CSCI4113

**LAB6 Notes** Milan Formanek April 22, 2019

# Configuring IPtables firewall settings for the DM network

## For this lab the DunderMifflin network gets it’s firewalls configured. Each of the machines A through F have to have unique IPtables rules based on their uses and running services. To make the deployment easy a .sh script is created, grabbed from github and run with the right parameters for each machine.

1. Configuring Machine B and F - Web Servers

Machines B and F have identical configuration allowing http and https traffic incoming from any ip along with local loopback, icmp traffic and ssh from the local networks. This is done by deploying the LAB6.sh script to the machine with arguments B and F:

1. [ root@carriage ~]# wget https ://raw. githubusercontent .com/ mformanek /Linux -Sys -Admin --- LAB6 / master / LAB6 .sh

- -2019 -04 -21 23:20:09 - - https ://raw. githubusercontent .com/ mformanek /Linux -Sys -Admin --- LAB6 /

master / LAB6 .sh

1. Resolving raw. githubusercontent .com (raw. githubusercontent .com)... 151.101.68.133
2. Connecting to raw. githubusercontent .com (raw. githubusercontent .com) |151.101.68.133|:443... connected .
3. HTTP request sent , awaiting response ... 200 OK
4. Length : 6233 (6.1 K) [ text / plain ]
5. Saving to: ‘LAB6 .’sh

7 100%[=============================================================================== >] 6 ,233 --.-K/s in 0s

8 2019 -04 -21 23:20:09 (29.4 MB/s) - ‘LAB6 .’sh saved [6233 / 6233]

1. [ root@carriage ~]# bash ./ LAB6 .sh B
2. iptables : Saving firewall rules to /etc/ sysconfig / iptables :[ OK ]
3. [ root@saddle ~]# wget https ://raw. githubusercontent .com/ mformanek /Linux -Sys -Admin --- LAB6 / master / LAB6 .sh
4. - -2019 -04 -21 23:25:59 - - https ://raw. githubusercontent .com/ mformanek /Linux -Sys -Admin --- LAB6 / master / LAB6 .sh
5. Resolving raw. githubusercontent .com (raw. githubusercontent .com)... 151.101.68.133
6. Connecting to raw. githubusercontent .com (raw. githubusercontent .com) |151.101.68.133|:443... connected .
7. HTTP request sent , awaiting response ... 200 OK
8. Length : 6233 (6.1 K) [ text / plain ]
9. Saving to: ‘LAB6 .’sh

8 100%[============================================================================ >] 6 ,233 --.-K/s in 0s

9 2019 -04 -21 23:25:59 (22.9 MB/s) - ‘LAB6 .’sh saved [6233 / 6233]

1. [ root@saddle ~]# bash ./ LAB6 .sh F
2. iptables : Saving firewall rules to /etc/ sysconfig / iptables :[ OK ]

# Configuring Machine C - FTP Server

## Machine C allows FTP connections from the 100.64.0/16 subnet along with local loop- back, icmp traffic, DNS requests from Machine D, http and https outbound traffic and ssh from the local networks. This is done by deploying the LAB6.sh script to the machine with argument C:

1. [ root@platen ~]# wget https ://raw. githubusercontent .com/ mformanek /Linux -Sys -Admin --- LAB6 / master / LAB6 .sh
2. - -2019 -04 -22 00:32:11 - - https ://raw. githubusercontent .com/ mformanek /Linux -Sys -Admin --- LAB6 / master / LAB6 .sh
3. Resolving raw. githubusercontent .com (raw. githubusercontent .com)... 151.101.68.133
4. Connecting to raw. githubusercontent .com (raw. githubusercontent .com) |151.101.68.133|:443... connected .
5. HTTP request sent , awaiting response ... 200 OK
6. Length : 6233 (6.1 K) [ text / plain ]
7. Saving to: ‘LAB6 .’sh

8 100%[============================================================================ >] 6 ,233 --.-K/s in 0s

9 2019 -04 -22 00:32:11 (26.6 MB/s) - ‘LAB6 .’sh saved [6233 / 6233]

1. [ root@platen ~]# bash ./ LAB6 .sh C
2. iptables : Saving firewall rules to /etc/ sysconfig / iptables :[ OK ]

# Configuring Machine D - DNS Server

## Machine D allows DNS queries from any source along with local loopback, icmp traffic and ssh from the local networks. This is done by deploying the LAB6.sh script to the machine with argument D:

1. [ root@chase ~]# wget https ://raw. githubusercontent .com/ mformanek /Linux -Sys -Admin --- LAB6 / master / LAB6 .sh
2. - -2019 -04 -21 23:32:30 - - https ://raw. githubusercontent .com/ mformanek /Linux -Sys -Admin --- LAB6 / master / LAB6 .sh
3. Resolving raw. githubusercontent .com (raw. githubusercontent .com)... 151.101.68.133
4. Connecting to raw. githubusercontent .com (raw. githubusercontent .com) |151.101.68.133|:443... connected .
5. HTTP request sent , awaiting response ... 200 OK
6. Length : 6233 (6.1 K) [ text / plain ]
7. Saving to: ‘LAB6 .’sh

8

9 100%[========================================================= >] 6 ,233 --.-K/s in 0s

10

11 2019 -04 -21 23:32:30 (25.0 MB/s) - ‘LAB6 .’sh saved [6233 / 6233]

12

1. [ root@chase ~]# bash ./ LAB6 .sh D
2. iptables : Saving firewall rules to /etc/ sysconfig / iptables :[ OK ]

# Configuring Machine E - File Server

## Machine E allows for Samba connections from the 10.21.32.0/24 subnet along with local loopback, icmp traffic and ssh from the same subnet. This is done by deploying the LAB6.sh script to the machine with argument E:

1. [ root@roller ~]# wget https ://raw. githubusercontent .com/ mformanek /Linux -Sys -Admin --- LAB6 / master / LAB6 .sh
2. - -2019 -04 -21 23:47:44 - - https ://raw. githubusercontent .com/ mformanek /Linux -Sys -Admin --- LAB6 / master / LAB6 .sh
3. Resolving raw. githubusercontent .com (raw. githubusercontent .com)... 151.101.68.133
4. Connecting to raw. githubusercontent .com (raw. githubusercontent .com) |151.101.68.133|:443... connected .
5. HTTP request sent , awaiting response ... 200 OK
6. Length : 6233 (6.1 K) [ text / plain ]
7. Saving to: ‘LAB6 .’sh

8

9 100%[========================================================================== >] 6 ,233 --.-K/s in 0s

10 2019 -04 -21 23:47:44 (22.5 MB/s) - ‘LAB6 .’sh saved [6233 / 6233]

1. [ root@roller ~]# bash ./ LAB6 .sh E
2. iptables : Saving firewall rules to /etc/ sysconfig / iptables :[ OK ]

# Configuring Machine A - Router

## Finally the most complicated one, Machine A. It has to forward packets to the other computers in the DM network. It also mirrors the firewall settings on the individual machines in order to provide an extra layer of security on the network. Rules to block Facebook.com,icanhas.cheezburger.com and cheezburger.com are also implemented here. Icanhas.cheezburger.com and cheezburger.com have the same IP making life a little easier.

1. [ root@router ~]# wget https ://raw. githubusercontent .com/ mformanek /Linux -Sys -Admin --- LAB6 / master / LAB6 .sh
2. - -2019 -04 -22 18:36:56 - - https ://raw. githubusercontent .com/ mformanek /Linux -Sys -Admin --- LAB6 / master / LAB6 .sh
3. Resolving raw. githubusercontent .com (raw. githubusercontent .com)... 151.101.68.133
4. Connecting to raw. githubusercontent .com (raw. githubusercontent .com) |151.101.68.133|:443... connected .
5. HTTP request sent , awaiting response ... 200 OK
6. Length : 6136 (6.0 K) [ text / plain ]
7. Saving to: ‘LAB6 .’sh

8 100%[============================================================================================================= >]

6 ,136 --.-K/s in 0s

9 2019 -04 -22 18:36:56 (19.2 MB/s) - ‘LAB6 .’sh saved [6136 / 6136]

1. [ root@router ~]# bash ./ LAB6 .sh A
2. iptables : Saving firewall rules to /etc/ sysconfig / iptables :[ OK ]

# LAB6.sh - Deployment Script

1 #!/bin/ bash

2 # ------------------------------------------------------------------

3 # By Milan Formanek LAB6 Deployment Script

4 # ------------------------------------------------------------------

1. VERSION =0.1.0
2. SUBJECT =some -unique -id
3. USAGE ="Run on the individual DM machines with the letter name of the machine as the parameter ."

8

9 if [ $# == 0 ] ; then

1. echo $ USAGE
2. exit 1;
3. fi

13

1. if [ $1 == "OFF" ] ; then # ENABLE EVERITHING IN IPTABLES for testing
2. iptables -P INPUT ACCEPT
3. iptables -P FORWARD ACCEPT
4. iptables -P OUTPUT ACCEPT
5. iptables -F # reset IPTABLES
6. service iptables save # make sure to save rules !!!
7. exit 1;
8. fi

22

23 iptables -F # reset IPTABLES

24

25 iptables -A INPUT -m conntrack -- ctstate ESTABLISHED , RELATED -j ACCEPT # Allow return packets for ESTABLISHED and RELATED packets

26

1. iptables -P INPUT DROP # set default DROP policy
2. iptables -P OUTPUT ACCEPT

29

1. iptables -A INPUT -i lo -j ACCEPT # Allow loopback traffic
2. iptables -A OUTPUT -o lo -j ACCEPT

32

1. iptables -A INPUT -p icmp --icmp - type echo - request -j ACCEPT
2. iptables -A INPUT -p icmp --icmp - type echo - reply -j ACCEPT
3. iptables -A INPUT -p icmp --icmp - type time - exceeded -j ACCEPT
4. iptables -A INPUT -p icmp --icmp - type destination - unreachable -j ACCEPT # ACCEPT ICMP packets .

37

1. if [ $1 != "E" ] ; then
2. iptables -A INPUT -p tcp -s 100.64.0.0 /16 --dport 22 -m state --state NEW , ESTABLISHED -j ACCEPT
3. iptables -A INPUT -p tcp -s 10.21.32.0 /24 --dport 22 -m state --state NEW , ESTABLISHED -j ACCEPT
4. iptables -A INPUT -p tcp -s 198.18.0.0 /16 --dport 22 -m state --state NEW , ESTABLISHED -j ACCEPT
5. iptables -A OUTPUT -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT
6. #On all machines exlcuding E allow inbound ssh connections from the 100.64.0.0 /16, 10.21.32.0 /24, and 198.18.0.0 /

16 subnets

1. fi

45

46 if [ $1 != "A" ] ; then

1. iptables -P FORWARD DROP # disble forwarding on non routers
2. else # RULES FOR ROUTER / MACHINE A
3. iptables -P FORWARD DROP # enable forwarding on routers
4. iptables -A FORWARD -s 157.240.28.35 -j DROP
5. iptables -A FORWARD -d 157.240.28.35 -j DROP # block FACEBOOK
6. iptables -A FORWARD -s 216.176.177.74 -j DROP
7. iptables -A FORWARD -d 216.176.177.74 -j DROP # block CHEESEBURGER .com

54

1. iptables -A FORWARD -p tcp -s 100.64.0.0 /16 --dport 22 -m state --state NEW , ESTABLISHED -j ACCEPT
2. iptables -A FORWARD -p tcp -s 10.21.32.0 /24 --dport 22 -m state --state NEW , ESTABLISHED -j ACCEPT
3. iptables -A FORWARD -p tcp -s 198.18.0.0 /16 --dport 22 -m state --state NEW , ESTABLISHED -j ACCEPT
4. iptables -A FORWARD -m state --state ESTABLISHED , RELATED -j ACCEPT # FORWARD SSH

59

1. iptables -A FORWARD -p tcp --dport 80 -m state --state NEW , ESTABLISHED , RELATED -j ACCEPT
2. iptables -A FORWARD -p tcp --dport 443 -m state --state NEW , ESTABLISHED , RELATED -j ACCEPT

62

1. iptables -A FORWARD -p icmp --icmp - type echo - request -j ACCEPT
2. iptables -A FORWARD -p icmp --icmp - type echo - reply -j ACCEPT
3. iptables -A FORWARD -p icmp --icmp - type time - exceeded -j ACCEPT
4. iptables -A FORWARD -p icmp --icmp - type destination - unreachable -j ACCEPT # ACCEPT ICMP packets

67

68

1. iptables -A FORWARD -p udp --sport 1024:65535 --dport 53 -m state --state NEW , ESTABLISHED -j ACCEPT
2. iptables -A FORWARD -p udp --sport 53 --dport 1024:65535 -m state --state ESTABLISHED -j ACCEPT
3. iptables -A FORWARD -p udp --sport 53 --dport 53 -m state --state NEW , ESTABLISHED -j ACCEPT
4. iptables -A FORWARD -p udp --sport 53 --dport 53 -m state --state ESTABLISHED -j ACCEPT
5. # allow inbound DNS lookup on chase

74

75

76

77 iptables -A FORWARD -m conntrack -- ctstate ESTABLISHED , RELATED -j ACCEPT # Allow return packets for ESTABLISHED and RELATED packets

78

79

80 fi

81

1. if [ $1 == "B" ] || [ $1 == "F" ] ; then # RULES FOR MACHINE B AND F
2. iptables -A INPUT -p tcp -m tcp --dport 80 -j ACCEPT
3. iptables -A INPUT -p tcp -m tcp --dport 443 -j ACCEPT # allow http and https inbound traffic
4. fi

86

1. if [ $1 == "C" ] ; then # RULES FOR MACHINE C
2. iptables -P OUTPUT DROP # defaul output drop

89

1. iptables -A INPUT -m state --state ESTABLISHED , RELATED -j ACCEPT
2. iptables -A OUTPUT -m state --state ESTABLISHED , RELATED -j ACCEPT # allow related connections

92

1. iptables -A OUTPUT -p udp -d 100.64.21.4 --dport 53 -m state --state NEW , ESTABLISHED -j ACCEPT
2. iptables -A INPUT -p udp -s 100.64.21.4 --sport 53 -m state --state ESTABLISHED -j ACCEPT
3. iptables -A OUTPUT -p tcp -d 100.64.21.4 --dport 53 -m state --state NEW , ESTABLISHED -j ACCEPT
4. iptables -A INPUT -p tcp -s 100.64.21.4 --sport 53 -m state --state ESTABLISHED -j ACCEPT # allow DNS lookup

on chase

97

1. iptables -A OUTPUT -p tcp -m tcp --dport 80 -j ACCEPT
2. iptables -A OUTPUT -p tcp -m tcp --dport 443 -j ACCEPT # allow outbound http and https traffic

100

101 iptables -A OUTPUT -p tcp --dport 22 -j ACCEPT # allow outgoing ssh

102

1. iptables -A OUTPUT -p icmp --icmp - type echo - request -j ACCEPT
2. iptables -A OUTPUT -p icmp --icmp - type echo - reply -j ACCEPT
3. iptables -A OUTPUT -p icmp --icmp - type time - exceeded -j ACCEPT
4. iptables -A OUTPUT -p icmp --icmp - type destination - unreachable -j ACCEPT # ACCEPT outbound ICMP packets # allow

outgoing ICMP

107

1. iptables -A OUTPUT -p tcp --sport 21 -m state --state ESTABLISHED -j ACCEPT
2. iptables -A OUTPUT -p tcp --sport 20 -m state --state ESTABLISHED , RELATED -j ACCEPT
3. iptables -A OUTPUT -p tcp --sport 1024: --dport 1024: -m state --state ESTABLISHED -j ACCEPT
4. iptables -A INPUT -p tcp --dport 21 -m state --state NEW , ESTABLISHED -j ACCEPT
5. iptables -A INPUT -p tcp --dport 20 -m state --state ESTABLISHED -j ACCEPT
6. iptables -A INPUT -p tcp --sport 1024: --dport 1024: -m state --state ESTABLISHED , RELATED ,NEW -j ACCEPT
7. #FTP Rules
8. fi

116

117 if [ $1 == "D" ] ; then # RULES FOR MACHINE D - DNS SERVER

118 SERVER\_IP="100.64.21.4"

1. iptables -A INPUT -p udp -s 0/0 --sport 1024:65535 -d $ SERVER \_IP --dport 53 -m state --state NEW , ESTABLISHED -j

ACCEPT

1. iptables -A OUTPUT -p udp -s $ SERVER \_IP --sport 53 -d 0/0 --dport 1024:65535 -m state --state ESTABLISHED -j

ACCEPT

1. iptables -A INPUT -p udp -s 0/0 --sport 53 -d $ SERVER \_IP --dport 53 -m state --state NEW , ESTABLISHED -j ACCEPT
2. iptables -A OUTPUT -p udp -s $ SERVER \_IP --sport 53 -d 0/0 --dport 53 -m state --state ESTABLISHED -j ACCEPT
3. # allow inbound DNS lookup on chase .
4. fi

125

1. if [ $1 == "E" ] ; then # RULES FOR MACHINE E - FILE SERVER
2. iptables -A INPUT -p tcp -s 10.21.32.0 /24 --dport 22 -m state --state NEW , ESTABLISHED -j ACCEPT
3. iptables -A OUTPUT -p tcp --sport 22 -m state --state ESTABLISHED -j ACCEPT # enable SSH connections only from

10.21.32.0 /24 net

129

1. iptables -A INPUT -m state --state NEW -p udp --dport 137 -j ACCEPT
2. iptables -A INPUT -m state --state NEW -p udp --dport 138 -j ACCEPT
3. iptables -A INPUT -m state --state NEW -p tcp --dport 139 -j ACCEPT
4. iptables -A INPUT -m state --state NEW -p tcp --dport 445 -j ACCEPT
5. # allow incoming connections for CIFS and SMB
6. fi

136

137 service iptables save # make sure to save rules !!!