Elijah Berumen
CSCI 4113
Lab 6: Firewalling and IPTables

Prerequisites

```
# yum install nc tcpdump nmap
```

All Machines

1. *Allow all traffic to and from the local loopback adapter lo, so each machine can talk to itself.*

On each machine, do the following

Note: The -i flag specifies input interface, and -o specified output interface

```
# iptables -A INPUT -i lo -j ACCEPT

# iptables -A OUTPUT -o lo -j ACCEPT
```

2. *Allow inbound icmp traffic for echo-request, echo-reply (ping), time-exceeded (traceroute), or destination-unreachable. (This lets ping and traceroute work but drops other commonly abused icmp packets.)*

For ping and traceroute functionality

```
# iptables -A INPUT -p icmp --icmp-type echo-request -j ACCEPT

# iptables -A INPUT -p icmp --icmp-type echo-reply -j ACCEPT

# iptables -A INPUT -p icmp --icmp-type time-exceeded -j ACCEPT

# iptables -A INPUT -p icmp --icmp-type destination-unreachable -j ACCEPT
```

3. On all machines, **except Machine E**, allow inbound ssh connections from the 100.64.0.0/16, 10.21.32.0/24, and 198.18.0.0/16 subnets. (This limits the subnets from which ssh can be initiated).

```
# iptables -A INPUT -p tcp -s 100.64.0.0/24 --dport=22 -j ACCEPT

# iptables -A INPUT -p tcp -s 10.21.32.0/24 --dport=22 -j ACCEPT

# iptables -A INPUT -p tcp -s 10.21.32.0/24 --dport=22 -j ACCEPT
```

4. *All machines should implement a default deny policy for inbound traffic.*

```
# iptables -P INPUT DROP
```

5. *Machines should also prevent bad actors from "bouncing" or forwarding packets through them if they're not intended as routers. This should be done both with ip forwarding disabled in the kernel and through the forward chain.*

For everything except A(our actual router) do the 2 following things

```
# iptables -P FORWARD DROP
```

Disabling ip forwarding in the kernel in /etc/sysctl.config:

```
net.ipv4.ip_forward = 0
```

For later:

```
# Allow DNS Lookups
iptables -A INPUT -p udp --sport 53 -s 100.64.5.4 -j ACCEPT
iptables -A OUTPUT -p udp --dport 53 -s 100.64.5.4 -j ACCEPT
iptables -A INPUT -p tcp --sport 53 -s 100.64.5.4 -j ACCEPT
iptables -A OUTPUT -p tcp --dport 53 -s 100.64.5.4 -j ACCEPT
```

Machine A/Router

1. *Allow the appropriate DHCP traffic to/from 100.64.N.0/24 & 10.21.32.0/24 (So your other machines can get their configs).*

For first subnet

```
# iptables -A OUTPUT -p udp -d 100.64.5.0/16 --dport=68 -j ACCEPT

# iptables -A INPUT -p udp -s 100.64.5.0/16 --dport=67 -j ACCEPT
```

Now for the other subnet

```
# iptables -A OUTPUT -p udp -d 10.21.32.0/24 --dport=68 -j ACCEPT

# iptables -A INPUT -p udp -s 10.21.32.0/24 --dport=67 -j ACCEPT
```

2. *Deny your users access to Facebook from any machine on your network. You need not block all Facebook IP addresses, just the one you receive from a one-time resolve of facebook.com. To go above and beyond the requirements by blocking all Facebook IP addresses, get a list as follows:*
   root@machineA# yum install jwhois
   root@machineA# whois -h whois.radb.net '!gAS32934'
3. *Deny your users access to icanhas.cheezburger.com and cheezburger.com. Again, you need not block all such IP addresses, just the ones you receive from a one-time resolve.*

Block Facebook

Facebook IP 157.240.28.35

```
# iptables -A INPUT -s 157.240.28.35 -j DROP
# iptables -A OUTPUT -d 157.240.28.35 -j DROP
# iptables -A FORWARD -s 157.240.28.35 -j DROP
# iptables -A FORWARD -d 157.240.28.35 -j DROP
```

Block Cheezburger
Cheezburger IP 216.176.186.210

```
# iptables -A INPUT -s 216.176.186.210 -j DROP
# iptables -A OUTPUT -d 216.176.186.210 -j DROP
# iptables -A FORWARD -s 216.176.186.210 -j DROP
# iptables -A FORWARD -d 216.176.186.210 -j DROP
```

4. *Only forward packets to/from machines behind the router, based on the intended purpose of that specific machine. In other words, there should be rules on Machine A that mimic the rules for Machines B-F. This is an added layer of security in case the firewall on one of the other machines is inadvertently dropped.*

Note: Intended packets are redirected to a User-defined chain for each machine

##### Machine A Double Layer Chains#####

User Chains(one for each type of machine so it would scale better)

```
iptables -N WEB
iptables -N FTP
iptables -N DNS
iptables -N FILE
```

Send to B Chain

```
iptables -A FORWARD -d 100.64.5.2 -j WEB
iptables -A FORWARD -s 100.64.5.2 -j WEB
```

Send to F Chain(which is the same as B chain)

```
iptables -A FORWARD -d 100.64.5.5 -j WEB
iptables -A FORWARD -s 100.64.5.5 -j WEB
```

Send to C Chain

```
iptables -A FORWARD -d 100.64.5.3 -j FTP
iptables -A FORWARD -s 100.64.5.3 -j FTP
```

Send to D Chain

```
iptables -A FORWARD -d 100.64.5.4 -j DNS
iptables -A FORWARD -s 100.64.5.4 -j DNS
```

Send to E Chain

```
iptables -A FORWARD -d 10.21.32.2 -j FILE
iptables -A FORWARD -s 10.21.32.2 -j FILE
```

B and F Rules for HTTP/HTTPS(WEB Chain)

```
iptables -A WEB -p tcp --dport=80 -j ACCEPT
iptables -A WEB -p tcp --sport=80 -j ACCEPT

iptables -A WEB -p tcp --dport=443 -j ACCEPT
iptables -A WEB -p tcp --sport=443 -j ACCEPT
```

C Rules for HTTP/HTTPS/FTP(FTP Chain)

```
iptables -A FTP -p tcp --dport=80 -j ACCEPT
iptables -A FTP -p tcp --sport=80 -j ACCEPT

iptables -A FTP -p tcp --dport=443 -j ACCEPT
iptables -A FTP -p tcp --sport=443 -j ACCEPT

iptables -A FTP -p tcp --dport=21 -j ACCEPT
iptables -A FTP -p tcp --sport=21 -j ACCEPT
```

Machines B (Carriage) & F (Saddle)

1. *Allow inbound http and https requests from any source IP.*

HTTP

```
# iptables -A INPUT -p tcp --dport=80 -j ACCEPT
# iptables -A OUTPUT -p tcp --sport=80 -j ACCEPT
```

HTTPS

```
# iptables -A INPUT -p tcp --dport=443 -j ACCEPT
# iptables -A OUTPUT -p tcp --sport=443 -j ACCEPT
```

Machine C/Platen

1. *Unlike the other machines, the default outbound policy for Machine C should be deny.*

```
# iptables OUTPUT -P DROP
```

2. *Allow ftp connections only from 100.64.0.0/16.*

```
# iptables -A INPUT -p tcp -s 100.64.0.0/16 --dport=21 -j ACCEPT

# iptables -A OUTPUT -p tcp -d 100.64.0.0/16 --dport=21 -j ACCEPT
```

3. *Allow dns requests to 100.64.N.4 (chase).*

```
# iptables -A OUTPUT -p udp -d 100.64.5.4 --dport=53 -j ACCEPT
```

4. *Allow outbound ftp, http, https, and ssh connections to any host.*

Note: Two-way connections so allowing in and out

```
# iptables -A OUTPUT -p tcp --sport=80 -j ACCEPT

# iptables -A INPUT -p tcp --dport=80 -j ACCEPT


# iptables -A OUTPUT -p tcp --sport=443 -j ACCEPT

# iptables -A INPUT -p tcp --dport=443 -j ACCEPT


# iptables -A OUTPUT -p tcp --sport=21 -j ACCEPT

# iptables -A INPUT -p tcp --dport=21 -j ACCEPT


# iptables -A OUTPUT -p tcp --sport=22 -j ACCEPT
# iptables -A INPUT -p tcp --dport=22 -j ACCEPT
```

5. *Allow outbound icmp traffic only for icmp-types echo-request, echo-reply (ping), time-exceeded (traceroute), or destination-unreachable.*

Outbound version of what we did for all machine in the first section

```
# iptables -A OUTPUT -p icmp --icmp-type echo-request -j ACCEPT

# iptables -A OUTPUT -p icmp --icmp-type echo-reply -j ACCEPT

# iptables -A OUTPUT -p icmp --icmp-type time-exceeded -j ACCEPT

# iptables -A OUTPUT -p icmp --icmp-type destination-unreachable -j ACCEPT
```

Note: default output policy and default input policy should both be DROP, on Machine C only, everywhere else it's only default DROP for input policy

```
# iptables -P INPUT DROP

# service iptables save
```

Machine D/Chase

1. *Allow DNS queries from any source.*

Allow DNS Requests(For both tcp and udp)

```
# iptables -A INPUT -p udp --dport=53 -j ACCEPT
# iptables -A OUTPUT -p udp --sport=53 -j ACCEPT
# iptables -A INPUT -p tcp --dport=53 -j ACCEPT
# iptables -A OUTPUT -p tcp --sport=53 -j ACCEPT

# service iptables save
```

Machine E/Roller

1. *Restrict connections to the file sharing services (CIFS and SMB) from the 10.21.32.0/24 network only. CIFS and SMB use port numbers: 135/tcp, 137-139/udp, and 445/tcp.*
2. *Allow SSH connections only from hosts in the 10.21.32.0/24 subnet.*

Only allow (port 135/tcp, 137-139/udp, 445/tcp) on 10.21.32.0/24

```
# iptables -A INPUT -p tcp -s 10.21.32.0/24 --dport=135 -j ACCEPT
# iptables -A INPUT -p tcp -s 10.21.32.0/24 --dport=445 -j ACCEPT
# iptables -A INPUT -p udp -s 10.21.32.0/24 --dport=137:139 -j ACCEPT
# iptables -A OUTPUT -p tcp -d 10.21.32.0/24 --sport=135 -j ACCEPT
# iptables -A OUTPUT -p tcp -d 10.21.32.0/24 --sport=445 -j ACCEPT
# iptables -A OUTPUT -p udp -d 10.21.32.0/24 --sport=137:139 -j ACCEPT

# service iptables save
```