Cyber Security Actionable Education during COVID19 Third Wave in India

Shailesh Khant
Faculty Computer Science and Applications
Charusat, Changa,
Anand, India
shaileshkhant.mca@charusat.ac.in

Atul Patel
Faculty Computer Science and Applications
Charusat, Changa,
Anand, India
atulpatel.mca@ charusat.ac.in

Sanskruti Patel
Faculty Computer Science and Applications
Charusat, Changa,
Anand, India
sanskrutipatel.mca@charusat.ac.in

Nilay Ganatra
Faculty Computer Science and Applications
Charusat, Changa,
Anand, India
nilayganatra.mca@ charusat.ac.in

Rachana Patel
Faculty Computer Science and Applications
Charusat, Changa,
Anand, India
rachanapatel.mca@ charusat.ac.in

2022 3rd International Conference on Intelligent Engineering and Management (ICIEM) | 978-1-6654-6756-8/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICIEM54221.2022.9853091

Abstract— **Still in many countries COVID19 virus is changing its structure and creating damages in terms of economy and education. In India during the period of January 2022 third wave is on its high peak. Many colleges and schools are still forced to teach online. This paper describes how cyber security actionable or practical fundamental were taught by school or college teachers. Various cyber security tools are used to explain the actionable insight of the subject. Main Topics or concepts covered are MITM (Man In the Middle Attack) using ethercap tool in Kali Linux, spoofing methods like ARP (Address Resolution Protocol) spoofing and DNS (Domain Name System) spoofing, network intrusion detection using snort , finding information about packets using wireshark tool and other tools like nmap and netcat for finding the vulnerability. Even brief details were given about how to crack password using wireshark.**

Keywords— Cyber Security, ARP spoofing, DNS spoofing, MITM, Wireshark,

I. INTRODUCTION

In India, still third wave is making its impact in terms of economy and education. Education sectors are suffering a lot in terms of conceptual knowledge gain by students while studying from home. An effort was made by several Indian school and college teachers to cope up these things. Almost in all courses whether its science, engineering or medical, cyber security education is must. Now a day's security is the major concerns in almost all kind of networks. Several online tools related to cyber security are used to make these complex task possible. Cyber security education is mandatory in all sectors. All commercial institutes and companies are suffering from cyber-attack. Even many political parties and government websites becomes victims of cyber-attack [1] [2].

To fight and to avoid cyber-attacks, fundamentals or conceptual knowledge of cyber security and its term is required. The most important term is vulnerability which defines the inability of your system to fight against various kinds of cyber-attacks. Cyber-attack can be performed by active or passive intruders. Intruders try to get the important things from your electronics machine and make wrong use of it. Some of the examples where intruders can attack your system are key logging, Social engineering, Data exploitation, malware advertising attack, phishing, introducing spam, ransomware or other such type of attack like inserting Trojan virus. User can get infections from many sources like downloading unknown email attachments, downloading music or video files from unknown sources, getting instant messages or peer to peer data transfer etc. Many of the people are still thinking that only rich people can be targeted or setting a strong password is ok but it's not true. Any person can be a victim of cyber-attack [3] [4].

II. CYBER SECURITY ACTIONABLE

Based on current problems and survey, following applications or tools are included in curriculum of school and undergraduate courses.

1. NMAP for vulnerability
2. Netcat tool and its uses
3. Man in the Middle Attack
4. Wireshark for packet information
5. ARP Spoofing
6. DNS Spoofing
7. SNORT for network intrusion

NMAP is used for TCP port scanning and it is used to perform various tasks like service detection, scanning single IP address, scanning multiple IP addresses, finding active machines and active ports in network, scanning TCP port, scanning UDP port and finding all sent and received packets [5].

Netcat tool is used for creating inbound or outbound connections in a network using either TCP or UDP protocols. Netct is having ability to use any kind of local port as a source address. ”nc” command is used in any network client to use

978-1-6654-6756-8/22/$31.00 ©2022 IEEE

netcat tool. Client on the network are using this utility to transfer files in the network or for sending instantaneous messages. Even it can work as port scanner.

Man in the Middle (MITM) attack is a kind of attack where any third person can attack on other two persons in the same network who are communicating two each other. For example assume that a person is receiving email from server to enter login credentials on that server but a person in between can create similar website to send an email to a person and get access of login credentials. MITM attack can easily possible in Wi-Fi or LAN network with poor security. Some of the malicious things included in MITM attack are IP spoofing, DNS spoofing, Https spoofing, email hijacking etc. MITB (Man in the browser) is an attack which is similar to MITM [6] [7].

Wireshark is a popular open source tool which can be used to analyze network protocols. It can be freely downloaded. It can be used by many government and non-government organizations as a part of security analysis. It can have many features like analyzing protocols, Live and offline packet capturing, VoIP analysis, LAN / MAN / WAN analysis etc. [8] [9].

ARP is an Address Resolution Protocol which maps IP address into its appropriate MAC address or vice versa. Originally ARP does not verify whether request to get MAC address is authorized or unauthorized. It simply checks whether request is from same network or not. This is a weak point of ARP and due to this unauthorized persons can use it as spoofing attacks. Spoofing tools such as Arpspoof and Driftnet can be used to send illegal requests. It is a kind of Man in the Middle attack where person can get unauthorized access. If an ARP spoofing is successful then an attacker can sniff packets from the network and steal important data. Attacker can even get access of victim's accounts. DoS (Denial of Service) and Distributed DoS is possible where attacker can attack on large number of machines [10][11].

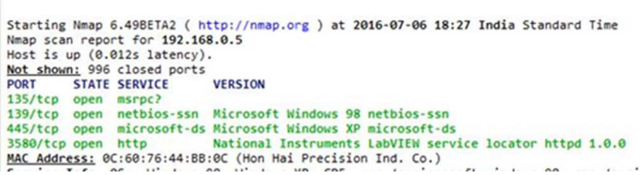
DNS is Domain Name System, where user can get access of website based on domain name instead of IP address. In DNS spoofing attacker can create a webpage which looks like a page of original website. For example a similar bank webpage is created where account holder can enter credentials details and handover this information to attacker. Tampering of existing DNS server can be done in DNS Spoofing. Similar to ARP spoofing, it is also a kind of MITM where any unauthorized person in the same network can stay in the middle and attack [12].

Snort is an IPS (Intrusion Prevention System). It is an open source tool which can alert users for malicious packet transmission or reception. It can analyze real time traffic and log information of real time packet. It needs to define series of rules. Based on these rules an alert can be generated. Main usage of snorts are packet logging, packet sniffing and tcpdump. Packet or data captured using snort will be stored in a database like MySQL and it can be analyzed further [13].

III. Simulation Results

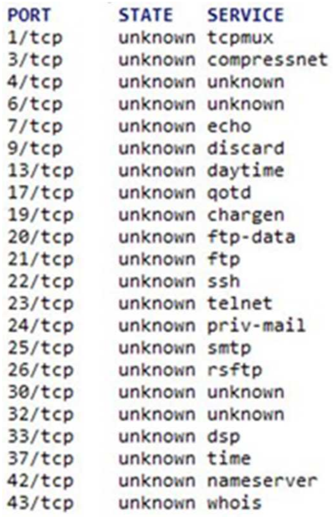
Cyber security experiments were performed for following applications.

A. NMAP Simulation: Nmap is an open source tool which can be freely downloaded. It is having basic commands for basic scanning, advanced scanning, port scanning and network discovery. Using basic commands user can get information related to nmap services as shown in Fig 1.



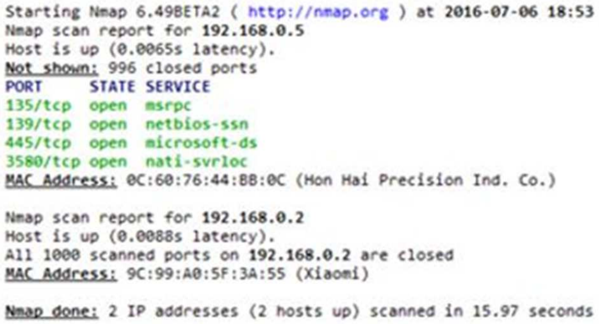
```
Starting Nmap 6.49BETA2 ( http://nmap.org ) at 2016-07-06 18:27 India Standard Time
Nmap scan report for 192.168.0.5
Host is up (0.012s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE      VERSION
135/tcp  open  msrpc?
139/tcp  open  netbios-ssn  Microsoft Windows 98 netbios-ssn
445/tcp  open  microsoft-ds Microsoft Windows XP microsoft-ds
3580/tcp open  http         National Instruments LabVIEW service locator httpd 1.0.0
MAC Address: 0C:60:76:44:BB:0C (Hon Hai Precision Ind. Co.)
```

Figure 1: NMAP Services



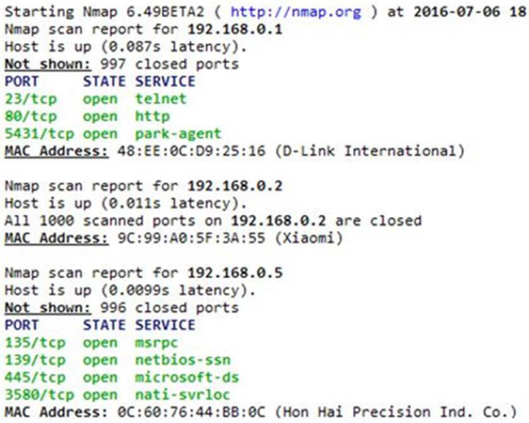
```
PORT     STATE   SERVICE
1/tcp    unknown tcpmux
3/tcp    unknown compressnet
4/tcp    unknown unknown
6/tcp    unknown unknown
7/tcp    unknown echo
9/tcp    unknown discard
13/tcp   unknown daytime
17/tcp   unknown qotd
19/tcp   unknown chargen
20/tcp   unknown ftp-data
21/tcp   unknown ftp
22/tcp   unknown ssh
23/tcp   unknown telnet
24/tcp   unknown priv-mail
25/tcp   unknown smtp
26/tcp   unknown rsftp
30/tcp   unknown unknown
32/tcp   unknown unknown
33/tcp   unknown dsp
37/tcp   unknown time
42/tcp   unknown nameserver
43/tcp   unknown whois
```

Figure 2: NMAP All Port Information



```
Starting Nmap 6.49BETA2 ( http://nmap.org ) at 2016-07-06 18:53
Nmap scan report for 192.168.0.5
Host is up (0.0065s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3580/tcp open  nati-svrloc
MAC Address: 0C:60:76:44:BB:0C (Hon Hai Precision Ind. Co.)

Nmap scan report for 192.168.0.2
Host is up (0.0088s latency).
All 1000 scanned ports on 192.168.0.2 are closed
MAC Address: 9C:99:A0:5F:3A:55 (Xiaomi)

Nmap done: 2 IP addresses (2 hosts up) scanned in 15.97 seconds
```

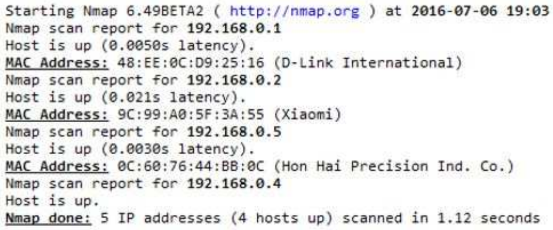
Figure 3: NMAP Single IP Address Scanning



```
Starting Nmap 6.49BETA2 ( http://nmap.org ) at 2016-07-06 18:57
Nmap scan report for 192.168.0.1
Host is up (0.087s latency).
Not shown: 997 closed ports
PORT     STATE SERVICE
23/tcp   open  telnet
80/tcp   open  http
5431/tcp open  park-agent
MAC Address: 48:EE:0C:D9:25:16 (D-Link International)

Nmap scan report for 192.168.0.2
Host is up (0.011s latency).
All 1000 scanned ports on 192.168.0.2 are closed
MAC Address: 9C:99:A0:5F:3A:55 (Xiaomi)

Nmap scan report for 192.168.0.5
Host is up (0.0099s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3580/tcp open  nati-svrloc
MAC Address: 0C:60:76:44:BB:0C (Hon Hai Precision Ind. Co.)
```

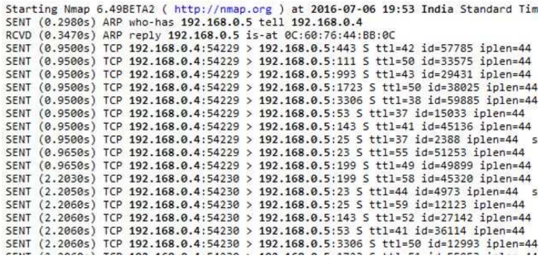
Figure 4: NMAP Multiple IP Address Scanning

In Fig 2, Fig 3 and Fig 4 usage of nmap to get port information, single IP address scanning and multiple IP address scanning are shown.



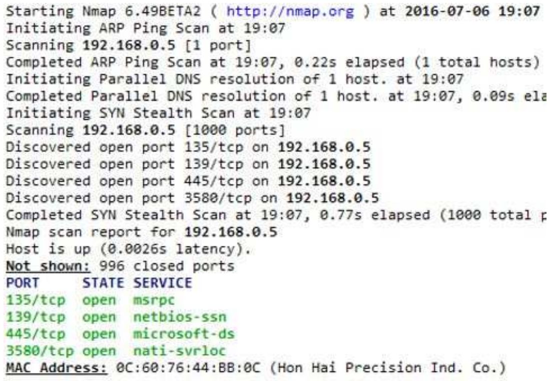
```
Starting Nmap 6.49BETA2 ( http://nmap.org ) at 2016-07-06 19:03
Nmap scan report for 192.168.0.1
Host is up (0.0050s latency).
MAC Address: 48:EE:0C:D9:25:16 (D-Link International)
Nmap scan report for 192.168.0.2
Host is up (0.021s latency).
MAC Address: 9C:99:A0:5F:3A:55 (Xiaomi)
Nmap scan report for 192.168.0.5
Host is up (0.0030s latency).
MAC Address: 0C:60:76:44:BB:0C (Hon Hai Precision Ind. Co.)
Nmap scan report for 192.168.0.4
Host is up.
Nmap done: 5 IP addresses (4 hosts up) scanned in 1.12 seconds
```

Figure 5: NMAP other address scanning



```
Starting Nmap 6.49BETA2 ( http://nmap.org ) at 2016-07-06 19:53 India Standard Tim
SENT (0.2980s) ARP who-has 192.168.0.5 tell 192.168.0.4
RCVD (0.3470s) ARP reply 192.168.0.5 is-at 0C:60:76:44:BB:0C
SENT (0.9500s) TCP 192.168.0.4:54229 > 192.168.0.5:443 S ttl=42 id=57785 iplen=44
SENT (0.9500s) TCP 192.168.0.4:54229 > 192.168.0.5:111 S ttl=50 id=33575 iplen=44
SENT (0.9500s) TCP 192.168.0.4:54229 > 192.168.0.5:993 S ttl=43 id=29431 iplen=44
SENT (0.9500s) TCP 192.168.0.4:54229 > 192.168.0.5:1723 S ttl=50 id=38025 iplen=44
SENT (0.9500s) TCP 192.168.0.4:54229 > 192.168.0.5:3306 S ttl=38 id=59885 iplen=44
SENT (0.9500s) TCP 192.168.0.4:54229 > 192.168.0.5:53 S ttl=37 id=15033 iplen=44
SENT (0.9500s) TCP 192.168.0.4:54229 > 192.168.0.5:143 S ttl=41 id=45136 iplen=44
SENT (0.9500s) TCP 192.168.0.4:54229 > 192.168.0.5:25 S ttl=37 id=2388 iplen=44  s
SENT (0.9650s) TCP 192.168.0.4:54229 > 192.168.0.5:23 S ttl=55 id=51253 iplen=44
SENT (0.9650s) TCP 192.168.0.4:54229 > 192.168.0.5:199 S ttl=49 id=49899 iplen=44
SENT (2.2030s) TCP 192.168.0.4:54230 > 192.168.0.5:199 S ttl=58 id=45320 iplen=44
SENT (2.2050s) TCP 192.168.0.4:54230 > 192.168.0.5:23 S ttl=44 id=4973 iplen=44  s
SENT (2.2060s) TCP 192.168.0.4:54230 > 192.168.0.5:25 S ttl=59 id=12123 iplen=44
SENT (2.2060s) TCP 192.168.0.4:54230 > 192.168.0.5:143 S ttl=52 id=27142 iplen=44
SENT (2.2060s) TCP 192.168.0.4:54230 > 192.168.0.5:53 S ttl=41 id=36114 iplen=44
SENT (2.2060s) TCP 192.168.0.4:54230 > 192.168.0.5:3306 S ttl=50 id=12993 iplen=44
```

Figure 6: NMAP Packets sent

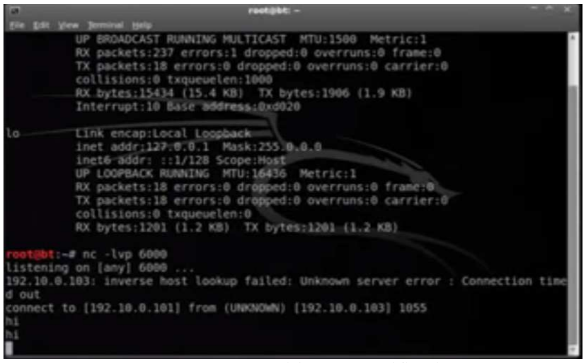


```
Starting Nmap 6.49BETA2 ( http://nmap.org ) at 2016-07-06 19:07
Initiating ARP Ping Scan at 19:07
Scanning 192.168.0.5 [1 port]
Completed ARP Ping Scan at 19:07, 0.22s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 19:07
Completed Parallel DNS resolution of 1 host. at 19:07, 0.09s ela
Initiating SYN Stealth Scan at 19:07
Scanning 192.168.0.5 [1000 ports]
Discovered open port 135/tcp on 192.168.0.5
Discovered open port 139/tcp on 192.168.0.5
Discovered open port 445/tcp on 192.168.0.5
Discovered open port 3580/tcp on 192.168.0.5
Completed SYN Stealth Scan at 19:07, 0.77s elapsed (1000 total p
Nmap scan report for 192.168.0.5
Host is up (0.0026s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
3580/tcp open  nati-svrloc
MAC Address: 0C:60:76:44:BB:0C (Hon Hai Precision Ind. Co.)
```

Figure 7: NMAP Open Ports

How addresses other than IP addresses are scanned is shown in Fig 5. The transmission and reception of packets is shown in Fig 6. The information related to open ports can also be found using nmap and it is represented in Fig 7.

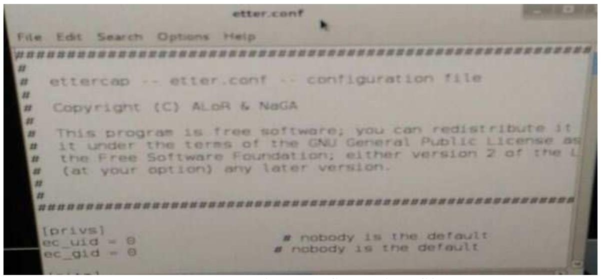
B. Netcat tool and its uses: To check whether the netcat tool is connected to the network just type the following command in the Linux operating system. Open Linux directly or use telnet and type “nc -lvp (port address, port number)”.After connection of netcat tool with the operating system a message will pop-up in the linux terminal as shown in Fig 8.



```
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:237 errors:1 dropped:0 overruns:0 frame:0
          TX packets:18 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:15434 (15.4 KB)  TX bytes:1906 (1.9 KB)
          Interrupt:10 Base address:0xd020

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:16436  Metric:1
          RX packets:18 errors:0 dropped:0 overruns:0 frame:0
          TX packets:18 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1201 (1.2 KB)  TX bytes:1201 (1.2 KB)

root@bt:~# nc -lvp 6000
listening on [any] 6000 ...
192.10.0.103: inverse host lookup failed: Unknown server error : Connection time
d out
connect to [192.10.0.101] from (UNKNOWN) [192.10.0.103] 1055
hi
hi
```

Figure 8: netcat with port specification

C. Man in the Middle Attack



```
##################################################################
#
#  ettercap -- etter.conf -- configuration file
#
#  Copyright (C) ALoR & NaGA
#
#  This program is free software; you can redistribute it
#  it under the terms of the GNU General Public License as
#  the Free Software Foundation; either version 2 of the L
#  (at your option) any later version.
#
#
##################################################################

[privs]
ec_uid = 0                # nobody is the default
ec_gid = 0                # nobody is the default
```

Figure 9: Ether configuration file

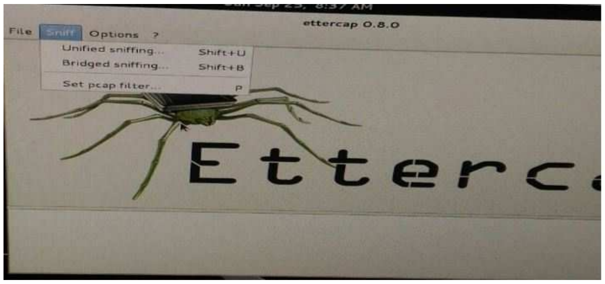
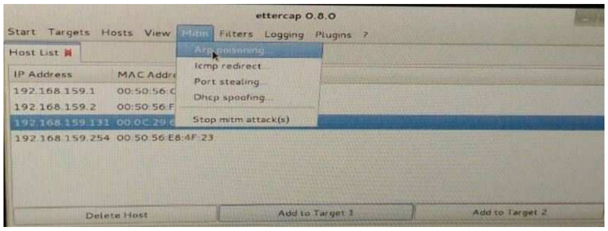


Figure 10: Ether cap unified sniffing

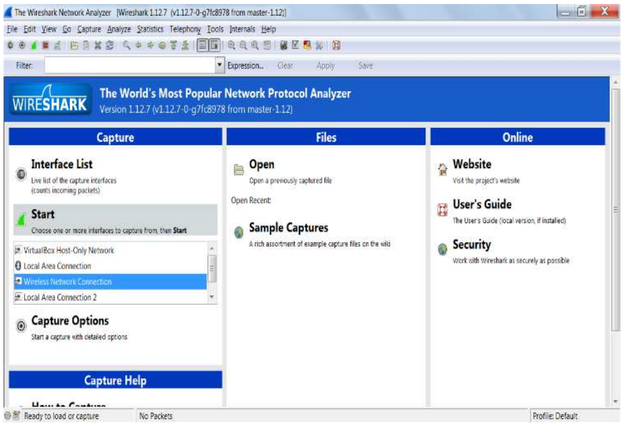


IP Address	MAC Addre
192.168.159.1	00:50:56:C
192.168.159.2	00:50:56:F
192.168.159.131	00:0C:29:(
192.168.159.254	00:50:56:E8:4F:23

Figure 11: Ether cap ARP Poisoning

To perform MITM attack free virtual box software is used where user can install 3 different operating system for testing purpose. In our case two Operating systems communicating each other are windows 7 and windows XP. Kali Linux operating system is used as an intruder which can perform MITM attack. User needs to configure ether configuration file as shown in Fig 9. How to apply ether cap unified niffing and ARP poisoning is shown in Fig 10 and Fig 11.

D. Wireshark for packet information


The Wireshark Network Analyzer [Wireshark 1.12.7 (v1.12.7-0-g7fc8978 from master-1.12)]

The World's Most Popular Network Protocol Analyzer
Version 1.12.7 (v1.12.7-0-g7fc8978 from master-1.12)

Capture

Interface List
Live list of the capture interfaces (counts incoming packets)

Start
Choose one or more interfaces to capture from, then Start

VirtualBox Host-Only Network
Local Area Connection
Wireless Network Connection
Local Area Connection 2

Capture Options
Start a capture with detailed options

Capture Help

Files

Open
Open a previously captured file

Open Recent:

Sample Captures
A rich assortment of example capture files on the wiki

Online

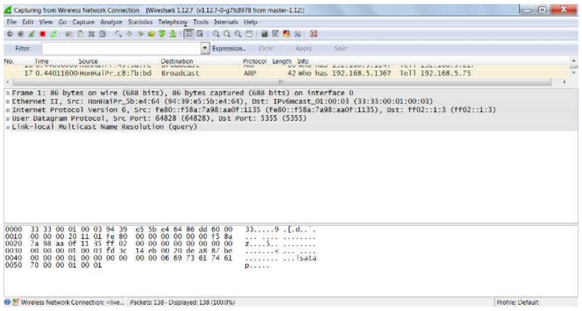
Website
Visit the project's website

User's Guide
The User's Guide (local version, if installed)

Security
Work with Wireshark as securely as possible

Ready to load or capture | No Packets | Profile: Default
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression...
Clear
Apply
Save
WIRESHARK

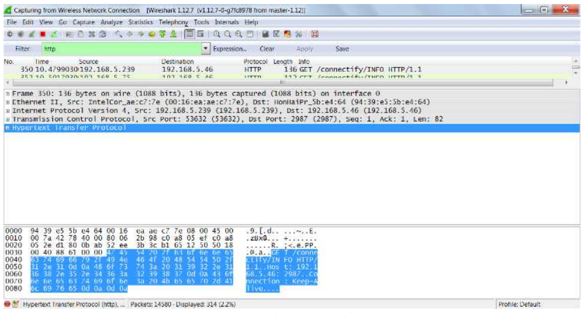
Figure 12: Wireshark Main screen


Capturing from Wireless Network Connection [Wireshark 1.12.7 (v1.12.7-0-g7fc8978 from master-1.12)]

```
No.  Time        Source              Destination  Protocol Length Info
  17 0.44011600 HonHaiPr_c8:7b:bd   Broadcast    ARP          42 Who has 192.168.5.136? Tell 192.168.5.75

⊞ Frame 1: 86 bytes on wire (688 bits), 86 bytes captured (688 bits) on interface 0
⊞ Ethernet II, Src: HonHaiPr_5b:e4:64 (94:39:e5:5b:e4:64), Dst: IPv6mcast_01:00:03 (33:33:00:01:00:03)
⊞ Internet Protocol Version 6, Src: fe80::f58a:7a98:aa0f:1135 (fe80::f58a:7a98:aa0f:1135), Dst: ff02::1:3 (ff02::1:3)
⊞ User Datagram Protocol, Src Port: 64828 (64828), Dst Port: 5355 (5355)
⊞ Link-local Multicast Name Resolution (query)

0000  33 33 00 01 00 03 94 39  e5 5b e4 64 86 dd 60 00   33.....9 .[.d..`.
0010  00 00 00 20 11 01 fe 80  00 00 00 00 00 00 f5 8a   ... .... ........
0020  7a 98 aa 0f 11 35 ff 02  00 00 00 00 00 00 00 00   z....5.. ........
0030  00 00 00 01 00 03 fd 3c  14 eb 00 20 de a8 87 be   .......< ... ....
0040  00 00 00 01 00 00 00 00  00 00 06 69 73 61 74 61   ........ ...isata
0050  70 00 00 01 00 01                                  p.....
```

Wireless Network Connection: <live... | Packets: 138 · Displayed: 138 (100.0%) | Profile: Default
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression...
Clear
Apply
Save

Figure 13: Wireshark Port information


Capturing from Wireless Network Connection [Wireshark 1.12.7 (v1.12.7-0-g7fc8978 from master-1.12)]

Filter: http

```
No.  Time        Source         Destination   Protocol Length Info
 350 10.4799030 192.168.5.239  192.168.5.46  HTTP        136 GET /connectify/INFO HTTP/1.1

⊞ Frame 350: 136 bytes on wire (1088 bits), 136 bytes captured (1088 bits) on interface 0
⊞ Ethernet II, Src: IntelCor_ae:c7:7e (00:16:ea:ae:c7:7e), Dst: HonHaiPr_5b:e4:64 (94:39:e5:5b:e4:64)
⊞ Internet Protocol Version 4, Src: 192.168.5.239 (192.168.5.239), Dst: 192.168.5.46 (192.168.5.46)
⊞ Transmission Control Protocol, Src Port: 53632 (53632), Dst Port: 2987 (2987), Seq: 1, Ack: 1, Len: 82
⊞ Hypertext Transfer Protocol

0000  94 39 e5 5b e4 64 00 16  ea ae c7 7e 08 00 45 00   .9.[.d.. ...~..E.
0010  00 7a 42 78 40 00 80 06  2b 98 c0 a8 05 ef c0 a8   .zBx@... +.......
0020  05 2e d1 80 0b ab 52 ee  3b 3c b1 65 12 50 50 18   ......R. ;<.e.PP.
0030  00 40 88 61 00 00 47 45  54 20 2f 63 6f 6e 6e 65   .@.a..GE T /conne
0040  63 74 69 66 79 2f 49 4e  46 4f 20 48 54 54 50 2f   ctify/IN FO HTTP/
0050  31 2e 31 0d 0a 48 6f 73  74 3a 20 31 39 32 2e 31   1.1..Hos t: 192.1
0060  36 38 2e 35 2e 34 36 3a  32 39 38 37 0d 0a 43 6f   68.5.46: 2987..Co
0070  6e 6e 65 63 74 69 6f 6e  3a 20 4b 65 65 70 2d 41   nnection : Keep-A
0080  6c 69 76 65 0d 0a 0d 0a                           live....
```

Hypertext Transfer Protocol (http), ... | Packets: 14580 · Displayed: 314 (2.2%) | Profile: Default
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression...
Clear
Apply
Save

Figure 14: Wireshark Packet capturing

Wireshark can be freely downloaded for both windows and Linux based operating system. Main screen is shown in Fig 12. From Capture menu select interface and get started. Once a start button is clicked, a main screen is opened as shown in Fig 13. Wireshark will now capture live packet tracing and network activities. From filter menu user can select specific protocol like http, TCP or UDP. One of the packet filtering for http is shown in Fig 14.

E. ARP Spoofing

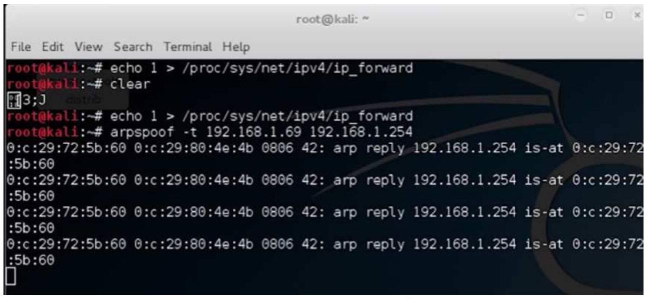

```
root@kali:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@kali:~# clear
3;J
root@kali:~# echo 1 > /proc/sys/net/ipv4/ip_forward
root@kali:~# arpspoof -t 192.168.1.69 192.168.1.254
0:c:29:72:5b:60 0:c:29:80:4e:4b 0806 42: arp reply 192.168.1.254 is-at 0:c:29:72
:5b:60
0:c:29:72:5b:60 0:c:29:80:4e:4b 0806 42: arp reply 192.168.1.254 is-at 0:c:29:72
:5b:60
0:c:29:72:5b:60 0:c:29:80:4e:4b 0806 42: arp reply 192.168.1.254 is-at 0:c:29:72
:5b:60
0:c:29:72:5b:60 0:c:29:80:4e:4b 0806 42: arp reply 192.168.1.254 is-at 0:c:29:72
:5b:60
```

Figure 15: ARP Spoofing

ARP spoofing is performed by finding the IP addresses of all three operating system used in MITM attack. The command used to apply ARP spoofing is “arpspoof”. The syntax of that command is “arpspoof -i eth0 -t (target ipv4 address of any os1, target address of os2)". One of the example is shown in Fig 15.

F. DNS Spoofing

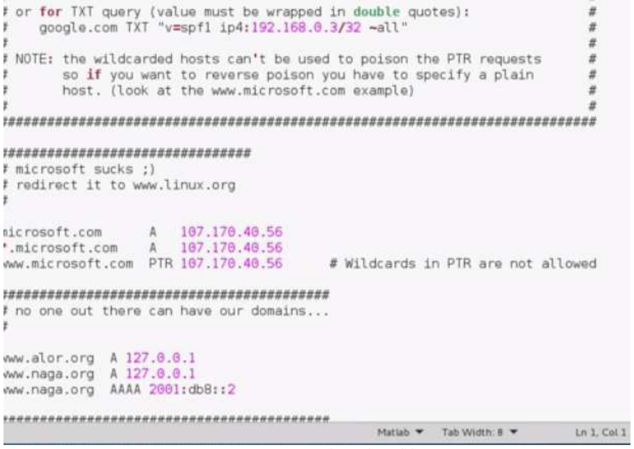
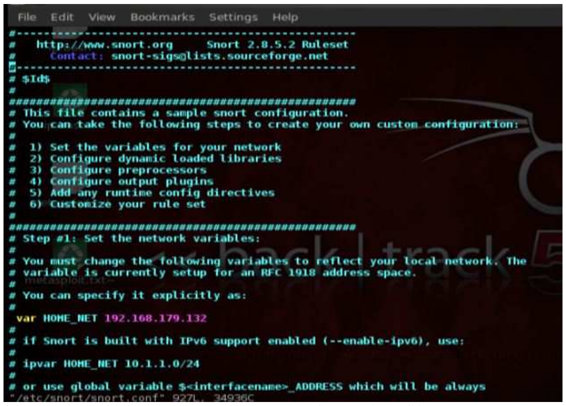

```
# or for TXT query (value must be wrapped in double quotes):              #
#    google.com TXT "v=spf1 ip4:192.168.0.3/32 ~all"                      #
#                                                                         #
# NOTE: the wildcarded hosts can't be used to poison the PTR requests     #
#       so if you want to reverse poison you have to specify a plain      #
#       host. (look at the www.microsoft.com example)                     #
#                                                                         #
###########################################################################

################################
# microsoft sucks ;)
# redirect it to www.linux.org
#

microsoft.com      A   107.170.40.56
*.microsoft.com    A   107.170.40.56
www.microsoft.com  PTR 107.170.40.56      # Wildcards in PTR are not allowed

##########################################
# no one out there can have our domains...
#

www.alor.org  A 127.0.0.1
www.naga.org  A 127.0.0.1
www.naga.org  AAAA 2001:db8::2

##########################################
```

Figure 16: Ether.dns file

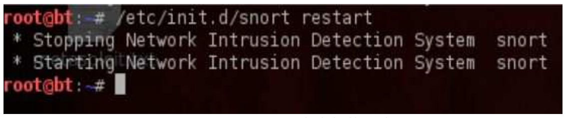
To perform DNS spoofing find IP address of Operating system and gateway address. Perform ARP poisoning or MITM attack and select dns_spoof plugin. Open etter.dns file and make changes according to redirection of webpage as shown in Fig 16. Once these changes are done, open specified address which is redirected to webpage specified by user.

G. SNORT Intrusion Detection: User needs to install windows based or Linux based snort for performing intrusion detection. In Linux based system first install library for snort and start editing snort.conf file as shown in Fig 17. Restart the snort as shown in Fig 18.To apply snort use “Snort –q –A console -i eth0 –c /etc./snort/snort.conf “command as shown in Fig 19. A log is generated for all activities as shown in Fig 20.



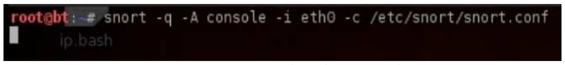
```
#--------------------------------------------------
#   http://www.snort.org     Snort 2.8.5.2 Ruleset
#     Contact: snort-sigs@lists.sourceforge.net
#--------------------------------------------------
# $Id$
#
###################################################
# This file contains a sample snort configuration.
# You can take the following steps to create your own custom configuration:
#
#  1) Set the variables for your network
#  2) Configure dynamic loaded libraries
#  3) Configure preprocessors
#  4) Configure output plugins
#  5) Add any runtime config directives
#  6) Customize your rule set
#
###################################################
# Step #1: Set the network variables:
#
# You must change the following variables to reflect your local network. The
# variable is currently setup for an RFC 1918 address space.
#
# You can specify it explicitly as:
#
 var HOME_NET 192.168.179.132
#
# if Snort is built with IPv6 support enabled (--enable-ipv6), use:
#
# ipvar HOME_NET 10.1.1.0/24
#
# or use global variable $<interfacename>_ADDRESS which will be always
"/etc/snort/snort.conf" 927L, 34936C
```

Figure 17: snort configuration file



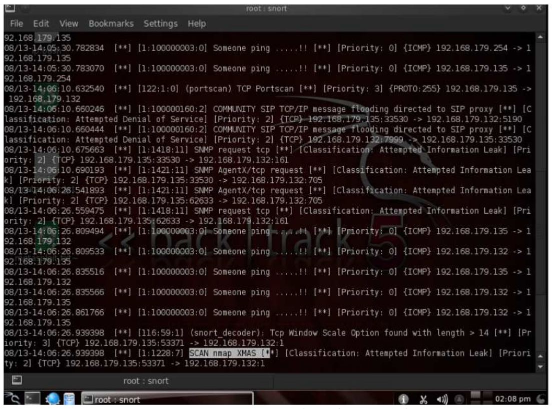
```
root@bt:~# /etc/init.d/snort restart
 * Stopping Network Intrusion Detection System  snort
 * Starting Network Intrusion Detection System  snort
root@bt:~# 
```

Figure 18: snort initialization



```
root@bt:~# snort -q -A console -i eth0 -c /etc/snort/snort.conf
```

Figure 19: snort implementation



```
92.168.179.135
08/13-14:05:30.782834  [**] [1:100000003:0] Someone ping .....!! [**] [Priority: 0] {ICMP} 192.168.179.254 -> 1
92.168.179.135
08/13-14:05:30.783070  [**] [1:100000003:0] Someone ping .....!! [**] [Priority: 0] {ICMP} 192.168.179.135 -> 1
92.168.179.254
08/13-14:06:10.632540  [**] [122:1:0] (portscan) TCP Portscan [**] [Priority: 3] {PROTO:255} 192.168.179.135 ->
 192.168.179.132
08/13-14:06:10.660246  [**] [1:100000160:2] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy [**] [C
lassification: Attempted Denial of Service] [Priority: 2] {TCP} 192.168.179.135:33530 -> 192.168.179.132:5190
08/13-14:06:10.660444  [**] [1:100000160:2] COMMUNITY SIP TCP/IP message flooding directed to SIP proxy [**] [C
lassification: Attempted Denial of Service] [Priority: 2] {TCP} 192.168.179.132:7999 -> 192.168.179.135:33530
08/13-14:06:10.675663  [**] [1:1418:11] SNMP request tcp [**] [Classification: Attempted Information Leak] [Pri
ority: 2] {TCP} 192.168.179.135:33530 -> 192.168.179.132:161
08/13-14:06:10.690193  [**] [1:1421:11] SNMP AgentX/tcp request [**] [Classification: Attempted Information Lea
k] [Priority: 2] {TCP} 192.168.179.135:33530 -> 192.168.179.132:705
08/13-14:06:26.541893  [**] [1:1421:11] SNMP AgentX/tcp request [**] [Classification: Attempted Information Lea
k] [Priority: 2] {TCP} 192.168.179.135:62633 -> 192.168.179.132:705
08/13-14:06:26.559475  [**] [1:1418:11] SNMP request tcp [**] [Classification: Attempted Information Leak] [Pri
ority: 2] {TCP} 192.168.179.135:62633 -> 192.168.179.132:161
08/13-14:06:26.809494  [**] [1:100000003:0] Someone ping .....!! [**] [Priority: 0] {ICMP} 192.168.179.135 -> 1
92.168.179.132
08/13-14:06:26.809533  [**] [1:100000003:0] Someone ping .....!! [**] [Priority: 0] {ICMP} 192.168.179.132 -> 1
92.168.179.135
08/13-14:06:26.835516  [**] [1:100000003:0] Someone ping .....!! [**] [Priority: 0] {ICMP} 192.168.179.135 -> 1
92.168.179.132
08/13-14:06:26.835566  [**] [1:100000003:0] Someone ping .....!! [**] [Priority: 0] {ICMP} 192.168.179.132 -> 1
92.168.179.135
08/13-14:06:26.861766  [**] [1:100000003:0] Someone ping .....!! [**] [Priority: 0] {ICMP} 192.168.179.132 -> 1
92.168.179.135
08/13-14:06:26.939398  [**] [116:59:1] (snort_decoder): Tcp Window Scale Option found with length > 14 [**] [Pr
iority: 3] {TCP} 192.168.179.135:53371 -> 192.168.179.132:1
08/13-14:06:26.939398  [**] [1:1228:7] SCAN nmap XMAS [**] [Classification: Attempted Information Leak] [Priori
ty: 2] {TCP} 192.168.179.135:53371 -> 192.168.179.132:1
```

Figure 20: Snort log for attack

IV. CONCLUSION AND FUTURE SCOPE

This paper suggests an innovative way of teaching cyber security course for school and under graduate students. Some basic and advanced tools like NMAP for vulnerability, netcat tool and its uses, Man in the Middle Attack, Wireshark for packet information, ARP Spoofing, DNS Spoofing and SNORT for network intrusion are covered. The results are obtained with kali Linux and Windows operating system. The goal is to create awareness among the students in terms of basic cyber security terms, attacks and how to handle the attacks. Students can download the freely available or open source tools to perform the actionable or practical.

V. ACKNOWLEDGEMENT

The authors would like to thank Principal of Smt. Chandaben Mohanbhai Patel Institute of Computer Applications (CMPICA), Faculty of Computer Science and Applications and management Charotar University of Science and Technology for their constant motivation and support. Without their constant motivation and support the work does not become so worthy.

REFERENCES

[1] N. Ahmad, U. A. Mokhtar, W. Fariza Paizi Fauzi, Z. A. Othman, Y. Hakim Yeop and S. N. Huda Sheikh Abdullah, "Cyber Security Situational Awareness among Parents," 2018 Cyber Resilience Conference (CRC), 2018, pp. 1-3, doi: 10.1109/CR.2018.8626830.

[2] A. Lodgher, J. Yang and U. Bulut, "An Innovative Modular Approach of Teaching Cyber Security across Computing Curricula," 2018 IEEE Frontiers in Education Conference (FIE), 2018, pp. 1-5, doi: 10.1109/FIE.2018.8659040.

[3] S. R. Kumar, S. A. Yadav, S. Sharma and A. Singh, "Recommendations for effective cyber security execution," 2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH), 2016, pp. 342-346, doi: 10.1109/ICICCS.2016.7542327.

[4] S. Khant and A. Patel, "COVID19 Remote Engineering Education: Learning of an Embedded System with Practical Perspective," 2021 International Conference on Innovative Practices in Technology and Management (ICIPTM), 2021, pp. 15-19, doi: 10.1109/ICIPTM52218.2021.9388360.

[5] M. Shah, S. Ahmed, K. Saeed, M. Junaid, H. Khan and Ata-ur-rehman, "Penetration Testing Active Reconnaissance Phase – Optimized Port Scanning With Nmap Tool," 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET), 2019, pp. 1-6, doi: 10.1109/ICOMET.2019.8673520.

[6] S. Umamaheshwari and J. N. Swaminathan, "Man-In-Middle Attack/for a Free Scale Topology," 2018 International Conference on Computer Communication and Informatics (ICCCI), 2018, pp. 1-4, doi: 10.1109/ICCCI.2018.8441202.

[7] L. Zhang, W. Jia, S. Wen and D. Yao, "A Man-in-the-Middle Attack on 3G-WLAN Interworking," 2010 International Conference on Communications and Mobile Computing, 2010, pp. 121-125, doi: 10.1109/CMC.2010.34.

[8] R. Das and G. Tuna, "Packet tracing and analysis of network cameras with Wireshark," 2017 5th International Symposium on Digital Forensic and Security (ISDFS), 2017, pp. 1-6, doi: 10.1109/ISDFS.2017.7916510.

[9] Shaoqiang Wang, DongSheng Xu and ShiLiang Yan, "Analysis and application of Wireshark in TCP/IP protocol teaching," 2010 International Conference on E-Health Networking Digital Ecosystems and Technologies (EDT), 2010, pp. 269-272, doi: 10.1109/EDT.2010.5496372.

[10] S. G. Bhirud and V. Katkar, "Light weight approach for IP-ARP spoofing detection and prevention," 2011 Second Asian Himalayas International Conference on Internet (AH-ICI), 2011, pp. 1-5, doi: 10.1109/AHICI.2011.6113951.

[11] S. Puangpronpitag and N. Masusai, "An efficient and feasible solution to ARP Spoof problem," 2009 6th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology, 2009, pp. 910-913, doi: 10.1109/ECTICON.2009.5137193.

[12] A. A. Maksutov, I. A. Cherepanov and M. S. Alekseev, "Detection and prevention of DNS spoofing attacks," 2017 Siberian Symposium on Data Science and Engineering (SSDSE), 2017, pp. 84-87, doi: 10.1109/SSDSE.2017.8071970.

[13] R. Gaddam and M. Nandhini, "An analysis of various snort based techniques to detect and prevent intrusions in networks proposal with code refactoring snort tool in Kali Linux environment," 2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), 2017, pp. 10-15, doi: 10.1109/ICICCT.2017.7975177.