**`LAB04-1 Scapy sniffing**

**Class student ID Name**

Simulation scenario

|  | Host | Target |
| --- | --- | --- |
| VM type | Virtual box, VMware |  |
| OS | Ubuntu, Centos, kali |  |
| IP | Test-bed IP |  |
| Attacking type | sniffing | |
| Attacking program | Python Scapy | |
| Attacking Process (13steps) | * Install scapy * Open scapy shell by typing “scapy” in terminal. * Sniff the packets use the sniff() function. * Summary(). * Restrict the number of packets to be captured sniff() allows a count parameter. * Filter packets while sniffing using the filter parameter * Filter any packet of source/destination IP address, * Filter any packet port number, protocol by using the BPF syntax. * Explicitly mention the interfaces that we would like to sniff on using the iface parameter. * Pass a function that executes with each packet sniffed * Store the sniffed packets in a pcap file. * Opening GfG.pcap using Wireshark * Sniff packets offline from pcap files | |

1. Install scapy => sudo apt-get install python3-scapy
2. open scapy shell by typing “**scapy**” in terminal. => **scapy**
3. sniff the packets use the **sniff()** function. => capture = sniff()
4. **summary(). =>**capture.summary()
5. restrict the number of packets to be captured sniff() allows a **count** parameter.

capture = sniff(count=5)

1. filter packets while sniffing using the **filter** parameter => sniff(filter="tcp", count=5)
2. filter any packet of source/destination IP address,
3. filter any packet port number, protocol by using the **BPF** syntax.
4. explicitly mention the interfaces that we would like to sniff on using the **iface** parameter.

sniff(iface="eth0", count=5)

1. pass a function that executes with each packet sniffed. This allows us to do some custom actions with each packet sniffed.

sniff(prn=lambda x:x.summary(), count=5)

1. store the sniffed packets in a **pcap** file.

wrpcap("<file name>", capture)

1. Opening GfG.pcap using Wireshark:Analyzing scapy sniffed packets in Wireshark
2. sniff packets offline from pcap files => sniff(offline="<file name>")

(Ref)

[**https://www.geeksforgeeks.org/packet-sniffing-using-scapy/**](https://www.geeksforgeeks.org/packet-sniffing-using-scapy/)