|  |  |  |
| --- | --- | --- |
| Name: Adarsh Kumar | SRN No: PES2UG20CS016 | Assignment No: 07 |
| Section: B | Date: 21/10/2022 |

|  |  |
| --- | --- |
| **Task 1:** | **Implementing a Simple Firewall** |
| Task 1.A: | Implement a Simple Kernel Module |
| Screenshot | Loading and removing a moduole to kernel.    We can see that our inserted kernel module is printing message to in the /var/log/syslog file. |
| Task 1.B: | Implement a Simple Firewall Using Netfilter. |
| 1.) | The above screenshot shows that the module is successfully compiled and loaded in the kernel.    We can see that is says that the connection time out no server could be reached.    Stopping the UDP packet of part 53 |
| 2.) | We can see that TCP, UDP Connections are working. |
| 3.) | Here we can see that All our ICMP Packets as well as Telnet Packets are dropped. |
| **Task 2** | **Experimenting with Stateless Firewall Rules** |
| Task 2.A: | Protecting the Router |
|  | Here we are accepting the ICMP Packet In, Out.    Questions:   1. Can you ping the router?   ANS: Yes, I can ping to the router   1. Can you telnet into the router?   ANS: No, I can’t able to telnet to the router because Telnet connection blocked.  Restoring the filter table to its original state |
| Task 2.B: | Protecting the Internal Network |
|  | 1. Outside hosts cannot ping internal hosts.      1. Outside hosts can ping the router.      1. Internal hosts can ping Outside Hosts.      1. All other packets between the internal and external networks should be blocked.     Cleaning iptables: |
| Task 2.C: | Protecting Internal Servers |
|  | 1. All the internal hosts run a telnet server (listening to port 23). Outside hosts can only access the telnet server on 192.168.60.5, not the other internal hosts      1. Outside hosts cannot access other internal servers.      1. Internal hosts can access all the internal servers.        1. Internal hosts cannot access external servers     Cleaning iptables: |
| **Task 3:** | **Connection Tracking and Stateful Firewall** |
| Task 3.A: | Experiment with the Connection Tracking |
|  | ICMP experiment:      How long can the ICMP connection state be kept?  Ans: ICMP requests has a default timeout of 30 seconds,  which you can change in the /proc/sys/net/ipv4/netfilter/ip\_ct\_icmp\_timeout entry.  UDP experiment:        How long can the UDP connection state be kept?  The default, UDP connection timeout 30 seconds. As soon we close the connection after 30 seconds router container 0 flow entries as shown.  TCP experiment:        Do you spot any difference?  Ans: In UDP, as soon as we close connection flow entries become 0 but in TCP flow entries doesn’t become 0 as we close our connection  How long can the TCP connection state be kept?  The default, the TCP connection timeout is 15 minutes it is a very long period of time so I haven’t waited to show that here, it is because the TCP have a three-way hand shake function |
| Task 3.B: | Setting Up a Stateful Firewall |
|  | On seed-router    1. All the internal hosts run a telnet server (listening to port 23). Outside hosts can only access the telnet server on 192.168.60.5, not the other internal hosts.    We are able to see that HOST -A can telnet to Host 1 internal server.  2. Outside hosts cannot access other internal servers.    We are able to see that HOST -A can’t able to telnet of Host 2 & Host 3 internal server.  3. Internal hosts can access all the internal servers      Taking internal Host 2 we can establish telnet connection to Host 1 & Host 3.  4. Internal hosts can access external servers.    Internal Host can even make tell net connection to external server Host-A.  Clean-up of Firewall rules |
| **Task 4:** | **Limiting Network Traffic** |
|  | On seed-Router      Here we can clearly see that initial 5 packets sequence no as regular but after that we were receiving  ICMP\_Sequence of jump 5 because we set constraint 10/packet per minute and but limit 5.  same task without the second rule –  On seed-Router      If second rule is omitted then we are able to get all the packet in ascending order  Please report your observation with screenshots and explain the purpose for each rule?   * 1st rule help to maintain 10 packet/minute and give a burst of 5 initial packet * 2nd rule gives a jump of 5 packet after bust time. |
| **Task 5:** | **Load Balancing** |
|  | Using the nth mode (round-robin) –    Implemented policies to equally divide the incoming packets between the three interval servers.  Host A    HOST 1 - 192.168.60.5    HOST 2 - 192.168.60.6    HOST 3 - 192.168.60.7    We can observe that when we enter in the text message in the HOST-A machine the message (the incoming packets) gets equally divided between the three interval servers, hence balancing the load on overall machine. The order of message is first one reaches to HOST 1 then HOST 2, then HOST 3 and next message again the next message reaches to HOST 1 |
|  | Using the random mode-    The following rule will select a matching packet with the probability P.  **Host A**    HOST 1 - 192.168.60.5    HOST 2 - 192.168.60.6    HOST 3 - 192.168.60.7    Here, in this rule unlike Round Robbin method this method selects the internal server based on probability it could be completely random no order is followed. In my case the arrival of the first packet is completely random and it(algorithm) could choose any internal server and the probability of getting selected is 33.33%. As you can see that the algorithm choice was HOST 1 for first packet so the next packet algorithm has 2 choices each having 50% chance of getting selected. In my case the algo choice was HOST 2, at the end algorithm got only one choice of HOST 3 with 100% probability of being selected. |

THE END