SDN Firewall with POX

Ruleset Overview

OMSCS/OMSCY CS 6250 Computer Networks

Topology Information

This network consists of the following hosts/networks:

- Headquarters Network (hq1-hq5). Subnet 10.0.0.0/24
- US Network (us1-us5). Subnet 10.0.1.0/24
- India Network (in1-in5). Subnet 10.0.2.0/24
- China Network (cn1-cn5). Subnet 10.0.3.0/24
- UK Network (uk1-uk5). Subnet 10.0.4.0/24

There is also a single host named wo1 that is at 10.0.200.1/32. This is to be used for testing a “world” scenario.

Ruleset Format

This file consists of several lines that describe each rule. A particular final rule may be any number of implementation rules (or lines) in this file. Each line has the following format:

Rule Number,Action,Source MAC,Destination MAC,Source IP,Destination IP,Protocol,Source Port,Destination Port,Comment/Note

- Rule Number = this is a rule number to help you track a particular rule - it is not used in the firewall implementation. It can be any value and is NOT validated in setup-firewall.py. DO NOT USE this field to match traffic.
- Action = Block or Allow Block rules will block traffic that matches the remaining parameters of this rule. Allow rules will override Block rules to allow specific traffic to pass through the firewall (see below for an example). The entry is a string in (Block,Allow).

Ruleset Format

Rule Number,Action,Source MAC,Destination MAC,Source IP,Destination IP,Protocol,Source Port,Destination Port,Comment/Note

- Source / Destination MAC address in form of xx:xx:xx:xx:xx:xx. You may use MAC Addresses to match an individual host. In the real world, you would use to match a particular piece of hardware. The MAC address of a particular host is defined inside the sdn-topology.py file.
- Source / Destination IP Address in form of xxx.xxx.xxx.xxx/xx in CIDR notation. You can use this to match either a single IP Address, a particular Subnet, or a particular subnetwork (based on CIDR). An entry here would look like: 10.0.0.1/32.

Ruleset Format

Rule Number,Action,Source MAC,Destination MAC,Source IP,Destination IP,Protocol,Source Port,Destination Port,Comment/Note

- Protocol = integer IP protocol number per IANA (0-254). An example is ICMP is IP Protocol 1, TCP is IP Protocol 6, etc. This must be an integer.
- Source / Destination Port = if Protocol is TCP or UDP, this is the Application Port Number per IANA. For example, web traffic is generally TCP Port 80.
- Comment/Note = this is for your use in tracking rules.

Ruleset Format

Special Notes about IP Addresses:

If you are using a CIDR mask, make sure that the IP Address shown is the Network Address. For instance, if you want to match the 192.168.1.x/24 subnet, the IP Address must be 192.168.1.0/24. This distinction becomes important if you use a sub-network CIDR mask – like a /30 network – the first address of that CIDR block is the network address you need to use. You can refer to https://github.com/att/pox/blob/master/pox/lib/addresses.py for more information on the format of CIDR addresses.

Any field not being used for a match should have a '-' character as its entry. A '-' means that the item is not being used for matching traffic. It is valid for any rule element except for Action to have a '-'. (i.e., a rule like: 1,Block,-,-,-,-,-,-,-,Block the world is valid)

Rules to Implement

You work for GT-SDN Corporation that has offices in the US, China, India, and the UK, with a US headquarters that acts as the datacenter for the company. Your task is to implement a firewall that accomplishes the following goals:

IMPORTANT NOTE:

To find the appropriate protocol and application port for a particular service, use https://en.wikipedia.org/wiki/List_of_TCP_and_UDP_port_numbers as your reference (linked in Project Description). Use the Ports for YES and Unofficial (in shades of green) – do not include Assigned ports (in Yellow).

Also, Time Protocol is NOT Network Time Protocol.

Rules to Implement

- On the headquarters network, you have two active DNS servers (using both the standard UDP service as well as the newer DNS-over-TLS standard). hq1 provides DNS service to the public and hq2 provides a private DNS service accessible only to the 5 corporate networks (i.e., the US, China, India, UK, and Headquarters network). Write a series of firewall rules that blocks the world (but allows the corporate networks) from accessing the DNS services on hq2 and ensures that the DNS services on hq1 are available to all.

  Hints:
  hq1 should be accessible to anyone – including the 5 corporate subnets.
  hq2 should only be accessible to the 5 corporate subnets.
  Should be two ports and two protocols for each.

Rules to Implement

- On the headquarters network, the host hq3 acts as a VPN server that connects to each of the other sites (hosts us3, uk3, in3, and cn3) using the OpenVPN server (standard ports – both TCP and UDP). Create a set of firewall rules that will only allow the 4 offsite hosts (us3, uk3, in3, and cn3) access to the hq3 OpenVPN server. No other hosts in the world should be able to access the OpenVPN server on hq3.

  Hints:
  Only us3, uk3, in3, and cn3 can access hq3 on the OpenVPN port. Use the standard Open VPN ports.

  Question:
  Can hq3 access a VPN server on hq3? Would this firewall stop it?

Rules to Implement

- Only the hosts on the Headquarters network should be pingable by the world. However, each host on a particular subnet (subnet list = the 5 different networks defined in the topology file) should be able to ping each other (i.e., us1 should be able to ping us5, but not in1). (Note that there are many different methods to reach this objective)

Hints:

The hq subnet can be pinged by anyone.

The in, cn, us, and uk subnets can only ping themselves.

The hq subnet should not be able to ping the in, cn, us, or uk subnets.

Blocking either the request or the response is an acceptable solution.

Rules to Implement

- One of the main routes for ransomware to enter a corporate network is through a Remote Desktop connection with either an insecure version of the server protocol or via leaked or weak credentials (using either the Microsoft Remote Desktop protocol or the Virtual Networking Computing (VNC) protocols). Write a set of firewall rules that will block any host in the world from connecting to a Remote Desktop connection on any of the 5 corporate networks (i.e., the US, China, India, UK, and Headquarters network) using the standard ports for the protocols.

Hints:

In this case, no computer on any network (including the hq/us/cn/uk/in subnets) should be able to access a remote desktop connection on any hq/us/cn/uk/in subnet).

Rules to Implement

- The servers located on hosts us3 and us4 run a micro webservice on TCP and UDP Port 8500 that processes financial information. Access to this service should be blocked from hosts uk2, uk3, uk4, uk5, in4, in5, us5, and hq5. (Hint: Note the IP Addresses for these hosts and you may use the smallest subnet mask that handles the listed hosts using CIDR notation).

  Hints:
  You can write individual rules to handle each case, or you can address the rule by using CIDR notation. Remember that if you use CIDR notation other than /32, you need to specify a network address.

Rules to Implement

- GT-SDN Corporation has banned the use of NetBIOS over TCPIP for the use of file and printer sharing. Write a series of rules that blocks NetBIOS over TCPIP to and from any host in the world (i.e., you can use a block for the world for this). Use the standard ports and protocols for NetBIOS over TCPIP.

Hints:
This also includes between hosts of the subnet. In other words, block incoming access to any of the 5 corporate subnets from anyone on the NetBIOS over TCPIP ports.

In the List of Services, use the three that are lablled NetBIOS.

Rules to Implement

- Block the ingress of the GRE IP Protocol into any of the 5 corporate networks (i.e., the US, China, India, UK, and Headquarters network). GRE is an encapsulation protocol primarily used in PPTP-based VPN connections. You only need to block the IP protocol and not any ports associated to PPTP-based VPN connections.

Hints:

Similar to the RDP rule, except this is only an IP Protocol.

Recommendations

Craft your ruleset in order to test all of the different parameters – i.e., have a rule use a Source IP and a Destination MAC, or create a random rule that has a source port.

Your code will be tested against an alternate configuration that WILL test all of the possible combinations of entries. The ruleset given has been crafted to allow you to potentially test all of these combinations.

The student provided autograder will only test if your submission meets the objective when used with your configuration file – it does not test the robustness of your code to ensure that you properly handle uncommon conditions.