MIECT: Advanced Network Security 2018-19

Practical Exercise: Linux firewalls with iptables

Due date: no date V 1.0

Contents

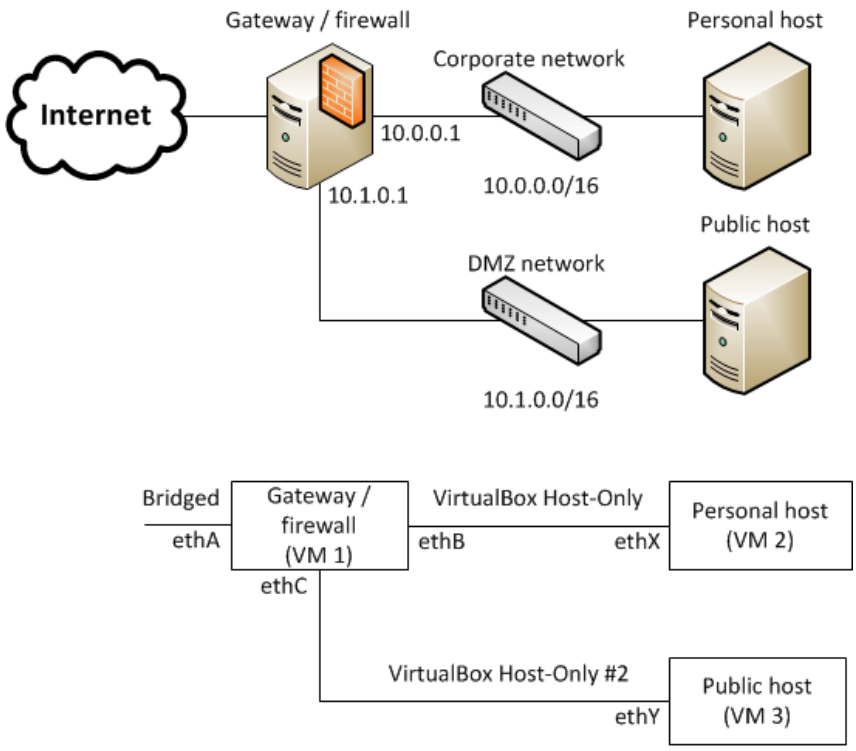

Figure 1: Conceptual network setting (top) and effective deployment using virtual machines (below)

1 Introduction

The goal of this work is to explore the functionalities of `iptables`, the Linux firewall, in a simple network setting. To facilitate the deployment of the network setting out of the laboratory, we will use only Linux virtual machines for implementing it. In this guide we will consider only the exploitation of `VirtualBox`.

2 Network setting

We will use the network setting of Figure 1. In this setting we have a gateway connected to the Internet and to two local networks, labelled DMZ and Corporate.

The DMZ and Corporate networks should be two host-only `VirtualBox` networks. These networks should be created with the network configuration interface of `VirtualBox` (accessible at File / Preferences / Network). As `VirtualBox` already features by default one network of this kind (`VirtualBox Host-Only Ethernet Adapter`), only another one needs to be created. For each host-only network disable the respective DHCP service provided by `VirtualBox`; the addresses referred in Figure 1 will be set manually.

Hereafter, for simplicity, we will refer each involved machine as VM 1, 2 and 3, according to Figure 1. The interfaces of the 3 virtual machines will also be named as A, B, C, X and Y, also according to the same figure.

3 Virtual machines

In this network setting we will use a Linux live distribution for all hosts. In this guide we will assume the Mint live distribution.

Create the virtual machine for VM 1 and clone it to create the other two. Don't forget to remove the useless network interfaces, leaving only one.

For reducing the workload while execution all 3 virtual machines simultaneously, use only, if possible, their console interface. For shutting down the graphical interface stop the graphical window manager service. In Mint we do so with the following command:

commentstyle
```
service mdm stop
```

Use the `ifconfig` command to manually set the interfaces of all virtual machines (except `ethA`) according to Figure 1. For `ethA` either configure it manually or launch a DHCP client, if enabled in the network that you are using as Internet:

4 Firewall configuration

4.1 Enable IP routing

By default Linux systems do not route IP packets; such feature has to be turned on. To turn it on you should execute the following command:

commentstyle
```
echo 1 > /proc/sys/net/ipv4/ip_forward
```

This command instructs the kernel that it should consider routing IP packets once they arrive to the machine and are not targeted to it. Execute this command on the gateway (VM 1).

Add a route for enabling inbound traffic from the internal networks to flow to the outside interface (referred as `ethA` in Figure 1):

commentstyle
```
ip route add default via [ethA IP address]
```

4.2 Default firewall policies

For each `iptables` chain we should define a default policy, which is nothing more than the default rule that applies to traffic not matched by any other rule. The advised approach is to follow a defensive policy: by default, deny when there may exist some risk of abuse. To do so, we should execute the following commands on the gateway (VM 1):

commentstyle
```
iptables -P INPUT DROP
iptables -P OUTPUT ACCEPT
iptables -P FORWARD DROP
```

Check the default policies with the following command:

commentstyle
```
iptables -n -L
```

The `-n` option should be always used along this guide when listing `iptables`' rules for avoiding useless (and time-consuming) reverse DNS translations.

4.3 Basic networking troubleshooting

Just for the sake of testing and troubleshooting, allow VM 1 network neighbors to ping it, and vice-versa:

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

4.4 Enable Corporate network traffic to the Internet

The Corporate network uses private addresses from network 10.0/16. Therefore, for enabling their computers to reach Internet services, one needs to authorize these addresses to be forwarded and enable NAT for all the packets arriving from this network. The NAT functionality must be defined in the `nat` table of POSTROUTING chain.

commentstyle
```
iptables -A FORWARD -i ethB -s 10.0/16 -j ACCEPT
iptables -A FORWARD -i ethA -d 10.0/16 -j ACCEPT
iptables -t nat -A POSTROUTING -s 10.0/16 -o ethA -j MASQUERADE
```

Remember that we, by default, denied forwarding as a general policy. Therefore, we need to authorize the traffic from and to the Corporate network with two different rules.

At this time, the host VM 2 should be able to reach Internet services, but some problems still exist with DNS name resolution, a fundamental requirement of nowadays Internet usage.

4.5 Resolution of DNS names

In this network setting we will configure all hosts in the Corporate and DMZ networks to use the gateway (VM 1) to provide DNS name resolution. For this to take place we need to edit manually the files `/etc/resolv.conf` in both VM 2 and VM 3. In those files, add and entry with the following content:

commentstyle
```
nameserver address
```

where `address` should be the address of VM 1 (10.0.0.1 for VM 2 and 10.1.0.1 for VM 3).

Now, in VM 1 we will not have any DNS server; instead, we will redirect DNS requests to an external server. Check the DNS server indicated for interface `ethA` (lets assume it is 192.168.1.1) and add the following rule:

commentstyle
```
iptables -t nat -A PREROUTING -i ethB -p udp --dport 53 -j DNAT \
                                                --to-destination 192.168.1.1
iptables -t nat -A PREROUTING -i ethC -p udp --dport 53 -j DNAT \
                                                --to-destination 192.168.1.1
```

These rules make use of Destination NAT, which is similar to Source NAT (or masquerading) but works on the destination address components (IP address and transport ports). They will replace the destination IP address of DNS queries (which use UDP port 53) with the IP address of the DNS server used by our network setting, while setting a reverse rule for packets flowing in the reverse direction (i.e., if a packet from X:x to Y:53 is changed to reach destination Z:53, the response from Z:53 will be modified on the route back to look like coming from Y:53).

Note: DNAT is most of the time used to perform port forwarding at the entrance of a NAT-isolated network, to enable inbound traffic to reach a service behind the NAT. In this case we are not doing so.

4.6 Redirection of traffic from the Internet to a DMZ service

Traffic should only reach the DMZ if targeting a service running on VM 3. Otherwise, there is no reason for allowing traffic to flow into the DMZ.

In this network setting we will install an SSH daemon in VM 3. For doing so, install the package `ssh`. For fetching the package from the Internet create a new user-defined chain (DmzNet) that will then be added or removed from the FORWARDING chain on a needed basis:

commentstyle
```
iptables -N DmzNet
iptables -A DmzNet -i ethC -s 10.1/16 -j ACCEPT
iptables -A DmzNet -i ethA -d 10.1/16 -j ACCEPT
iptables -A FORWARD -j DmzNet
```

For enabling the host VM 3 to reach the Internet we only need to transform its outbound traffic leaving VM 1 through NAT. This is necessary to allow responses to such traffic to be routed back to the originating host (an application):

commentstyle
```
iptables -t nat -A POSTROUTING -s 10.1/16 -o ethA -j MASQUERADE
```

After installing and starting the SSH daemon in VM 3, delete the forwarding rule that provides Internet access to VM 3:

commentstyle
```
iptables -D FORWARD -j DmzNet
```

Note that we removed the actual usage of the chain `DmzNet` but not its definition. To delete the chain we would have first to flush it (with the `-F` option) and, once empty, to delete it (with the `-X` option):

commentstyle
```
iptables -F DmzNet
iptables -X DmzNet
```

Note: Do not delete this chain, in order to allow VM 3 to contact the Internet on special cases, such as for installing software on certain occasions, just by adding the above referred forwarding rule.

After removing the `DmzNet` chain from the forwarding rules, the DMZ loses all capacity to contact or be contacted from the Internet, However, we do need this last capability, otherwise the public servers in DMZ could not be accessed from the Internet. Consequently, create the following chain:

commentstyle
```
iptables -N DmzIn
iptables -A DmzIn -i ethA -d 10.1/16 -m state --state NEW,ESTABLISHED -j ACCEPT
iptables -A DmzIn -i ethC -s 10.1/16 -m state --state RELATED,ESTABLISHED -j ACCEPT
iptables -A FORWARD -j DmzIn
```

In these rules we used the `state` module, which enables `iptables` to take decisions based on the state of interactions. In our case, we want to enable inbound traffic that initiates or continues a dialog, and outbound traffic that complements the initiation or continues a dialog. With these rules, it is not possible to initiate a TCP connection from VM 3 to outside the DMZ, nor to initiate any UDP request-response dialog (other than the DNS previously configured).

Then create a mapping between an external IP address and port for the service. Assuming that the external address of VM 1 is 192.168.1.92 and that the address of VM 3 is 10.1.0.101, we must do the following redirection:

commentstyle
```
iptables -t nat -A PREROUTING -i ethA -d 192.168.1.92 \
                                -p tcp --dport 22 -j DNAT --to-destination 10.1.0.101
```

Once this done, we can use an SSH client to address VM 3 using the VM 1 external (public) address and port 22 (the default SSH port). If required, other TCP ports can be used to multiplex several similar services (e.g. SSH) to different hosts behind the firewall.

5 Saving and restoring `iptables`' rules

The `iptables`' rules installed dynamically can be saved using the command `iptables-save`. The output of this command can be stored in a file for latter use with the counterpart command, `iptables-restore`.

6 References

1. iptables, `http://en.wikipedia.org/wiki/Iptables`
2. "The netfilter.org "iptables" project", `http://www.netfilter.org/projects/iptables`
3. Osvaldo Santos, "Firewalls: Soluções práticas", FCA, 2011, ISBN 978-972-722-688-7