INFO0045: Introduction to Computer Security
Firewalls

B. Donnet, Y. Vanaubel
Université de Liège

1 Overview

In this assignment, you will learn how to configure firewalls and NATs for a large network. This network will be emulated in a virtual environment with *netkit*. In this environment, you will use `iptables` to implement the different rules of the firewalls.

2 Network Description

A well-known multinational company, called PEAR2PEAR™, specialized in high-tech products, has decided to extend its business in Belgium. A whole new research department has been built in Liège a few month ago. Recently, a new network was deployed in the building. Knowing the excellent reputation of the Computer Scientists and Engineers from the University of Liège, the CEO, Steve ROBS, asked you to secure their network by configuring their different firewalls.

The gateway of the company's new network implements a first stateful firewall *FW1* (with interfaces `172.14.4.100/172.14.5.1/172.14.6.1`). *FW1* is connected to the Internet via its interface `172.14.4.100`. This gateway is connected to two other stateful firewalls *FW2* (`172.14.5.2/172.16.5.1/172.16.6.1`) and *FW3* (`172.14.6.2/172.14.7.1/172.16.4.1`). Finally, a last stateful firewall *FW4* (`172.16.7.1/172.16.8.1/172.16.9.1`) delimits, with *FW2*, a sandwich demilitarized zone (DMZ).

In this DMZ are located four different devices: a `DHCP` server (*DHCP*, `172.16.6.2/172.16.7.2`), an `SSH` relay (*SSH*, `172.16.6.3/172.16.7.3`), an `HTTP(S)` proxy[1] (*HTTP*, `172.16.6.4/172.16.7.4`), and a local `DNS` server (*LDNS*, `172.16.6.5/172.16.7.5`).

Two desktop computers, *I1* (`172.16.8.3`) and *I2* (`172.16.4.2`) are available for the employees. These computers have a static IP address, and therefore, they do not communicate with the `DHCP` server.

The DHCP server is used to assign IP addresses to the different laptops that can connect to the networks `172.16.5.0/24` and `172.16.8.0/24`. Two `DHCP` relays *DHCP_R1* (`172.16.8.2`) and *DHCP_R2* (`172.16.5.2`) were deployed on each network to forward `DHCP` requests to the `DHCP` server.

An `RSYNC` server (*RSYNC*, `172.16.9.2`) is used by the employees to backup their documents. The data is sensitive, and must be transferred securely to the server (meaning, the communication must be be encrypted). Only computers connected to the network `172.16.8.0/24` are allowed to transfer their data without encryption[2].

The two desktop computers *I1* and *I2* may be controlled remotely via `SSH` connections. If connections are established with *I1* from outside the network `172.16.8.0/24`, the `SSH` relay must be used. This relay is also needed if data is transferred to *RSYNC* from outside `172.16.8.0/24`. Also note that *I1*, *I2*, and laptops are allowed to establish `SSH` connections with devices in the Internet.

None of the computers (desktops and laptops) of PEAR2PEAR™ is allowed to send directly `HTTP(S)` requests to the Internet. They must use the `HTTP(S)` proxy. This proxy is the only device for which `HTTP(S)` requests addressed to a server outside the network will be accepted.

The local `DNS` server is used by the devices connected to networks `172.16.5.0/24` to `172.16.8.0/24`. Another `DNS` server, a public one (*PDNS*, `172.14.5.3`), is available for other requests (from the Internet or other devices from the network).

[1] This proxy works with both `HTTP` and `HTTPS`.
[2] It does not mean they can not use a secure transfer.

The company also owns its own Web server (*web*, `172.14.6.3`). The device hosting the Web server (meaning *web*) also runs an `FTP` server. This `FTP` server is used to update the website of the company.

Finally, a last device, named *SMTP*, is available at the IP address `172.14.7.2`. This device hosts three servers: an `SMTP` relay, a `POP` server, and a `TELNET` server. The different users can solicit the mail server to receive or send their emails (from inside and outside the company network). Note that only the computers on network `172.16.8.0/24` are not allowed to use the mail services due to security reasons.

The employees are allowed to home work, and so, *RSYNC*, *I1* and *I2* must be reachable from the Internet.

3 Laptops

As said in Sec. 2, laptops are allowed to connect to networks `172.16.5.0/24` and `172.16.8.0/24`. In this assignment, we will assume the devices *U1* and *U2* are two laptops connected respectively to `172.16.8.0/24` and `172.16.5.0/24`.

4 Accounts

Steve Robs and Tim Crook work for the company. They have an account on most of the devices in the network. Yo have access to their accounts to test your security system deployment. Their login are `tim` and `steve`. The passwords are identical to their login, for the sake of simplicity. By default, you are logged as `root` (password: `root`) on each *netkit* device. If you want to change user, you must use the command `su steve` or `su tim`.

Steve and Tim also have their own mail addresses in the company:

`{steve|tim}@pear2pear.apl`

5 Additional Information

Here, we provide you additional information on how to manage the various services run by Pear2Pear™ on a Netkit virtual environment.

5.1 Private IP Addresses

Netkit is run behind a NAT. Each time a packet is sent to the Internet, its source IP address is replaced by the IP address of your computer. So, even if the company's network is connected to the Internet, you cannot start communicating with it from the outside of the virtual environnement. Then, in order to allow you to realize other tests, you have access to the personal computer of Tim, *T* (`172.14.3.10`)[3]. This device is configured to use `SMTP` as relay for the emails, and `PDNS` as `DNS` server.

You may observe public IP addresses in this assignment. Even if these IP addresses may have been assigned somewhere else in the Internet, the elements in Netkit are configured to send their packets to the devices inside the virtual environnement. So, you can consider these addresses as *real public IP addresses*.

5.2 `DNS`

The `DNS` servers are already configured and contain the following entries:

- `www.pear2pear.apl`: address of the web server
- `mail.pear2pear.apl`: address of the `SMTP`/`POP` server
- `ssh.pear2pear.apl`: address of the `SSH` relay
- `ssh2.pear2pear.apl`: address of *I2*

[3]In the virtual environment, the device named *router* is used to forward packets between the Internet, the company's network and the computer of Tim. You can consider this device as invisible

- `rsync.pear2pear.apl`: address of the `RSYNC` server

The local `DNS` server also contains several entries mapping some machine names to their IP addresses. Only machines with a unique interface are considered.

5.3 Mail Service

Steve and Tim can send and receive emails with their PEAR2PEAR™ addresses. The different user machines in the virtual environment (*I1, I2, U1, U2, and T*) implement a mail client, named **mutt**. The client is configured to use the `SMTP` device of the company as `SMTP` relay and `POP` server.

In order to use **mutt** from a device, you can simply type `mutt` in the terminal of the virtual machine. The client is configured to use the mail address of Steve (or Tim), depending on which account you are logged.

Remark: By default, the `SMTP` device in the network uses the `SMTP` server of the university as `SMTP` relay (`smtp.ulg.ac.be`). When you are connected to the ULg network, you can send emails from a PEAR2PEAR™ address to another domain (GMAIL, ULG, ...). This feature does not hold anymore when you are outside the university. Indeed, the SEGI's relay will not accept to forward your emails. Note also that it is useless to try to send an email from the Internet to a PEAR2PEAR™ address, because this domain only exists in the virtual environment.

5.4 Web Browser

A web browser (*lynx*[4]) is available on each machine in *netkit*. To request a web page, simply type

```
lynx http://website.domain
```

If you need a secure transfer and want to use `https`, replace `http` by `https` in the command.
In the virtual environment, *lynx* is configured to use automatically the `HTTP(S)` proxy when needed.

5.5 FTP

`FTP` (File Transfer Protocol[5]) is a protocol that allows the exchange of files on a computer network. In this assignment, you can type the following command to connect to the `FTP` server:

```
ftp server
```

where `server` is the address of the `FTP` server. Once connected, you are asked to specify a username and a password (`root` is not allowed). After the authentication, you obtain a prompt. You can then used the following commands:

- `?`: get a list of the available commands
- `ls`: list the content of the current directory on the server
- `!ls`: list the content of the current directory on the client
- `cd`: change the current directory on the server
- `lcd`: change the current directory on the client
- `mkdir`: create a directory on the server
- `put`: send a file to the server
- `get`: get a file from the server

[4] See `http://lynx.browser.org`
[5] See RFC959 – `http://www.ietf.org/rfc/rfc959.txt`

5.6 SSH through a relay

When you need to establish an SSH connection with a device behind an SSH relay, you must create an SSH tunnel starting from the source to the destination, and going through the relay. This can be achieved by using the following command:

```
ssh -t user@relay ssh user@destination
```

where user is your username.

Note that each user can use a public-key authentication. This allow them to not type any password to connect to the different SSH devices. The username can also be removed in the command. To use this public-key authentication, you must be logged as the user on the computer establishing the connection.

5.7 RSYNC

RSYNC[6] is a software used to synchronize files. In the PEAR2PEAR™ company, it is used to implement a remote backup system.

In order to synchronize a file with the server, you can use the following command:

```
rsync -v file user@server::module
```

where

- file is the file you want to synchronize
- user is your username (root is not allowed)
- server is the address of the RSYNC server
- module is the name of a module. A module gathers a set of information for RSYNC (where the synchronized files are stored on the server, the user that may send data, etc). Two modules are defined on the server. For Tim, use the module *backup_tim*. For Steve, use *backup_steve*. The files are synchronized in the home directory of each user on the server (Tim and Steve have an account on RSYNC).

During the transfer, the data is not encrypted. It means anyone could read the documents being sent. So, if important information must be synchronized, you need to use RSYNC with SSH. To do so, the command to type becomes:

```
rsync -v file user@server:destination_directory
```

where destination_directory is the directory on the RSYNC server where the file must be synchronized. The path of this directory can be absolute, or relative to the home directory of the user.

Sometimes, the only way to reach RSYNC is by establishing a connection through the SSH relay (i.e. creating a tunnel). In this situation, the command[7] is:

```
rsync -av -e "ssh -A user@relay ssh" file user@server:destination_directory
```

where relay is the address of the SSH relay. Again, if you want to use the public-key authentication, you must be logged as user.

5.8 TELNET

TELNET[8] is a network protocol that was created to provide a fairly general, bi-directional, byte oriented communications facility. In the network company, TELNET is used to test the SMTP and POP servers. To use it, simply type the command:

```
telnet server port
```

[6] See http://rsync.samba.org

[7] Before using this command for the first time, you should have connected to *RSYNC* through the relay with SSH in order to add *RSYNC* in the known hosts list of *SSH*.

[8] See https://tools.ietf.org/html/rfc854

where `server` and `port` are respectively the address and the port of the server to which a connection must be established. You can use `TELNET`, for example, to ask the `SMTP` server to send emails, or check emails stored by the `POP` server. Commands needed to interact with the servers can be easily found on the Internet.

6 Assignment Rules

The submission of your solution will be done in two steps: (*i*) you have to draw the network of PEAR2PEAR™ (Sec. 6.1) and (*ii*), you have to write high-level firewalls rules, implement them using `iptables`, and use the virtual machines provided to test your implementation (Sec. 6.2). In this second step, you are also asked to discuss the architecture of the company's network (pros and cons of the structure, and how to improve it).

6.1 Step 1: Drawing the Network

6.1.1 Purpose

The very first step of your assignment is to draw the network infrastructure described in Sec. 2. In this drawing, you have to notify IP addresses, possible DMZ, firewalls, NAT interfaces, proxies, servers, end-hosts, etc.

6.1.2 Agenda

A report containing your drawing and an explanation of the drawing is due to **April, 3rd, 08:00 AM**.

Later on April, 3rd, the correct network drawing will be provided on the course website. For the second step (firewall rules and discussion– See Sec. 6.2), you will have to work based on the provided network drawing. In addition, a set of scripts and virtual machines based on Netkit implementing the PEAR2PEAR™ network will be provided. You will use them to test your `iptables` rules.

6.1.3 Submission

The submission of the first part of your assignment is subject to the following rules:

1. you must give back a `PDF` file named as followed: `Group-XX.pdf`, where `XX` refers to your group ID.
2. your PDF file will include the following items:
   - a drawing representing the network to defend;
   - an explanation of the drawing.
3. your PDF file must be uploaded on the submission platform (see `http://submit.run.montefiore.ulg.ac.be`)
4. the deadline is **April 3rd, 2015, 08:00AM**. The deadline is strict. The first part of the assignment cannot be uploaded after the due date. This means that, after the due date, assignments not uploaded will receive a zero.

6.1.4 Gradings

The first part of the firewall assignment will count for 20% of the final assignment grade.

6.2 Step 2: Firewall Rules Implementation

On April, 3rd, on the course web site, the correct schema of the PEAR2PEAR™ network will be provided to you. In addition, we will provide you a bunch of scripts and virtual machines (based on Netkit) that implement this network. Please, use those scripts and virtual machines to test your firewall rules.

6.2.1 Purpose

The objective of the second step is to provide high-level firewalls rules (as done during the course and the exercises sessions). It is also asked to translate those rules into `iptables` rules.

6.2.2 Netkit

You may install Netkit on your personal computer. The instructions for the installation are available on the Netkit website[9]. Note that Netkit only runs under Linux systems.

If you are running Windows or Mac OSX, you may emulate Linux in a virtual environment (VirtualBox, for example). Note however that Netkit also creates a virtual environment. You may then not be able to emulate the network if your machine is not enough powerful.

6.2.3 Implementation

Your Iptables rules must be written in four different files (one per firewall). These files are located in `path_to_lab/FWx/root/config_FWx.sh` where `x` is 1, 2, 3 or 4 depending on the firewall you want to configure. These files are automatically run each time the devices are turned on.

6.2.4 Agenda

A report containing your discussion about the architecture of the network, your high level firewalls rules (as well as their description) and several files containing your `iptables` rules are due to **April, 27th, 08:00 AM**.

6.2.5 Submission

The submission of the second part of your assignment is subject to the following rules:

1. you must upload on the `http://submit.run.montefiore.ulg.ac.be` platform an archive, named `Group-XX.tar.gz` where `XX` refers to your group ID. This archive must contain a report (presented as a `PDF` file) and three configuration files.
2. your `PDF` file must be named as followed: `Group-XX.pdf`, where `XX` refers to your group ID.
3. your PDF file will include the following items:
   - the high level firewalls rules
   - any explanation of those rules.
4. three configuration files named `config_FWx.sh` containing the `iptables` implementation of firewall `x` (where $\texttt{x} \in [1;4]$).
5. the deadline is **April 27th, 2015, 08:00AM**. The deadline is strict. The first part of the assignment cannot be uploaded after the due date. This means that, after the due date, assignments not uploaded will receive a zero.

6.2.6 Gradings

The second part of the firewall assignment will count for 80% of the final assignment grade.

[9] `http://www.netkit.org`