19 4:56:35 PM	GET	http://google.com/sitemap.xml?query=%22%3E%3C%21--%...	200	OK	56 ms	384 bytes	1,709 bytes

Fig. 3. The scanning process in ZAP.

978-1-7281-1261-9/19/$31.00 ©2019 IEEE

Once the scanning process is done, the threats or the vulnerabilities discovered through scanning are listed in the alert section.

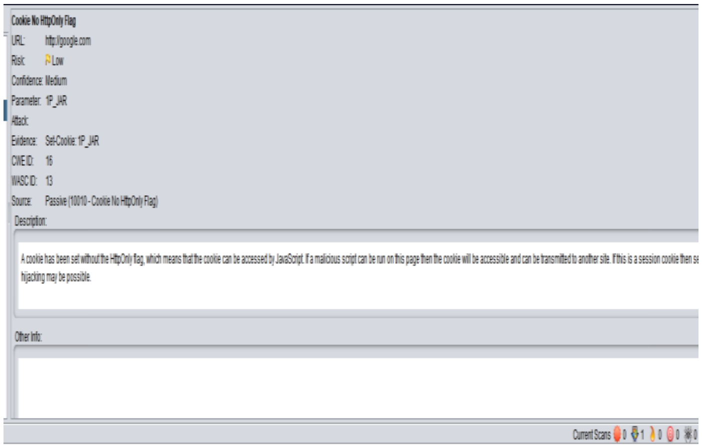
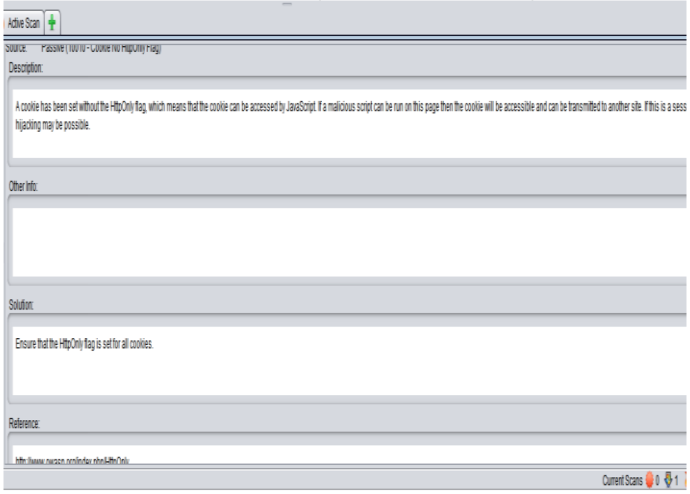

Cookie No HttpOnly Flag
URL: http://google.com
Risk: Low
Confidence: Medium
Parameter: 1P_JAR
Attack:
Evidence: Set-Cookie: 1P_JAR
CWE ID: 16
WASC ID: 13
Source: Passive (10010 - Cookie No HttpOnly Flag)
Description:
A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a session cookie then se
hijacking may be possible.
Other Info:

Fig. 4. The description, type and other details of a threat scanned in scanning process.

The description of the problem and also the possible solution for the threat is also given as shown in figure.



Source: Passive (10010 - Cookie No HttpOnly Flag)
Description:
A cookie has been set without the HttpOnly flag, which means that the cookie can be accessed by JavaScript. If a malicious script can be run on this page then the cookie will be accessible and can be transmitted to another site. If this is a sess
hijacking may be possible.
Other Info:
Solution:
Ensure that the HttpOnly flag is set for all cookies.
Reference:
http://www.owasp.org/index.php/HttpOnly

Fig. 5. The solution for the problem given after the description of the vulnerability.

Thus, Zed Attack Proxy is an effective tool to ensure security of the web applications or websites.

III. AIRCRACK-NG

A. FEATURES

It contains a suite of tools for the purpose of checking wireless networks. It is mainly used for:

- *Monitoring:* Packet capturing and exporting of data to text files which is further used for processing by third party tools
- *Attacking:* Replaying attacks, de-authentication, fake access points and others through packet injection
- *Testing:* Checking driver capabilities and Wi-Fi cards (capture and injection)
- *Cracking:* WEP and WPA PSK

It contains four necessary parts [6]:

- *Airodump-ng:* it records the coordinates of the entry or access points and captures the raw 802.11 packets.
- *Aireplay-ng:* it is used for injecting frames into wireless networks.
- *Aircrack-ng:* it is used for recovering keys once enough packets have been captured.
- *Airdecap-ng:* it is used for decrypting the encrypted capture files.

Aircrack-ng is used mainly to test the weaknesses of wireless networks by accessing the network using WEP and WPA-PSK keys by decrypting encrypted packets stored.

B. WORKING OF AIRCRACK-NG

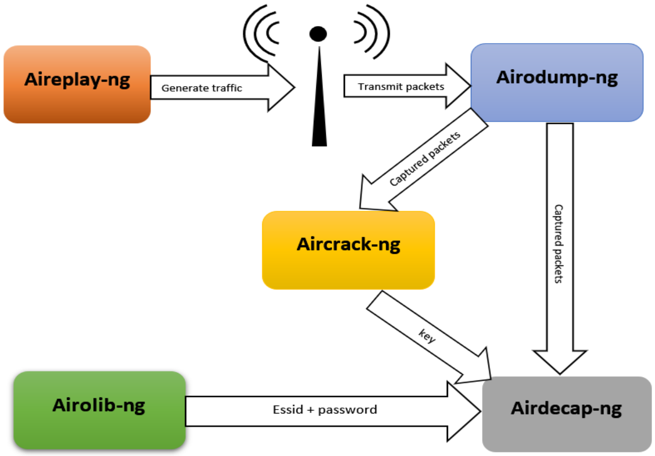


Fig. 6. Block diagram of working of aircrack-ng

Firstly , the aireplay-ng tool generates packets for simulating the traffic. Then the traffic generated, transmits packets and sends it to the airodump-ng tool. The airodump-ng tool captures the packets and then classifies them. Along with this tool, airmon-ng tool changes the adapter to monitor mode to capture all the packets[7]. Once, enough packets are captured, aircrack-ng recovers the key from captured packets. Finally, the airdecap-ng starts decrypting the contents of packets using the recovered key. The airolib-ng tool helps in fastening the password cracking process by precomputing the pairwise master keys[8].

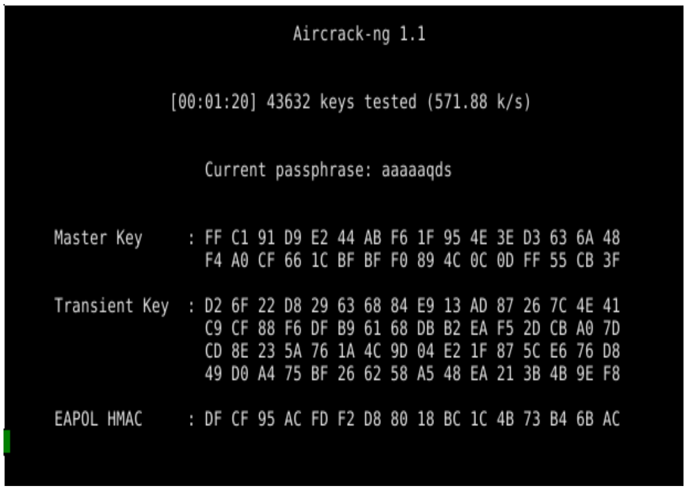
C. WORKING SCENARIO

The password is cracked by checking out all the possibilities in the password list. The output during the cracking process is as shown in fig. 7.

Once the password is cracked, the aircrack-ng tool shows the respective password with the message as “KEY FOUND!”.

The following figures show the output screen while aircracking takes place.

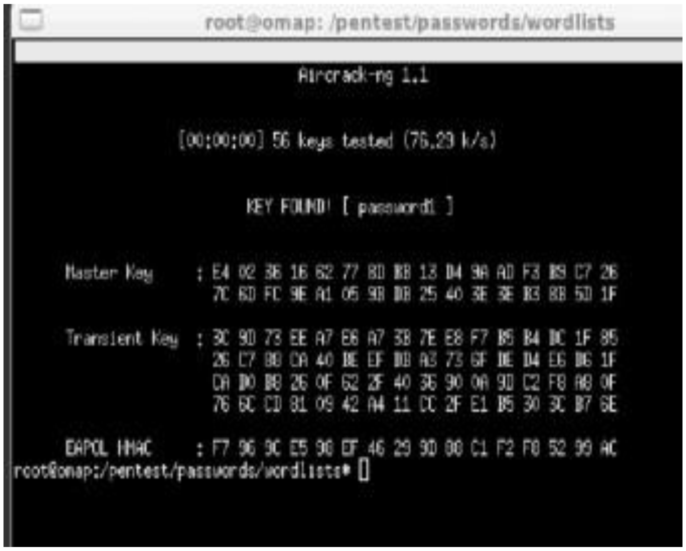
The current passphrase represents the currently parsed password in the password list [9].



```
                         Aircrack-ng 1.1

              [00:01:20] 43632 keys tested (571.88 k/s)

                  Current passphrase: aaaaaqds

  Master Key     : FF C1 91 D9 E2 44 AB F6 1F 95 4E 3E D3 63 6A 48
                   F4 A0 CF 66 1C BF BF F0 89 4C 0C 0D FF 55 CB 3F

  Transient Key  : D2 6F 22 D8 29 63 68 84 E9 13 AD 87 26 7C 4E 41
                   C9 CF 88 F6 DF B9 61 68 DB B2 EA F5 2D CB A0 7D
                   CD 8E 23 5A 76 1A 4C 9D 04 E2 1F 87 5C E6 76 D8
                   49 D0 A4 75 BF 26 62 58 A5 48 EA 21 3B 4B 9E F8

  EAPOL HMAC     : DF CF 95 AC FD F2 D8 80 18 BC 1C 4B 73 B4 6B AC
```

Fig. 7. Output screen while checking for password in the password list.

Once the password is cracked, the output screen is as shown.



root@omap: /pentest/passwords/wordlists

```
                         Aircrack-ng 1.1

              [00:00:00] 56 keys tested (76.23 k/s)

                     KEY FOUND! [ password1 ]

  Master Key     : E4 02 36 16 62 77 8D 8B 13 D4 9A AD F3 B9 C7 26
                   7C 6D FC 9E A1 05 9B DB 25 40 3E 3E B3 8B 5D 1F

  Transient Key  : 3C 9D 73 EE A7 E6 A7 3B 7E E8 F7 B5 B4 DC 1F 85
                   26 C7 8B CA 40 BE EF DB A3 73 6F DE D4 E6 B6 1F
                   CA DD B8 26 0F 62 2F 40 36 90 0A 9D C2 F8 A8 0F
                   76 6C CD 81 09 42 A4 11 CC 2F E1 B5 30 3C B7 6E

  EAPOL HMAC     : F7 96 9C E5 96 DF 46 29 9D 08 C1 F2 F8 52 99 AC
root@omap:/pentest/passwords/wordlists# []
```

Fig. 8. The final output screen when the password is cracked with the message as KEY FOUND and the respective password.

IV. NMAP

Nmap is basically known as foot printing tool or recognisant tool. This is the first step to find information about the target or the IP or the website. Nmap in normally the tool used with Linux, OS and the nmap for windows is known as zenmap.

Nmap is expanded as Network Mapper. It uses innovative raw IP packets to determine which hosts are available on the network, which hosts the services (application name and version), the operating system (and OS versions) that they work on, what packet filters or firewalls are in use and many other characters. It was initially designed to scan large networks quickly. Vulnerability detection, penetration testing, host discovery. The main use is for network exploration and security audits. We will also get to know what the flaws in our network are. It will help us find available ports and services. It will likewise be utilized for finding and abusing vulnerabilities in a system[10]. Nmap supports dozens of advanced techniques to map the network filled with IP filters, firewalls, routers and other hurdles. Many post scanning procedures (both UDP and TCP), OS detection, version detection, ping scraping etc. Nmap is a port scanner and is a tool used by system administrators and attackers (s) to identify vulnerability in operating systems[11]. This takes an IP address or host name and finds the source information associated with it. If an IP address is provided, it then finds the host to which it belongs to[12]. Nmap comes into use when we need to scan huge networks, nearly hundreds of thousands of machines. Nmap is a tool that can be used to find services that operate on Internet connection systems. It could potentially be used for black hat hacking as any other device, as precursor attempts to gain unauthorized access to computer equipment. Nmap is often used by security and systems administration to access the network for impairment.

A. WORKING OF NMAP

Traditional command line and graphical (GUI) versions are available to suit our priorities. Binaries are also available for those who do not want to compile the NMP from the source.Nmap can be used for advanced ethical hacking as well as for scanning a network by using a command line “nmap –v –A target host”.

In the fig 9 the TCP scanning that is done using Nmap. First the command line “nmap –sS 192.168.86.1” is used to syn scan the TCP. This command sends a syn message to the target system (Responder in figure1) using the TCP IP stack, then the responder sends a syn+ack message to the initiator and the initiator sends the ack message which confirms the 3-way hand shake.

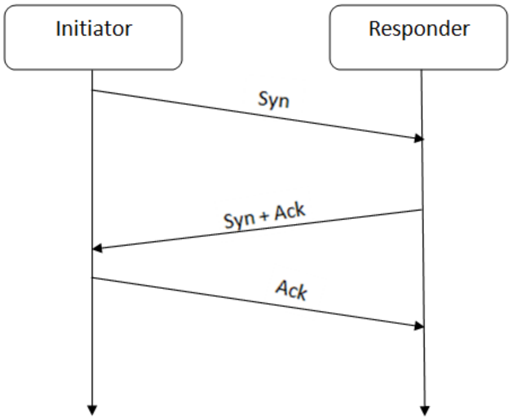


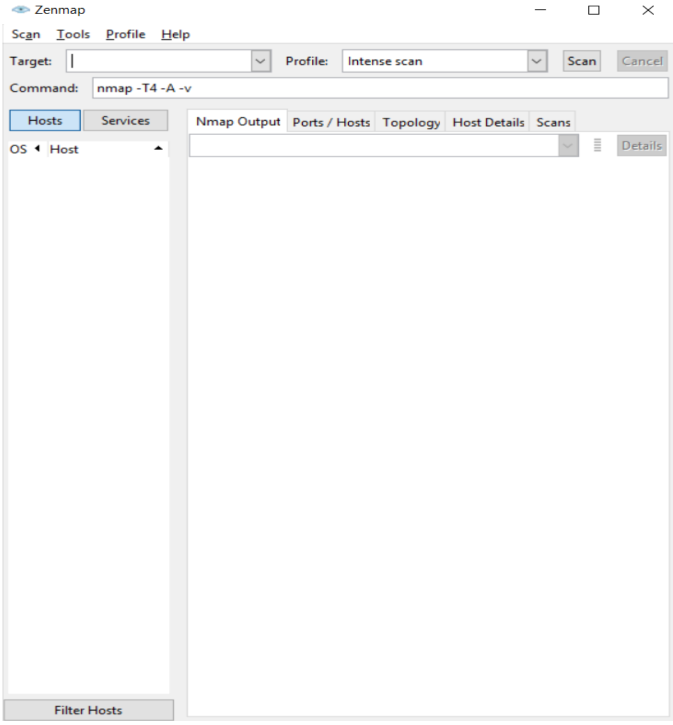
Fig. 9. TCP scanning using nmap.

B. WORKING SCENARIO

Fig 10 shows the initial window of Nmap. Since Nmap is a scanning tool let us take the example of the tcp scanning and show the working scenario of Nmap.

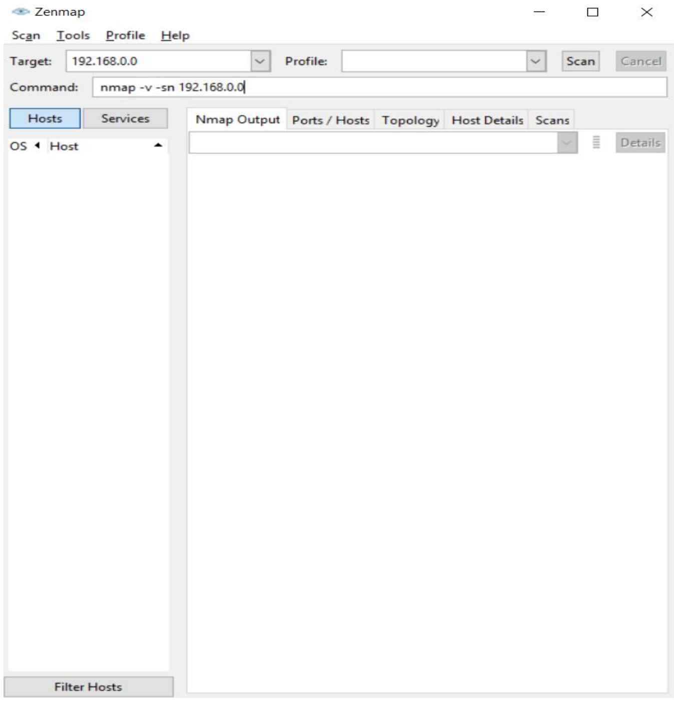
Fig 10 is the initial window of the nmap and this is how it looks when we first open the window. To scan the TCP IP, we need to enter the command line “nmap –v –sn 192.168.0.0” in the command space as shown in fig 11. And once the scanning of the TCP is done the final window looks like the fig 12 which is the window that shows the entire scanning of TCP.

This is basically just an example to make us understand about the working of nmap. Similarly using the command lines, we can scan any network as shown in the example.



Target:
Profile: Intense scan
Command: nmap -T4 -A -v

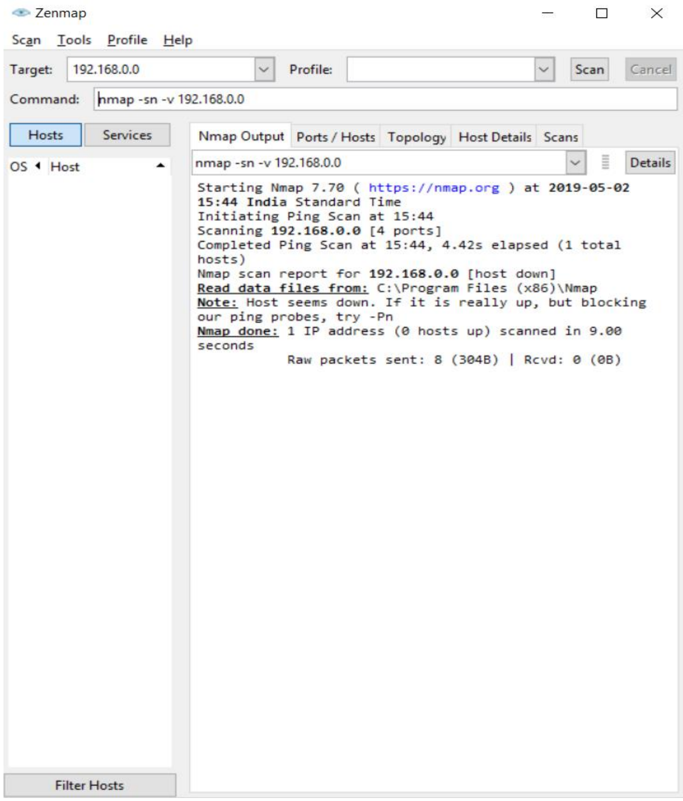
Fig 10. Initial window of nmap.



Target: 192.168.0.0
Profile:
Command: nmap -v -sn 192.168.0.0

Fig 11: command entering

Command line should be entered to scan the TCP, this is done by using the command line "nmap –v –sn 192.168.0.0"



Target: 192.168.0.0
Profile:
Command: nmap -sn -v 192.168.0.0

nmap -sn -v 192.168.0.0

```
Starting Nmap 7.70 ( https://nmap.org ) at 2019-05-02
15:44 India Standard Time
Initiating Ping Scan at 15:44
Scanning 192.168.0.0 [4 ports]
Completed Ping Scan at 15:44, 4.42s elapsed (1 total
hosts)
Nmap scan report for 192.168.0.0 [host down]
Read data files from: C:\Program Files (x86)\Nmap
Note: Host seems down. If it is really up, but blocking
our ping probes, try -Pn
Nmap done: 1 IP address (0 hosts up) scanned in 9.00
seconds
           Raw packets sent: 8 (304B) | Rcvd: 0 (0B)
```

Fig 12:Final window after scanning.

V. WIRESHARK

Wireshark is an open-source and open packet analyzer. It is used for network analysis, software and communications protocol development, troubleshooting and education. It puts the network card into promiscuous mode. This basically means it can intercept packets into the network, it is done by using libraries called as winpcap and libpcap. These winpcap and libpcap provide capture and filter capabilities on different platforms. Wireshark is a free and open-source protocol/ packet tracer[13]. Network attacks are almost identifiable by observing incoming and outgoing traffic, because unusual behavior is caused by the suspicious pattern of packets[14].

A. FEATURES OF WIRESHARK

- The main feature of wireshark is, it has live capture and offline analysis.
- It is a multiplatform i.e., it runs on multiple flatforms like windows, OS, linux etc.
- Captured network data can be browsed through a graphical interface.
- For dissecting new protocols, plug-ins can be created.
- Most importantly it is a data capturing tool and a packet analyzer.

B. WORKING OF WIRESHARK

The first thing we see is the list of all the network interfaces. We have to choose which interface we want wireshark to capture on. Most of the time it will be the Ethernet or Wi-Fi interface. Once we select the interface the wireshark starts capturing the packets.

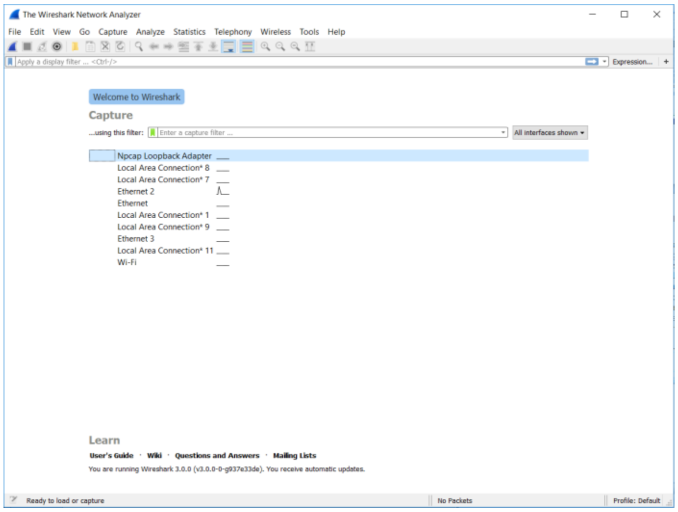
Let us take an example of ping packet. Ping packets are a type of internet control message or part of the ICMP protocol. We can ask the wireshark to give us only ICMP packets.

The top part of the screen wireshark shows a list of all the packets seen and gives a brief information about them.
The middle section of the wireshark window shows the lines that corresponds to one of the layers of the internet. Each of the layer is represented by the header that is contained by that packet. So, whenever the wireshark intercepts the packet it looks at all of the header on that packet so that we get complete information about the packet-ed every step of the network stack.

The bottom most part of the wireshark window shows the actual bits of the packets and what their encoding means.

C. *WORKING SCENARIO*

Fig 13. shows the initial window of wireshark where all the network interfaces are present.


The Wireshark Network Analyzer
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Apply a display filter ... <Ctrl-/>
Expression...
Welcome to Wireshark
Capture
...using this filter:
Enter a capture filter ...
All interfaces shown

Npcap Loopback Adapter
Local Area Connection* 8
Local Area Connection* 7
Ethernet 2
Ethernet
Local Area Connection* 1
Local Area Connection* 9
Ethernet 3
Local Area Connection* 11
Wi-Fi

Learn
User's Guide · Wiki · Questions and Answers · Mailing Lists

You are running Wireshark 3.0.0 (v3.0.0-0-g937e33de). You receive automatic updates.

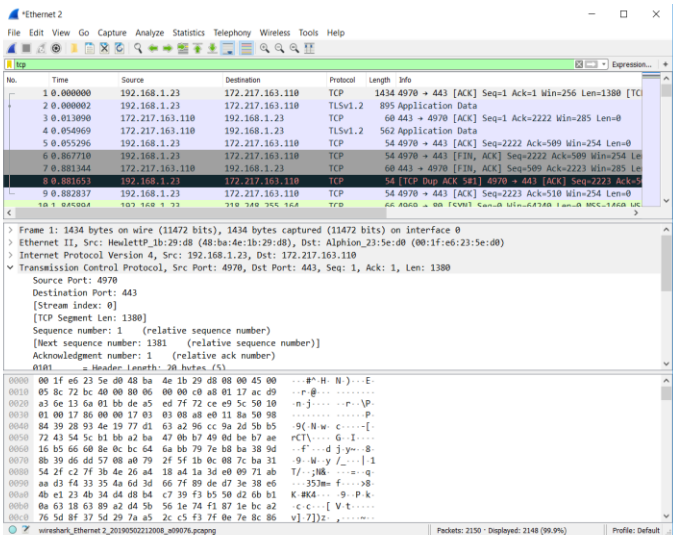
Ready to load or capture | No Packets | Profile: Default

Fig 13: initial window of wireshark.

Fig 14 shows that the packets are being scanned. Here as for the example we have taken i.e., TCP, the packets are being scanned for that.

We can also get the input output graph for this TCP packet analyzer. This option is found in the statistics option and the graph is as shown in fig 15.

Statistics also has many options such as packet length, end points, resolved addresses etc and one such option is capture file properties and this is shown in fig 16.


*Ethernet 2
File Edit View Go Capture Analyze Statistics Telephony Wireless Tools Help
Expression...

tcp

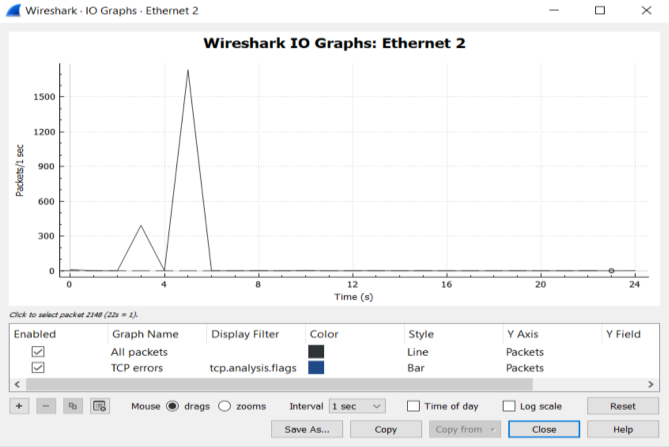
```
No.  Time      Source           Destination      Protocol Length Info
   1 0.000000  192.168.1.23     172.217.163.110  TCP      1434 4970 → 443 [ACK] Seq=1 Ack=1 Win=256 Len=1380 [TC
   2 0.000002  192.168.1.23     172.217.163.110  TLSv1.2   895 Application Data
   3 0.013090  172.217.163.110  192.168.1.23     TCP        60 443 → 4970 [ACK] Seq=1 Ack=2222 Win=285 Len=0
   4 0.054969  172.217.163.110  192.168.1.23     TLSv1.2   562 Application Data
   5 0.055296  192.168.1.23     172.217.163.110  TCP        54 4970 → 443 [ACK] Seq=2222 Ack=509 Win=254 Len=0
   6 0.867710  192.168.1.23     172.217.163.110  TCP        54 4970 → 443 [FIN, ACK] Seq=2222 Ack=509 Win=254 Le
   7 0.881344  172.217.163.110  192.168.1.23     TCP        60 443 → 4970 [FIN, ACK] Seq=509 Ack=2223 Win=285 Le
   8 0.881653  192.168.1.23     172.217.163.110  TCP        54 [TCP Dup ACK 5#1] 4970 → 443 [ACK] Seq=2223 Ack=5
   9 0.882837  192.168.1.23     172.217.163.110  TCP        54 4970 → 443 [ACK] Seq=2223 Ack=510 Win=254 Len=0
  10 1.845894  192.168.1.23     218.248.255.164  TCP        66 4969 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS
```

```
> Frame 1: 1434 bytes on wire (11472 bits), 1434 bytes captured (11472 bits) on interface 0
> Ethernet II, Src: HewlettP_1b:29:d8 (48:ba:4e:1b:29:d8), Dst: Alphion_23:5e:d0 (00:1f:e6:23:5e:d0)
> Internet Protocol Version 4, Src: 192.168.1.23, Dst: 172.217.163.110
v Transmission Control Protocol, Src Port: 4970, Dst Port: 443, Seq: 1, Ack: 1, Len: 1380
    Source Port: 4970
    Destination Port: 443
    [Stream index: 0]
    [TCP Segment Len: 1380]
    Sequence number: 1    (relative sequence number)
    [Next sequence number: 1381    (relative sequence number)]
    Acknowledgment number: 1    (relative ack number)
    0101 .... = Header Length: 20 bytes (5)
```

```
0000  00 1f e6 23 5e d0 48 ba  4e 1b 29 d8 08 00 45 00   ···#^·H· N·)···E·
0010  05 8c 72 bc 40 00 80 06  00 00 c0 a8 01 17 ac d9   ··r·@··· ········
0020  a3 6e 13 6a 01 bb de a5  ed 7f 72 ce e9 5c 50 10   ·n·j···· ··r··\P·
0030  01 00 17 86 00 00 17 03  03 08 a8 e0 11 8a 50 98   ········ ······P·
0040  84 39 28 93 4e 19 77 d1  63 a2 96 cc 9a 2d 5b b5   ·9(·N·w· c····-[·
0050  72 43 54 5c b1 bb a2 ba  47 0b b7 49 0d be b7 ae   rCT\···· G··I····
0060  16 b5 66 60 8e 0c bc 64  6a bb 79 7e b8 ba 38 9d   ··f`···d j·y~··8·
0070  8b 39 d6 dd 57 08 a0 79  2f 5f 1b 0c 08 7c ba 31   ·9··W··y /_···|·1
0080  54 2f c2 7f 3b 4e 26 a4  18 a4 1a 3d e0 09 71 ab   T/··;N&· ···=··q·
0090  aa d3 f4 33 35 4a 6d 3d  66 7f 89 de d7 3e 38 e6   ···35Jm= f····>8·
00a0  4b e1 23 4b 34 d4 d8 b4  c7 39 f3 b5 50 d2 6b b1   K·#K4··· ·9··P·k·
00b0  0a 63 18 63 89 a2 d4 5b  56 1e 74 f1 87 1e bc a2   ·c·c···[ V·t·····
00c0  76 5d 8f 37 5d 29 7a a5  2c c5 f3 7f 0e 7e 8c 86   v]·7])z· ,····~··
```

wireshark_Ethernet 2_20190502212008_a09076.pcapng | Packets: 2150 · Displayed: 2148 (99.9%) | Profile: Default

Fig 14: Packet scanning.


Wireshark · IO Graphs · Ethernet 2
Wireshark IO Graphs: Ethernet 2
Packets/1 sec
1500
1200
900
600
300
0
0
4
8
12
16
20
24
Time (s)

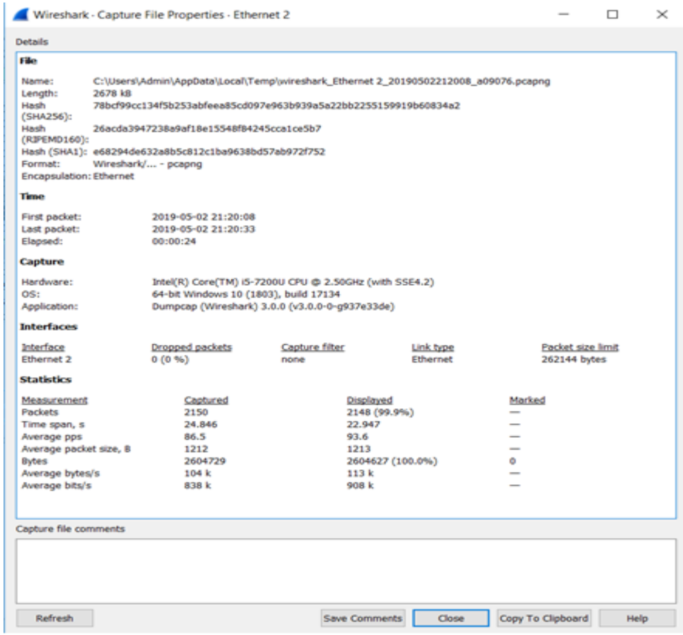
Click to select packet 2148 (22s = 1).

Enabled	Graph Name	Display Filter	Color	Style	Y Axis	Y Field
☑	All packets		■	Line	Packets	
☑	TCP errors	tcp.analysis.flags	■	Bar	Packets	

Mouse ● drags ○ zooms Interval 1 sec □ Time of day □ Log scale

Fig 15: i/o graph.

Under ideal conditions, the graph represents a line develops over time indicating efficient performance the TCP connection [15].



File

Name: C:\Users\Admin\AppData\Local\Temp\wireshark_Ethernet 2_20190502212008_a09076.pcapng
Length: 2678 kB
Hash (SHA256): 78bcf99cc134f5b253abfeea85cd097e963b939a5a22bb2255159919b60834a2
Hash (RIPEMD160): 26acda3947238a9af18e15548f84245cca1ce5b7
Hash (SHA1): e68294de632a8b5c812c1ba9638bd57ab972f752
Format: Wireshark/... - pcapng
Encapsulation: Ethernet

Time

First packet: 2019-05-02 21:20:08
Last packet: 2019-05-02 21:20:33
Elapsed: 00:00:24

Capture

Hardware: Intel(R) Core(TM) i5-7200U CPU @ 2.50GHz (with SSE4.2)
OS: 64-bit Windows 10 (1803), build 17134
Application: Dumpcap (Wireshark) 3.0.0 (v3.0.0-0-g937e33de)

Interfaces

Interface	Dropped packets	Capture filter	Link type	Packet size limit
Ethernet 2	0 (0 %)	none	Ethernet	262144 bytes

Statistics

Measurement	Captured	Displayed	Marked
Packets	2150	2148 (99.9%)	—
Time span, s	24.846	22.947	—
Average pps	86.5	93.6	—
Average packet size, B	1212	1213	—
Bytes	2604729	2604627 (100.0%)	0
Average bytes/s	104 k	113 k	—
Average bits/s	838 k	908 k	—

Capture file comments

Fig 16: capture file properties.

VI. COMPARISON TABLE OF THE NETWORK SECURITY TOOLS

NO.	TOOL	FUNCTION	USES
1.	Zed Attack Proxy	Web application security scanner	Scans vulnerabilities in a web application, and gives solutions to those problems.
2.	Aircrack-ng	Wi-fi network security checker	used to test the weaknesses of wireless networks by accessing the network.
3.	NMAP	Network mapper	Gives all the information about a network about hosts, operating systems etc.
4.	Wireshark	Packet analyzer	used for network analysis, software and communications protocol development, troubleshooting and education.

VII. CONCLUSION

Here we have exhibited the subject of Network Security and some of the Network Security Tools that will help us in finding the vulnerabilities in a website. In this paper we see the usage, main features, principles and the working scenarios of network security tools like zap, aircrack-ng, Nmap and Wireshark. Thus, all these tools help us in securing the network and preventing it to be vulnerable from any third-party system.

VII. REFERENCES

[1] Importance of Cryptography in network security by T. Rajani Devi

[2] A survey on network security tools for open source by Nabanita Mandal ; Sonali Jadhav

[3] Network security management tool for distribution systems by Zbigniew A. Styczynski ; Chris O. Heyde ; Bernd M. Buchholz ; Olaf Ruhle

[4] Evaluation of vulnerability scanners by Yuma Makino,University of Aizu, Tsuruga, Ikki-Machi, Aizu-Wakamatsu, Fukushima, Japan, 965-8580, Vitaly Klyuev, University of Aizu, Tsuruga, Ikki-Machi, Aizu-Wakamatsu, Fukushima, Japan, 965-8580

[5] Security testing as a service with docker containerization by P. P. W. Pathirathna, V. A. I. Ayesha, W. A. T. Imihira, W. M. J. C. Wasala, Nuwan Kodagoda, E. A. T. D. Edirisinghe, Computing Department, Sri Lanka Institute of Information Technology, Malabe, Sri Lanka

[6] Wi-Fi Protected Access (WPA) –PSK (Phase Shift Keying) Key Cracking Using AIRCRACK-NG by Sheikh Md. Rabiul Islam

[7] Shuaib, K., Boulmalf M., Sallabi F. and Lakas A.," Performance Analysis: Co-existence of IEEE 802.11g with Bluetooth", Second IFIP International Conference on Wireless and Optical communication Networks, WOCN 2005, sponsored by IEEE. Dubai, March 6-9, 2005

[8] Comparison of Wireless Network Penetration Testing Tools on Desktops and Raspberry Pi Platforms Aparicio Carranza, PhD1 , Daniel Mayorga, BTech1 , Casimer DeCusatis, PhD2 and Hossein Rahemi, PhD3 1 New York City College of Technology - CUNY, Brooklyn, NY USA, acarranza@citytech.cuny.edu 2 Marist College, Poughkeepsie, NY USA, casimer.decusatis@marist.edu 3 Vaughn College of Aeronautics & Technology, East Elmhurst, NY USA, hossein.rahemi@vaughn.edu

[9] Z. Trabelsi, K. Hayawi, A. Braiki, and S. Mathew, Network Attacks and Defenses: A Hands-on Approach, Boca Raton, Florida: CRC Press, 2013.

[10] Ms. Gurline Kaur and Navjot Kaur, "Penetration Testing – Reconnaissance with NMAP Tool", International Journal of Advanced Research in Computer Science, Volume 8, No. 3, March – April 2017.

[11] Sandeep Kumar Yadav* Daya Shankar Pandey Shrikant Lade, "A Network Based Approach to Discover Security Vulnerability on Host System", International Journal of Advanced Research in Computer Science and Software Engineering, Volume 4, Issue 12, December 2014.

[12] Sheetal Bairwa, Bhawna Mewara and Jyoti Gajrani, "Vulnerability Scanners: A Proactive Approach to Assess Web Application Security", International Journal on Computational Sciences & Applications (IJCSA) Vol.4, No.1, February 2014.

[13] Jhilam Biswas and Ashutosh, "An Insight in to Network Traffic Analysis using Packet Sniffer", International Journal of Computer Applications (0975 – 8887) Volume 94 – No 11, May 2014.

[14] Vivens Ndatinya, Zhifeng Xiao, Vasudeva Rao Manepalli, Ke Meng and Yang Xiao, "Network forensics analysis using Wireshark", Int. J. Sensor Networks, Vol. 10, No. 2, 2015.

[15] Borja Merino Febrer, Eduardo Carozo and Manuel Belda "Traffic Analysis with Wireshark", The National Communications Technology Institute, February 2011.