Data Network Security, Homework 1 Report

Dorjan Hitaj, Matricola 1740478

November 5, 2017

1 Abstract

A firewall is a system that protects itself and other hosts on a network from attackers on untrusted networks, such as the internet. The linux kernel has included several different firewall implementations over the years, such as IPfwadmin and IPchains. On the Linux kernels of the series 2.4 and above IPtables firewall is included. IPtables are more flexible and powerful than their predecessors. Frequently what has to be achieved in terms of security from the IPtables for a specific network configuration is expressed in natural language in what are called security policies. The scope of this homework is: Given a network configuration and a set of policies described in natural language translate them into IPtables rules to fullfill the policy requirements.

2 Introduction

An IPtables firewall is made of three different kinds of objects. Those are **tables**, **chains**, and **rules**. Each of the three tables contains two or three standard chains, and possibly other user defined custom chains. Each chain is composed of zero or more rules, which are applied to packets received by or sent out from the firewall itself. These rules determine what will happen to each specific packet. When a network packet is processed by a chain, each rule in the chain is executed in order. Every rule has a set of conditions that determine whether the rule matches or not, and an action is taken in case of a match. The actions that can be taken on a packet involve: *accept*, *drop*, *modify* or *continue execution*. If none of the chain rules matches the packet then in the end of the chain the default action is going to be applied for that packet. Keeping this short theory in mind lets head up to the problem description.

A company, named **Onemorecomp** has just set up its network infrastructure and needs to build layers of security amongst network regions that compose the whole architecture. These regions need to be secured, and to secure them the IPtables firewall needs to be used. The company has compilated a set of security policies for each of the regions of the network and these policies need to be written as IPtables rules.

In the following sections a brief infrastructure description will be shown followed by the policies and for each policy its IPtable rules in the affected firewalls will be shown and the motivation behind each of those rules will be explained.

3 Network Topology

Onemorecomp company has setup an IT system whose firewalls need to be configured. The architecture is a split DMZ architecture, with an external DMZ network which contains a number of servers that are to be accessible from outside the company and an internal DMZ network that should be accessible only from inside the company. The topology is composed also of a user network, which represents the company employees machines and a cluster of machines that is used for extensive computations. These 4 regions are divided between each other by means of firewalls. The firewalls are set up in this way:

- Boundary firewall which resides between Internet and the external DMZ
- Main firewall which resides beetween external and internal DMZ
- Internal firewall which resides between internal DMZ and the user network
- Cluster firewall which resides beetween external DMZ and the cluster network

In total there are 4 firewall machines to be set up.

4 Policies

The policies are to be satisfied by the responsible firewall machines out of the 4 total firewalls that are in our architecture. In the firewalls section I will mention the policiy rules that should be satisfied by each firewall in a short summary and then I will explain technically how they are done using IPtables. Some rules of the policy required in the assignment involve more than one firewall to be satisfied so in the sections below I will divide those requirements into specific requiremnts for each firewall and then proceed to the IPtables rules for the firewalls one by one. To not duplicate myself I have included a lot of comments also on the shell scripts to better explain the rules and why they are needed.

An assumption I made while defining the rules is that the firewalls are transparent machines, they do not do routing, they just check packets that flow through them. However the Cluster front-end is not transparent, because it is supposed to receive and send requests as it will be explained in details below.

4.1 Common firewall rules

In each of the IPtables rules for the firewalls apart from the ones mentioned that satisfy the policy, there is also the default policy which is same in all of the firewalls for each of the chains. So the default policy is set to **DROP** in **INPUT**, **FORWARD**, **OUTPUT** chains, meaning that any packet that does not satisfy any of the policy rules in the destined environment will be dropped. This should be present in any IPtables configuration of this kind in order to prevent open possibilites that malicious entities can exploit to compromise our network and our machines. Apart for that, every machine is allowed to perform loopback requests(ping the localhost). All the requests that are already **established** or **related** are therefore accepted in all of the IPtables chains.

4.2 Boundary Firewall

The Boundary Firewall should:

- Allow user lookups from the internet and those should be handled only from the external DNS

- External DNS should allow resolving names only for the systems residing in the external DMZ

- Zone transfers from outside world should not reach internal DNS

- Zone transfers from external world should be accepted from a set of trusted servers

- Allow HTTP queries from the internet to be handled only by the external web server

- Allow external systems send mail to the Mail server in the external DMZ

- Allow mails from the companys network toward external users

- Allow SSH from internet users to the cluster frontend

4.2.1 IPtables rules

All the requests done from the external world or the internal network will be handled by the **FORWARD** chain of the Boundary firewall since it is not providing any of those services, but is just there to protect. Before going to any of the policy rules I have included some rules to prevent IP spoofing. In the FORWARD chain of the boundary firewall I DROP any packet that comes to the external interface(eth0) and has as source ip one of the IPs of the companys network.

To allow user lookups from the internet I ACCEPT in the FORWARD chain requests that are using protocol **udp**, destination ip of the external DNS and destination port **53**. Before this rule, in order to satisfy the policy that the external DNS should resolve names only for servers residing in the external DMZ I have included two rules that check if the packet contains a certain string, in my case **fwcluster** and **ext**. In case it contains **fwcluster** I allow it to be resolved and in case it does not contain **ext** I drop it. The idea of this rule came while reading. [3]

To allow zone transfers only from the external DMZ I ACCEPT in the FORWARD chain the requests destined only to the external DNS in the port **53** and protocol **tcp**. Moreover in this rule I also require that the source IP, which is the IP of the server requesting the zone transfer, to be the IP of the trusted servers, which in this case, for testing purposes, I have set it to be the IP of the LX machine which belongs to the "internet". This rule implies that requests for zone transfer do not reach internal DNS because the default policy is set to DROP if no rule is matched.

To allow HTTP queries from the internet the rule in the firewall checks if the protocol used is

tcp, the destination is the **Web server** and the port is **80**. I have also added a rule to allow requests from the internal proxy to go out in the internet toward HTTP servers, to make possible that the users in the users network to reach HTTP servers out in the internet.

To allow external systems to send mail to the Mail server I allow in the Boundary firewall requests that have as destination the Mail server in the port **25** and are using protocol **tcp**.

Also to allow the mails from the company network to go toward mail servers in the internet I allow packets that generate from the internal network and are destined toward the internet in port **25** using **tcp** as protocol.

To allow external systems to connect via SSH to the cluster frontend I allow the requests that are destined to the cluster frontend IP on port **22** that are using the **tcp** protocol.

4.3 Cluster Frontend Firewall

The Cluster Frontend should:

- Allow cluster computers to perfrom lookups and those should be handled by the external DNS
- Allow SSH login from the users in the user network or the internet to the Cluster Frontend(not directly with the cluster machines)
- Allow cluster machines to send mails to the mail server in the external DMZ or mail servers in the internet
- The cluster frontend should be able to communicate with the cluster machines by using SSH
- The cluster frontend should test the integrity of the cluster network by using ICMP (ping)

4.3.1 IPtables rules

To allow cluster computers to perform lookups, I allow packets to go through in the firewall if they are destined to the external DNS, are using **udp** protocol and have destination port **53**. This is handled in the FORWARD chain of the IPtables because is a packet that does not have the firewall as destination nor is generated by the firewall.

To allow users perform SSH login in the Cluster frontend I allow requests using protocol **tcp** on port **22** that are destined to the Cluster Frontend, thus this is handled in the INPUT chain of the IPtables since the packets are destined for the firewall itself. Source IP is not checked here since the policy states that the company employees can connect to the cluster frontend even from outside.

To allow cluster machines send mails to the mail servers I allow packets to go through if they are using the protocol **tcp** and the port they are destined is **25**. I do not check for the destination IP here since they should send mails also to the internet outside, so basically the destination IP can be anything.

To allow the Cluster frontend to perform SSH login in the cluster machines I allow it in the OUTPUT chain, since this will be a packet that is generated by the Cluster itself, packets that are using protocol **tcp** on port **22**.

To allow the Cluster frontend to perform integrity check by using ICMP I allow in the OUTPUT chain packets that are echo-requests to go through and allow in the INPUT chain packets that are echo-replys.

4.4 Main Firewall

The Main firewall should:

- Allow HTTP queries from the internal DMZ to the external DMZ HTTP server
- Allow Mail server in external DMZ to forward mails to the Admin server in internal DMZ
- Allow users in the internal DMZ to send mails to users in the internet
- Allow users from internal DMZ to perform SSH login to the cluster frontend

4.4.1 IPtables rules

The HTTP queries originated by the users in the user network will be handled by the HTTP proxy in the internal DMZ. To make possible the users to access the content provided by the Web server that resides in the external DMZ the Main firewall should allow requests coming from the web proxy that are destined to the web server in the external DMZ or any HTTP server in the internet. This is handled by allowing in the IPtables rules of the Main firewall requests that come from the Web proxy and are destined to any HTTP server using the protocol **tcp** on port **80**.

To allow the mails to be sent to the users in the user network we should allow the Mail server to forward mails to the Admin server in the internal DMZ. To accomplish this in the Main firewall I allow packets that are coming in the interface facing toward the external DMZ which are using protocol **tcp** on port **25** to go through.

To allow users in the user network send mails to the internet the Admin server should be used as a proxy, which will contact the Mail server in the external DMZ or mail servers in the internet. So in IPtables rules I allow packets to go through if they are using protocol **tcp** and destined to port **25** of the Mail server.

To allow SSH login to the Cluster frontend I allow requests to go through if they are using **tcp** protocol on port **22** and are destined to the Cluster frontend.

4.5 Internal Firewall

The Internal firewall should:

- Allow user lookups from the user network and those lookups should be handled by internal DNS server
- Allow HTTP queries from the user network and those should be handled by the Proxy server in the internal DMZ
- Allow users in the user network to POP mails from the ADMIN server in the internal DMZ using TLS
- Allow users to send mails to the outer world by using the Admin server as a SMTP proxy
- Allow users in the user network to perform SSH connection to the Cluster frontend

4.5.1 IPtables rules

In a similar fashion as is done in the Boundary firewall explained above, in the Internal firewall we allow DNS lookups that are destined to the internal DNS and that are using protocol **udp** on port **53**.

To allow users in the user network to perform HTTP queries they should use the Web proxy so to do this I allow packets to go through from the user network if they are using the protocol **tcp** on port **8080** destined to the Internal proxy server in the external DMZ.

To allow users to retreive the mail from the Admin server by using POP3S we should allow requests desitned to the Admin server using protocol **tcp** on port **995** which is the standard port for connecting using SSL.

To allow users in the user network to send mail in the outer world they should contact the Admin server and use it as a proxy, so in the Internal firewall rules I allow user requests on port **25** using protocol **tcp** destined for the Admin server.

Similarly as the rule above in the Main firewall, to allow SSH login to the Cluster frontend I allow requests to go through if they are using **tcp** protocol on port **22** and are destined to the Cluster frontend.

5 Conclusion

This homework was about defining a set of IPtables rules in 4 different firewall machines to satisfy a security policy that was provided in natural language. Most of the IPtables, but not all rules to be defined had to deal with the **FORWARD** chain of the IPtables because the machines that served as firewalls were not providing the requested service from various clients, they just had to check if they should allow the packets to go through them to their destined target machines.

6 Source files

The names of the files containg the shell script for each of the firewalls are:

- **boundary_firewall_rules.sh**
- **main_firewall_rules.sh**
- **internal_firewall_rules.sh**
- **cluster__firewall_rules.sh**

7 Refferences

1. IANA IANA port assignments
2. Frozentux IPtables tutorial
3. Spiceworks forum Filter DNS lookups
4. Blog Testing a POP3 server SSL
5. Lecture slides
6. Geekstuff IPtables fundamentals
7. Geekstuff IPtables rule examples
8. Digitalocean IPtables essentials and examples