INFO0045: Introduction to Computer Security
Assignment - Firewalls

B. Donnet, J. Iurman
Université de Liège

1 Overview

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

2 Context

This is it... On November 3rd this year, the presidential elections will take place in the USA. This means that the candidates almost went through their whole campaign process.

In order to maximize his chance to stay the President of the United States, and in a final attempt to reverse the trend of votes, Donald has created a company called Fakenews^{TM}. The goal of this company is to discredit the fake news about him (well, also real news but he just wants to win...) started by the opposite side. Also, this will be a channel to in turn propagate fake news about Joe Biden, his main opponent. After all, Donald does not like fake (or real?) news about himself, but appreciates when it is about other people on his way. This, hopes Donald, will help him to go back-to-back, he who defeated the covid-19 without wearing a mask **demonic laugh**. He also knows he can still count on his Russian friends to manipulate the results once again, so he is confident.

You are a network security expert and have been hired, for a big salary, by Donald to reach his goal with the company. Another expert, G.I. Joe, has also been hired and will work with you remotely due to the covid-19. Is it secretly Joe Biden? Who knows... You are both asked to configure all firewalls so that everything is secured and works as expected. Hurry, time is almost up. If you fail, you will serve as a laboratory rat and receive a lot of disinfectant injections, just to prove his theory against the covid-19 is correct. You are warned, be careful !

3 Network Description

In the company's network, a gateway named *FW1* (with interfaces `172.15.29.130/172.15.30.1/172.16.31.1/172.16.32.1`) implements a **stateful** firewall. It is connected to the Internet via its interface `172.15.29.130`. This gateway is connected to another **stateful** firewall named *FW4* (`172.16.31.2/192.168.1.1`).

Two other **stateful** firewalls *FW2* (`10.10.3.1/10.10.4.1/192.168.2.1`) and *FW3* (`10.10.1.1/10.10.2.1/192.168.3.1`) are also part of the network.

FW1 is separated from *FW2* by following devices: an `HTTP(S)` proxy[1] (*HTTP*, `10.10.3.2/172.16.32.2`), a `DHCP` server (*DHCP*, `10.10.3.3/172.16.32.3`), a local `DNS` server (*LDNS* `10.10.3.4/172.16.32.4`) and a mail server (*MAIL*, `10.10.3.5/172.16.32.5`).

FW4 is separated from *FW3* by following devices: an `SSH` relay (*SSH*, `10.10.1.3/172.16.31.3`) and an `RSYNC` server (*RSYNC*, `10.10.1.4/172.16.31.4`).

The `DHCP` server is used to assign IP addresses to the different laptops that may connect to the `192.168.1.0/24` and `192.168.2.0/24` networks. Two `DHCP` relays, *DHCP_R1* (`192.168.1.2`) and

[1] This proxy works with both `HTTP` and `HTTPS`.

DHCP_R2 (192.168.2.2), have been deployed in these subnets to forward DHCP requests to the DHCP server.

Two DNS servers are available in the company's network. The local DNS server is used by the devices connected to the 192.168.1.0/24 and 192.168.2.0/24 networks. The other DNS server, the public one (*PDNS*, 172.15.30.3), is available for requests from the Internet. They must be able to forward a request to another dns server on the Internet, in case they do not know a requested domain.

The mail server is composed of an SMTP relay and an IMAP server. The different users can then solicit the mail server to receive or send their emails (from inside the network for 192.168.1.0/24 and 192.168.2.0/24, and from outside the company's network). An authentication is required to download emails from the server. Moreover, the security plan requires the following rules:

- The SMTP relay listens only on port 25, and allows the client to encrypt the communication with TLS.

- To consult their emails, the different users must establish a connection with the IMAP server. If the connection stays inside the company's network, it must not be encrypted to allow deep packet inspection. However, any connection that crosses the Internet must be encrypted. In this case, IMAPS must be used.

The company has its own public web server (*PWEB*, 172.15.30.2). It is used by the different users having a FAKENEWS^{TM} account. Sensitive information may be sent to the server. That's why this web server accepts both HTTP and HTTPS connections.

Another web server (*LWEB*, 10.10.4.2) is available on the network. It is used only by computers connected to a part of the company's network, meaning devices on the 192.168.1.0/24 and 192.168.2.0/24 networks. This private web server hosts different internal information that must be communicated to the employees, such as schedules, internal news, ... This server accepts only HTTP connections, and also runs an FTP server. This FTP server is used to update the HTML pages of the internal website. Only devices connected to the network 192.168.2.0/24 are allowed to transfer data to the server with FTP.

The HTTP(S) must be used by any computer connected to the 192.168.1.0/24 and 192.168.2.0/24 networks, whatever the destination web server, even if located in the Internet. However, two exceptions are made:

- The devices connected to the network 192.168.1.0/24 are allowed to connect to the company's public web server (*PWEB*) without using the relay.

- The computers connected to 192.168.2.0/24 are allowed to connect to the local web server (*LWEB*) without using the relay.

A desktop computer (*U3*, 192.168.3.2) is also available for the employees. It is used for research purpose and can be accessed via SSH from the 192.168.1.0/24 network and from the Internet.

The SSH relay is used to forward SSH connections:

- from the Internet to *U3*, and the opposite direction too.

- from laptops in the 192.168.1.0/24 network to *U3*.

- from laptops in the 192.168.1.0/24 network to the Internet.

The RSYNC server is used by the employees to backup their documents. Only the devices connected to the 192.168.1.0/24 and 192.168.3.0/24 networks are allowed to transmit documents to the server. Backups from the Internet are then prohibited. The data can be transferred securely (meaning, the communication must be encrypted), or without any encryption.

Finally, a NFS server (*NFS*, 10.10.2.2) is used by *U3* to backup/synchronize some documents.

3.1 Laptops

As said in the previous description, laptops are allowed to connect to the 192.168.1.0/24 and 192.168.2.0/24 networks. In this assignment, we will assume the devices *U1* and *U2* are two laptops connected respectively to 192.168.1.0/24 and 192.168.2.0/24.

3.2 Accounts

Joe and Donald have an account on most of the devices in the network. You have access to their accounts to test your security system deployment. Their login are `joe` and `donald`. 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 - login
```

where *login* is either `joe` or `donald`. To log off, just press `CTRL + D`.

On each user device, you will find different text files in the home directories of each account. These files can be used to test your configurations (`RSYNC`, `SSH`, `NFS` or `FTP` transfers, for example).

3.3 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 environment. Then, in order to allow you to realize other tests, you have access to the personal computer of Donald, *DT* (`172.15.28.2`)[2]. This computer is configured to use the mail server, and `PDNS` as `DNS` server. It also runs an `SSH` server, allowing Donald to connect to *DT* from the company's network, when allowed.

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.*

3.4 SSH

When you need to establish an `SSH` connection with a device behind an `SSH` relay, you must create an `SSH` tunnel 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, `relay` is the address of the `SSH` relay, and `destination`, the address of the destination device.

Note that each user can use a public-key authentication. This allows them to not type any password to connect to the different `SSH` devices. The username can then 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.

In order to simplify the connection process, *U1* and *U3* have been configured to automatically use the proxy when needed, *if the name of the device is used as destination.* With this method, you can omit the relay in the `SSH` command. As an example, if you need to establish a connection from *U3* to *DT*, simply log you in *U3* as Donald, and type:

```
ssh DT
```

This method will work only if the source is intended to communicate with the destination. So the previous command will not work if you are logged as Joe, or if the destination is *PWEB*, for example. Note that if the method is used correctly, the device's names do not require any `DNS` resolution. However, if you want to access an external ssh server (anywhere on the Internet) by hostname or ip, you will have to use the tunnel command (be careful with hostnames, they require a dns resolution).

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

3.5 NFS

NFS (Network File System) is a protocol used to share data between devices on a network. The interest of this system is that you can use a directory (even the entire file system) of a remote computer as if it was a hard drive directly connected to your computer. In the FAKENEWS^{TM} network, the directory */home/sharing* on the `NFS` server is shared with *U3*. Each time this computer is turned on, it mounts automatically the shared directory. In other words, this operation tells the file system of the clients that the directory on the `NFS` server can be used as a hard drive. On *U3*, the directory is also mount at */home/sharing*, for the sake of simplicity. Once *U3* modifies some data in the directory, the modifications are sent to the server. The server notifies then the other clients (if any) that the directory was modified. In addition of predefined/fixed `NFS` ports, you will need to allow ports 2046, 2047 and 2048 respectively for *status*, *nlockmgr* and *mountd*.

3.6 Remote Synchronisation (RSYNC)

`RSYNC`[3] is a software used to synchronize files. In the FAKENEWS™ 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 JOE, use the module *backup_joe*. For DONALD, use *backup_ donald*. The files are synchronized in the home directory of each user on the server (JOE and DONALD 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. Again, the remarks done in Section 3.4 apply (public-key authentication and relay).

Note: For secured rsync, you must force the use of the `SSH` relay (no direct ssh connection, even when it is possible). For unsecured rsync, you need to use the IP address of `RSYNC` because `RSYNC` is only defined in ssh context.

3.7 Domain Name Servers (DNS)

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

- `www.fakenews.tweet`: address of the public web server
- `local.fakenews.tweet`: address of the local web server
- `mail.fakenews.tweet`: address of the mail server

[3] See `http://rsync.samba.org`

3.8 Mail Service

Joe and Donald can send and receive emails with their Fakenews^{TM} addresses:

```
{joe|donald}@fakenews.tweet
```

The different user machines in the virtual environment (*U1, U2*, and *DT*) implement a mail client, named `mutt`. The client is configured to use the mail server of the company as `SMTP` relay and `IMAP` server. It will use the mail address of Joe or Donald, depending on which account you are logged in. Note also that `mutt` knows when it must use `IMAP` or `IMAPS` to respect the security plan.

To use `mutt` from a device, log as Joe or Donald, and type `mutt` in the terminal of the virtual machine.

Remark: By default, the mail 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 Fakenews^{TM} 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 Fakenews^{TM} address, because this domain only exists in the virtual environment.

3.9 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 on each computer to use automatically the `HTTP(S)` proxy when needed.

3.10 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 use 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

3.11 HTTP proxy

The `HTTP(S)` proxy runs squid[6]. Be careful: the listening port is not the one you may think about intuitively. Note also that both `http` and `https` are handled by the same port.

[4] See `http://lynx.browser.org`

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

[6] `http://www.squid-cache.org`

4 Assignment Rules

The submission of your solution will be done in **three** steps: (*i*) you have to draw the network (Sec. 4.1), (*ii*) you have to write high-level firewalls rules, and NAT rules if any (Sec. 4.2), and (*iii*) you have to implement them using `iptables`, and use the virtual environment provided to test your implementation (Sec. 4.3). At the end, each group will be scheduled for a short demo.

4.1 Step 1: Drawing the Network

4.1.1 Purpose

The very first step of your assignment is to draw the network infrastructure described in Sec. 3. In this drawing, you have to notify IP addresses, possible (sandwich) DMZ, zones and firewalls, NAT interfaces, proxies, servers, end-hosts, etc.

In case of questions, use the eCampus forum (a forum has been created specifically for this assignment).

4.1.2 Agenda

A report containing your drawing, all zones and zones per firewall (in priority order), and an explanation, is due to **October 30$^{\text{th}}$, 2020, 08:00 AM**.

Later on October 30$^{\text{th}}$, the correct network drawing and zones will be provided on the course website. For the second step (high-level rules – See Sec. 4.2), you will have to work based on the provided network drawing.

4.1.3 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 protect, with delimited zones.
   - an explanation of the drawing, zones, and their order per firewall.
3. your PDF file must be uploaded on the submission platform (see `http://submit.montefiore.ulg.ac.be`)
4. the deadline is **October 30$^{\text{th}}$, 2020, 08:00 AM**. This 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 get a zero.

4.1.4 Gradings

The first step of the firewall assignment will count for 20% of this project.

4.2 Step 2: High-Level Rules

4.2.1 Purpose

The objective of the second step is to provide high-level firewalls (and NAT, if any) rules, as done during the course and exercise session.

In case of questions, use the eCampus forum (a forum has been created specifically for this assignment).

4.2.2 Agenda

A report containing your high level firewalls (and static NAT, if any) rules (as well as their description), is due to **November 13th, 2020, 08:00 AM**.

Later on November 13th, the correct high-level rules will be provided on the course website. For the third step (iptables rules – See Sec. 4.3), you will have to work based on the provided high-level rules, and implement them with iptables. In addition, a set of scripts and virtual machines based on *Netkit* implementing the network will be provided. You will use them to test your `iptables` rules.

4.2.3 Submission

The submission of the second 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:
   - the high level rules, structured as seen during the exercise session.
   - any explanation of those rules.
3. your PDF file must be uploaded on the submission platform (see `http://submit.montefiore.ulg.ac.be`)
4. the deadline is **November 13th, 2020, 08:00AM**. The deadline is strict. The second part of the assignment cannot be uploaded after the due date. This means that, after the due date, assignments not uploaded will get a zero.

4.2.4 Gradings

The second part of the firewall assignment will count for 40% of this project.

4.3 Step 3: iptables Rules

4.3.1 Purpose

The objective of the third and last step is to write `iptables` rules, based on high-level rules. You will test those rules in the provided virtual environment.

In case of questions, use the eCampus forum (a forum has been created specifically for this assignment).

4.3.2 Agenda

A short report as well as all firewall configuration files, are due to **November 20th, 2020, 08:00 AM**.

Each group will have to go through a live demo. During the demo, your implementation will be tested in the virtual environment. The results of this test phase will be taken into account in the final grade. Organizational details for the live demo will be provided on eCampus.

4.3.3 Netkit

Netkit is already installed on the virtual machine provided for this class. Note that *Netkit* only runs under Linux systems.

4.3.4 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 executed each time the virtual devices are created.

4.3.5 Submission

The submission of the third and last part of your assignment is subject to the following rules:

1. you must upload on the `http://submit.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 (`PDF` file) and all firewalls configuration files.

2. your `PDF` file must be named as followed: `Group-XX.pdf`, where `XX` refers to your group ID.

3. all firewalls configuration files named `config_FWx.sh` containing the `iptables` implementation of firewall `x` (where $\texttt{x} \in [1;4]$).

4. the deadline is **November 20th, 2020, 08:00AM**. The deadline is strict. The third part of the assignment cannot be uploaded after the due date. This means that, after the due date, assignments not uploaded will get a zero.

4.3.6 Gradings

The third part of the firewall assignment will count for 40% of this project.