Lab 7 - Firewalling and IPTables

Start Assignment

Due Sunday by 11:59pm **Points** 100 **Submitting** a file upload

D-M corporate headquarters recently sent a memo banning all employees from visiting Facebook. Michael Scott has asked you to explicitly prevent all computers at our branch from even being able to talk with Facebook machines. Also, someone recently uploaded an incredibly hilarious picture of him on cheezburger.com so he wants it blocked too.

You decide to use iptables to drop this traffic. This situation has also brought to light the need to be more restrictive about what traffic you allow in and out of the production network.

With some help from a veteran system administrator you've developed the following rules for the hosts at your branch:

All Machines

1. Allow all traffic to and from the local loopback adapter lo, so each machine can talk to itself.
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.)
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).
4. All machines should implement a default deny policy for inbound traffic.
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.

6. 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'
```

7. 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.

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).

Machines B (Carriage) & F (Saddle)

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

Machine C/Platen

1. Unlike the other machines, the default outbound policy for Machine C should be deny.
2. Allow ftp connections only from 100.64.0.0/16.
3. Allow dns requests to 100.64.N.4 (chase).
4. Allow outbound ftp, http, https, and ssh connections to any host.
5. Allow outbound icmp traffic only for icmp-types echo-request, echo-reply (ping), time-exceeded (traceroute), or destination-unreachable.

Machine D/Chase

1. Allow DNS queries from any source.

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.

Rationale

It seems reasonable to prevent our machines from talking with facebook.com using iptables but this approach has significant disadvantages. First, Facebook's IP addresses may change so the firewall rules will need significant maintenance. Second, employees can simply proxy or tunnel traffic through one of the many sites that offer such services.

Ultimately, the best way to block facebook is to use something known as an http application firewall which can inspect the http traffic and ignore requests that include the URL of facebook.com, or common content exchanged. Even then, technically savvy employees can circumvent this by using a VPN or encrypted proxy.

Prerequisites

Please finish the reading before starting the lab. You are strongly advised to familiarize yourself with the procedures and commands involved on your own virtual machine before trying this on the production machines.

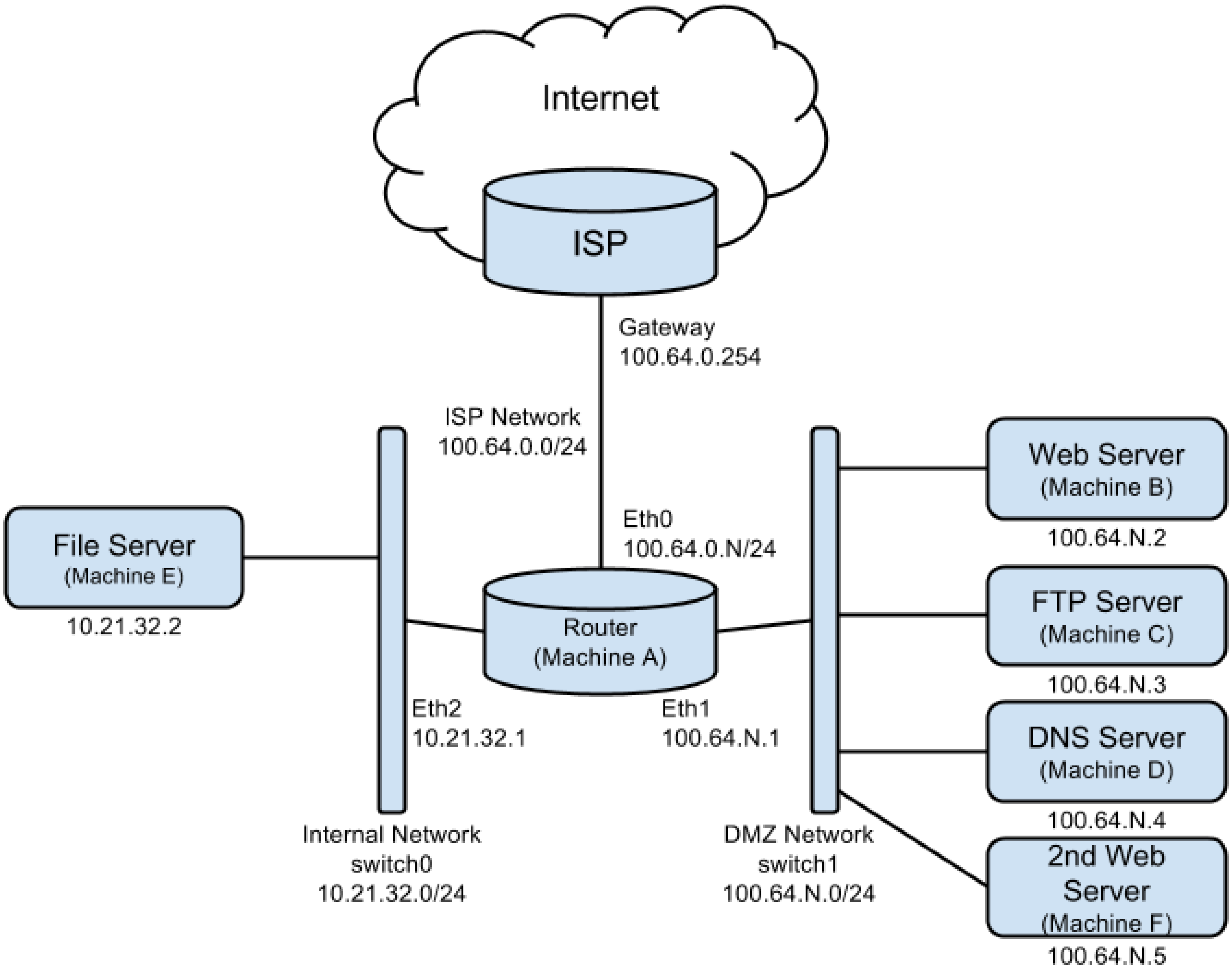
The netcat, tcpdump, and nmap packages are not installed in your production network. Install them as follows:

```
root@machineX# yum install nc tcpdump nmap ...
```

Dunder Mifflin Network Topology & Configuration

Dunder Mifflin currently has the following CentOS Linux machines:

1. The router does dhcp, ip_forwarding, and network/port address translation (nat/pat).
2. The http server, carriage, hosts the corporate websites.
3. The ftp server, platen, updates prices and accepts batch orders.
4. The dns server, chase, maps between domain names and IP addresses.
5. The file server, roller, stores company files centrally.
6. The secondary http server, saddle, makes Michael Scott happy.



Submission Requirements

1. Please submit your lab notes in as either a word, PDF or text document. Minimally they should contain the commands and procedures to configure iptables on all D-M production machines. You may use the script program to record commands in a text file.
2. The active iptables rules for each machine must match the overview above. They must be configured to be applied at boot time.
3. All services on the machines should remain accessible from the specified locations when the lab is finished.

Hints & Troubleshooting

- iptables -vL shows the number of packets each rule matches. This can be used for troubleshooting.
- To log messages, use the iptables logging module. It can be enabled with the -m flag. For example:
  root@machineX# iptables -A INPUT -j LOG --log-prefix "IPTABLES-DROP: "
- For services you cannot test directly, try using the wget, ping, nmap, tcpdump and netcat (nc) programs to verify that your firewall rules behave as expected.