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 firewalls rules.

2 Context

Months after his defeat in the last Presidential Election of the United States of America, Donald Trump is preparing his comeback in the dark. He has been banned from most of social platforms due to lots of provocations and aggressive speeches, but that won't stop him. His goal is, of course, to win the next Election and beat Joe Biden. To reach that goal, he asks for help to his old friend Vladimir Poutine. As a result, the Russian government immediately creates a new secret department called MDGA™ (`"Make Donald Great Again"`). `Donald` is given full remote access to the MDGA™ network, from his laptop.

But, what `Donald` ignores is that `Vlad` has secretly given the order to his military department to spy on him. While `Donald` thinks his friend is trustworthy, `Vlad` sees another opportunity to increase his power over America, thanks to his favorite puppet. `Donald` is supposed to have access to the entire MDGA™ network. However, a small part is kept secret by the Russian government. A honeypot with fake data is available, just to make `Donald` think everything is fine.

You are a network security expert and have been hired by `Vlad`, for a big salary (and for your silence, obviously). He asked you to configure all firewalls so that everything will happen as expected. If you fail, not only the scandal will spread but you may be severely punished by `Vlad`. In Russia, missing persons are quite common. You have been warned, be careful !

3 Network Description

In the MDGA™ network, a gateway named *FW1* (with interfaces `172.31.5.1/172.31.6.1/172.32.4.100/172.32.5.1`) implements a **stateful** firewall. It is connected to the Internet via its interface `172.32.4.100`.

Two other **stateful** firewalls *FW2* (`10.10.1.1/10.10.2.1/192.168.1.1/192.168.2.1`) and *FW3* (`10.10.3.1/10.10.4.1/192.168.3.1`) are also part of the network.

FW1 is separated from *FW2* by following devices: a `DHCP` server (*DHCP*, `10.10.1.2/172.31.5.2`), a local `DNS` server (*LDNS* `10.10.1.3/172.31.5.3`), an `HTTP(S)` proxy[1] (*HTTP*, `10.10.1.4/172.31.5.4`), a mail server (*MAIL*, `10.10.1.5/172.31.6.5`) and an ssh server (*SSH*, `10.10.1.6/10.10.4.6/172.31.6.6`).

FW3 is separated from both *FW1* and *FW2* by the ssh server (*SSH*).

The `DHCP` server is used to assign IP addresses to the different computers 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 *DHCP_R2* (`192.168.2.2`), have been deployed in these subnets to forward `DHCP` requests to the `DHCP` server.

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

Two DNS servers are available in the 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*, 172.32.5.3), 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 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 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 MDGA™ network has its own public web server (*PWEB*, 172.32.5.2). Sensitive information may be sent to the server. That's why this web server accepts both HTTP and HTTPS connections.

Another web server (*LWEB*, 10.10.2.2) is available in the network. It is used only by computers connected to a part of the 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 192.168.1.0/24 are allowed to transfer data to the server with FTP.

The HTTP(S) proxy 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. *LWEB* must be directly reachable from the 192.168.1.0/24 and 192.168.2.0/24 networks, without using the relay.

A server located inside the network and containing fake data (*HONEYPOT*, 192.168.3.2) can be accessed via SSH. Another machine (*U3*, 192.168.3.3), with real and sensitive data, is hidden but can be used to ssh to other devices.

The SSH relay is used to forward SSH connections:

- from the Internet to *HONEYPOT* and to computers in 192.168.1.0/24 and 192.168.2.0/24.
- from computers in 192.168.1.0/24 to *PWEB* (for SCP), to *RSYNC*, and to the Internet.
- from *U3* to computers in 192.168.1.0/24 and 192.168.2.0/24, to *HONEYPOT* and to the Internet.

The RSYNC server (*RSYNC*, 10.10.3.3) is used by *U3* to backup sensitive documents. The data can be transferred securely (meaning, the communication must be encrypted), or without any encryption. It is also used by the web team (192.168.1.0/24) to backup web data. In that case, the data **must** be encrypted.

Finally, the NFS server (*NFS*, 10.10.3.2) is used by *HONEYPOT* to backup/synchronize some fake 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 (web team) and 192.168.2.0/24 (desk team).

3.2 Accounts

Vlad and Donald have an account on most of the devices in the network. Two other accounts also exist: one for the web team (webteam) and one for the desk team (deskteam). You have access to their accounts to test your security system deployment. Their login are vlad, donald, webteam and deskteam. 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 `vlad`, `donald`, `webteam` or `deskteam`. 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`, `SCP`, `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 MDGA™ 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* (`172.32.3.2`)[2]. This computer is configured to use the mail server, the `SSH` server, and `PDNS` as `DNS` server. It also runs an ssh server, allowing incoming connections.

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 environment. 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` is the address of the destination device.

Note that each user uses a public-key authentication. This allows to not enter 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, *DT*, *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 `vlad`, and enter:

```
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 `webteam`, or if the destination is *PDNS*, 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 host names, they require a dns resolution).

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 network, the directory */home-/sharing* on the `NFS` server is shared with *HONEYPOT*. Each time this server 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 *HONEYPOT*, the directory is also mount at */home/sharing*, for the sake of simplicity. Once *HONEYPOT* 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*.

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

3.6 Remote Synchronisation (RSYNC)

RSYNC[3] is a software used to synchronize files. In the network, 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). Only one module is defined on the server. For `Vlad`, use the module *backup_ vlad*. The files are synchronized in the home directory of each user on the server (`vlad` and `webteam` have an account on RSYNC, but only `vlad` has a module).

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. Remarks in Sec. 3.4 still apply (public-key authentication and relay).

Note: For secured rsync, you must force the use of the SSH relay. As for unsecured rsync, you need to use the IP address of RSYNC since it is only defined in ssh context.

3.7 Domain Name Servers (DNS)

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

- `www.mdga.com`: address of the public web server
- `local.mdga.com`: address of the local web server
- `mail.mdga.com`: address of the mail server

3.8 Mail Service

`Vlad` and `Donald` (as well as members of web and desk teams) can send and receive emails with their MDGA™ addresses:

```
{vlad|donald|webteam|deskteam}@mdga.com
```

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 the 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 in as an existing user on that device*, 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 ULiège network, you can send emails from a MDGA™ 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 MDGA™ address, because this domain only exists in the virtual environment.

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

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 SCP

Secure Copy Protocol or SCP is a means of securely transferring computer files between a local host and a remote host or between two remote hosts. It is based on the Secure Shell (SSH) protocol.

The web team (*U1*) uses `SCP` to transfer files from/to the public web server (*PWEB*).

```
scp file PWEB:destination_directory
```

where `destination_directory` can either be `/home/webteam/` (for private files) or `/var/www/` (for public web files).

You need to be logged in as `webteam` (`su - webteam`).

3.12 `HTTP(S)` 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 rules 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

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

4.1.1 Submission

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

1. you must submit a `PDF` file named `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 zones and their order per firewall.
3. your PDF file must be uploaded on the submission platform (see `http://submit.montefiore.ulg.ac.be`) - **Please use the same group/team ID you were assigned to create a group on the submission platform**
4. the deadline is **October 29th, 2021, 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. Later on that day, the correct network drawing and zones will be provided on eCampus.

4.1.2 Gradings

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

4.2 Step 2: High-Level Rules

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

4.2.1 Submission

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

1. you must submit a `PDF` file named `Group-XX.pdf`, where `XX` refers to your group ID.
2. your PDF file will include the following items:
   - the high level rules, per firewall, structured as seen during the exercise session.
3. your PDF file must be uploaded on the submission platform (see `http://submit.montefiore.ulg.ac.be`) - **Please use the same group/team ID you were assigned to create a group on the submission platform**
4. the deadline is **November 12th, 2021, 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. Later on that day, the correct high-level rules will be provided on eCampus.

4.2.2 Gradings

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

4.3 Step 3: iptables Rules

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 *Netkit* virtual environment (you can execute it inside the virtual machine provided for the course, or build your own virtual machine if you prefer, or even run it directly if you have Linux installed on your machine).

4.3.1 Implementation

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

4.3.2 Submission

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

1. you must submit an archive named `Group-XX.tar.gz`, where `XX` refers to your group ID.
2. your archive will include the following items:
   - your report (as a `PDF` file) named `Group-XX.pdf`, where `XX` refers to your group ID.
   - all firewalls configuration files named `config_FWx.sh` containing the `iptables` implementation of firewall `x` (where $x \in [1;3]$).
3. your archive must be uploaded on the submission platform (see `http://submit.montefiore.ulg.ac.be`) - **Please use the same group/team ID you were assigned to create a group on the submission platform**
4. the deadline is **November 19th, 2021, 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. Each group will have to do a live demo. During the demo, your implementation might be tested in the virtual environment and you will have to answer several questions. The demo will be taken into account in the final grade. Organizational details will be provided on eCampus.

4.3.3 Gradings

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