**Name: Oscar Ynoa**

**Student Id: 00308560**

**Course: Linux Administration (CIS-245-O1A)**

**Subject: Firewall CentOS**

**Professor: Adrianna Holden-Gouveia**

**Date: 11/07/2021**

**Deliverables:**

**A text document including the answers to each question, (scripts or statements or commands are fine) but explain why you choose what you choose. Include any assumptions made about what you consider "dealing with" as well as the screenshots showing Wireshark to prove your script/statement/command works.**

**Documentation of your IPtables for each of your servers should include a list of changes to the chains you've made including which table you think needs to be changed when the change was made. Include a copy of your original IPtable and your updated IPtable at the end of this assignment.**

**A short document explaining how you think you could prevent a DDOS attack using IPtables. Cite at least 2 sources.**

**Explanation of what to do.**

**First, we are going to need Wireshark to prove the command works and list our iptables.**

**How to install Wireshark:**

**We already know that CentOS is a little bit tedious. Let start with becoming sudo.**

**We use su root and type our password.**

**Interfaz de usuario gráfica, Texto

Descripción generada automáticamente**

**Now let start with wireshark**

**We type the command dnf install wireshark-cli this is just to install Wireshark in the command line because we do not have a GUI.**

**Texto

Descripción generada automáticamente**

**We hit y**

**Texto

Descripción generada automáticamente**

**First step completed. Now for the next part of the assignment we need to list our iptables in the firewall.**

**Something funny. According to all the sources we must use sudo for running these commands. But because we are already in su root mode this saves typing this. Nice trick I would say.**

**The command to see the iptables is iptables -L or iptables –list. It works any way.**

**Texto

Descripción generada automáticamente con confianza media**

1. **Deal with web server (open needed ports, and forward port 80 traffic to 8080)**

**Let’s start with making the transfer from port 80 reaches the port 8080.**

**I chose this because after the research I got in the conclusion of this should be the easiest method to solve the transfer of data from port to port.**

**iptables -A INPUT -i eth0 -p tcp --dport 80 -j ACCEPT This allows me to confirm the rule before doing the redirecting command.**

**iptables -A INPUT -i eth0 -p tcp --dport 8080 -j ACCEPT Same as the other. Granting permission before executing the action.**

**iptables -A PREROUTING -t nat -i eth0 -p tcp --dport 80 -j REDIRECT --to-port 8080**

**This is the command that allows me to do the transfer but first I needed permission from the other 2 ports**

**I used the command iptables -t nat -L -n to display the ports using nat.**

**Interfaz de usuario gráfica, Texto

Descripción generada automáticamente**

**I used this because Wireshark don’t want to show ports on this one. Maybe because it is for http, and I don’t have any webpage opened in my server. I am running Wireshark on my home pc. But put the instructions of how to download it in the command line. I will try with the other one but in the server to see if I get any result.**

**Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media**

**I got some results. Command tshark -i ens33 -f "port 80 or port 8080"**

**Texto

Descripción generada automáticamente**

**/sbin/iptables -L -v -n --line-numbers List the rules and the list of your firewall.**

**Calendario

Descripción generada automáticamente**

**# /sbin/iptables -D INPUT 2 Used this to delete a firewall rule after typing something by accident.**

**iptables -t nat -D PREROUTING 1 and this one is for deleting routings. Just in case you typed the same twice by accident.**

1. **Deal with MySQL service (open needed ports)**

**The default port for MYSQL is 3306 which is used by the same client , connectors, and utilities. Btw MYSQL is an abbreviation of database. Unless we have something here, we are not getting anything. I will display the port of course.**

**I decided to use this as an easy method of displaying this port.**

**The command is iptables -A INPUT -p tcp --dport 3306 -j ACCEPT**

****

**iptables -L -nImagen de la pantalla de un celular con letras

Descripción generada automáticamente con confianza media**

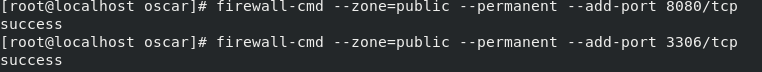
**The command is tshark -i ens33 -f "port 3306" The command works to display the port. Sadly, I am not getting anything. I will try pinging the port instead .**

**Texto

Descripción generada automáticamente**

**I tried opening the ports just in case**

**firewall-cmd --zone=public --permanent --add-port 3306/tcp**

****

**Make sure to reload the firewall.**

**Texto

Descripción generada automáticamente**

**firewall-cmd --list-all to make sure it worked**

**Texto

Descripción generada automáticamente**

**This should give us what we need to work with those ports.**

**Ping 192.168.40.130 3306 I used this command to ping my port 3306. We just needed the Ip of our server and the desire port to be pinged.**

**Texto

Descripción generada automáticamente**

**At least I got something there.**

**I used telnet in this one. Same as ping but with the variation of telnet.**

**telnet 192.168.40.130 3306**

**Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente**

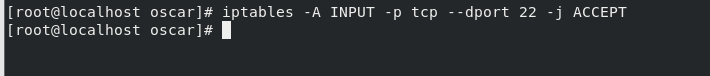
**The tshark command will not give me an output or something there unless I have some kind of traffic coming trough that port. But at least I checked the state of the ports and the availability in all of the assigned one’s.**

1. **Deal with SSH service (allow incoming and outgoing SSH, second script to deny SSH)**

**Let’s start with the basic then. SSH is a protocol for securely exchanging data between two computers over an untrusted network. We will allow first the incoming and outgoing SSH.**

**To allow incoming traffic from SSH :**

**iptables -A INPUT -p tcp --dport 22 -j ACCEPT Asking permission before the execution**

****

**To allow outgoing traffic from SSH:**

**iptables -A OUTPUT -p tcp --dport 22 -j ACCEPT**

****

**Now let’s Deny SSH.**

**iptables -I INPUT -p tcp --dport 22 -j DROP**

**We use Drop to discharge all traffic. -I it is used to insert a new rule instead of overwriting it like what could happened when you append something for example using this with -A. With this anybody who try to abuse of this port could be stopped.**

****

**Here we have the evidence of dropping the port 22 that is SSH.**

**Imagen que contiene teclado, computadora

Descripción generada automáticamente**

**I used telnet 192.168.40.130 22 to ping the port.**

**Texto

Descripción generada automáticamente**

1. **Deal with Email service (Such as allow or block incoming SMTP,POP3...)**

**They are a couple of ports here that allow SMPT ports. These ports are 25, 587, 465, 2525.**

**To allow incoming traffic from SMTP :**

**iptables -A INPUT -p tcp --dport 25 -j ACCEPT**

**iptables -A INPUT -p tcp --dport 587 -j ACCEPT**

**iptables -A INPUT -p tcp --dport 465 -j ACCEPT**

**iptables -A INPUT -p tcp --dport 2525 -j ACCEPT**

**Texto

Descripción generada automáticamente**

**To Block incoming traffic from SMTP:**

**iptables -A INPUT -p tcp --dport 25 -j DROP**

**iptables -A INPUT -p tcp --dport 587 -j DROP**

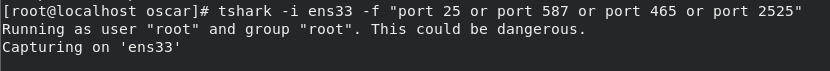
**iptables -A INPUT -p tcp --dport 465 -j DROP**

**iptables -A INPUT -p tcp --dport 2525 -j DROP**

**Un conjunto de letras blancas en un fondo blanco

Descripción generada automáticamente con confianza media**

**This is how we allow and block that traffic for the email service. Luckily, we don’t have to add the ports like the 8080 or 3306. Now let’s check if these ports receive any signal.**

****

**I always run tshark but as I said. I don’t have any kind of traffic. But I can ping the ports like in the other examples to ensure they are listening to the request. But instead in this one I will list it with the rules in the iptable.**

**Imagen que contiene Calendario

Descripción generada automáticamente**

**POP3**

**This is also for mail services. For this the ports are 143 and 993.So then, Let’s start with allowing the traffic from these ports.**

**iptables -A INPUT -p tcp --dport 143 -j ACCEPT**

**iptables -A INPUT -p tcp --dport 993 -j ACCEPT**

**Texto

Descripción generada automáticamente**

**To block Incoming traffic:**

**iptables -A INPUT -p tcp --dport 143 -j DROP**

**iptables -A INPUT -p tcp --dport 993 -j DROP**

**Texto

Descripción generada automáticamente**

1. **Script to allow/block specific hosts, MAC addresses**

**Let’s say there is a specific user trying to do something malicious. How do I block them then? That would be easier than you think. Let’s say am the one doing it. I would grab the Ip of the user and block from sending traffic.**

**iptables -I INPUT -s 123.456.789.1 -j DROP This Ip is an example. Just Write a regular Ip and works.**

**How to allow incoming connection from a specific host**

**iptables -I INPUT -s 123.456.789.1 -j ACCEPT Same Here. This Ip are examples.**

**To Block specific MAC addresses:**

**iptables -A INPUT -m mac --mac-source 00:0a:95:9d:68:16 -j DROP**

**To use the MAC address function, we must type : -m mac and --mac-source to specify what is the MAC source.**

**To Allow specific MAC addresses:**

**iptables -A INPUT -m mac --mac-source 00:0a:95:9d:68:16 -j ACCEPT**

1. **A script/command to block telnet, block ping**

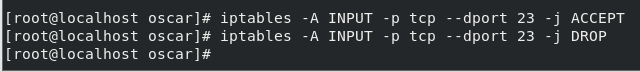
**The port 23 is telnet by default.**

**This is for allowing telnet communication.**

**iptables -A INPUT -p tcp --dport 23 -j ACCEPT**

**How to block incoming telnet communication.**

**iptables -A INPUT -p tcp --dport 23 -j DROP**

****

**How to disable ping request.**

**iptables -A INPUT -p --icmp-type echo-request -j REJECT**

**The ping request uses ICMP packages.**

**To allow back the ping request we use this command.**

**iptables -D INPUT -p icmp --icmp-type echo-request -j REJECT**

**The -D function will delete the previous rule allowing us to stablish the ping command again.**

1. **Use Wireshark to show that your statements/scripts are working properly on BOTH servers.**
2. **Write the specs for how you think you could prevent a DDOS attack using Iptables.  Can you write a script for this?**

**Let’s start with some basics here. I will ping myself to see how my server is doing so far. In order to do that. I type ping followed with my Ip. In this case ping 192.168.40.130**

**Texto

Descripción generada automáticamente**

**Everything seems normal so far. But let’s say someone could be trying to perform an attack trough my ports. I will use the following commands or rules to stop a couple of basic and most common DDOS attacks.**

**/sbin/iptables -A INPUT -p tcp --destination-port 80 -j DROP This one is for dropping the port 80. Http cross trough this port.**

**iptables -t mangle -A PREROUTING -p icmp -j DROP This drops all ICMP packets. ICMP is only used to ping a host to find out if it’s still alive. With this I will stop slowing the network so far. But they are other’s that could help me a little more with this.**

**iptables -t mangle -A PREROUTING -f -j DROP For example, this one is for blocking fragmented packages.**

**iptables -t mangle -A PREROUTING -m conntrack --ctstate INVALID -j DROP This rule blocks all packets that are not a SYN packet and don’t belong to an established TCP connection.**

**iptables -A OUTPUT -p udp -j DROP This one for dropping udp traffic.**

**So, let’s get into context. I imagine myself working with my server when I detect some slowing down with the traffic on my network. I will execute a script runed with the previous commands to stop the most commons DDOS attacks. It could be another type, but I believe this that I presented would fit more in the common terms. Hopefully by dropping those fragmented packages, ping connections, udp traffic and else. My network would stablish itself in a steadier way.**

**References**

[**https://linuxconfig.org/install-wireshark-on-linux-redhat-8**](https://linuxconfig.org/install-wireshark-on-linux-redhat-8)

[**https://linuxhint.com/wireshark-command-line-interface-tshark/**](https://linuxhint.com/wireshark-command-line-interface-tshark/)

[**https://www.wireshark.org/docs/man-pages/tshark.html**](https://www.wireshark.org/docs/man-pages/tshark.html)

[**https://www.wireshark.org/docs/wsug\_html\_chunked/ChCustCommandLine.html**](https://www.wireshark.org/docs/wsug_html_chunked/ChCustCommandLine.html)

[**https://www.cyberciti.biz/faq/how-to-list-all-iptables-rules-in-linux/**](https://www.cyberciti.biz/faq/how-to-list-all-iptables-rules-in-linux/)

[**https://serverfault.com/questions/320614/how-to-forward-port-80-to-another-port-on-the-samemachine**](https://serverfault.com/questions/320614/how-to-forward-port-80-to-another-port-on-the-samemachine)

[**https://www.cyberciti.biz/faq/deleting-firewall-rules/**](https://www.cyberciti.biz/faq/deleting-firewall-rules/)

[**https://serverfault.com/questions/159544/undoing-port-forwarding**](https://serverfault.com/questions/159544/undoing-port-forwarding)

[**https://dev.mysql.com/doc/mysql-port-reference/en/mysql-ports-reference-tables.html**](https://dev.mysql.com/doc/mysql-port-reference/en/mysql-ports-reference-tables.html)

[**https://serverfault.com/questions/610332/iptables-allow-outgoing-ssh**](https://serverfault.com/questions/610332/iptables-allow-outgoing-ssh)

[**https://www.thegeekstuff.com/2011/03/iptables-inbound-and-outbound-rules/**](https://www.thegeekstuff.com/2011/03/iptables-inbound-and-outbound-rules/)

[**https://www.sparkpost.com/blog/what-smtp-port/**](https://www.sparkpost.com/blog/what-smtp-port/)

[**https://www.digitalocean.com/community/tutorials/iptables-essentials-common-firewall-rules-and-commands**](https://www.digitalocean.com/community/tutorials/iptables-essentials-common-firewall-rules-and-commands)

[**https://docs.rackspace.com/support/how-to/open-ports-in-the-linux-firewall-to-access-pop-and-imap-mail-servers/**](https://docs.rackspace.com/support/how-to/open-ports-in-the-linux-firewall-to-access-pop-and-imap-mail-servers/)

[**https://www.cyberciti.biz/tips/howto-block-ipaddress-with-iptables-firewall.html**](https://www.cyberciti.biz/tips/howto-block-ipaddress-with-iptables-firewall.html)

[**https://unix.stackexchange.com/questions/11851/iptables-allow-certain-ips-and-block-all-other-connection**](https://unix.stackexchange.com/questions/11851/iptables-allow-certain-ips-and-block-all-other-connection)

[**https://serverfault.com/questions/519480/iptables-allow-access-from-certain-mac-addresses**](https://serverfault.com/questions/519480/iptables-allow-access-from-certain-mac-addresses)

[**https://tecadmin.net/mac-address-filtering-using-iptables/**](https://tecadmin.net/mac-address-filtering-using-iptables/)

[**https://unix.stackexchange.com/questions/207697/how-to-block-telnet-using-iptables-command**](https://unix.stackexchange.com/questions/207697/how-to-block-telnet-using-iptables-command)

[**https://tecadmin.net/block-ping-responses-in-linux/**](https://tecadmin.net/block-ping-responses-in-linux/)

[**https://www.layerstack.com/resources/tutorials/How-to-enable-and-disable-Ping-from-IPTables-on-Linux-Cloud-Servers**](https://www.layerstack.com/resources/tutorials/How-to-enable-and-disable-Ping-from-IPTables-on-Linux-Cloud-Servers)

[**https://www.crybit.com/prevent-dos-attack/**](https://www.crybit.com/prevent-dos-attack/)

[**https://serverfault.com/questions/583523/linux-command-to-prevent-dos-attack-by-using-netstat-and-iptables**](https://serverfault.com/questions/583523/linux-command-to-prevent-dos-attack-by-using-netstat-and-iptables)

[**https://www.agix.com.au/stop-ddos-attack-using-iptables/**](https://www.agix.com.au/stop-ddos-attack-using-iptables/)

[**https://devconnected.com/how-to-ping-specific-port-number/**](https://devconnected.com/how-to-ping-specific-port-number/)

[**https://linuxconfig.org/redhat-8-open-and-close-ports**](https://linuxconfig.org/redhat-8-open-and-close-ports)

[**https://docs.rackspace.com/support/how-to/block-an-ip-address-on-a-Linux-server/**](https://docs.rackspace.com/support/how-to/block-an-ip-address-on-a-Linux-server/)

[**https://www.youtube.com/watch?v=eZYtnzODpW4&ab\_channel=NetworkChuck**](https://www.youtube.com/watch?v=eZYtnzODpW4&ab_channel=NetworkChuck)

[**https://javapipe.com/blog/iptables-ddos-protection/**](https://javapipe.com/blog/iptables-ddos-protection/)