Pentesting Scenario (I). Introduction and objectives (WIP)

2020-09-15

A virtual scenario will be designed to emulate the infrastructure of a medium-sized company, although in order to reduce the use of resources required for the deployment of the scenario, the services that have coherence in the same machine will be grouped together, making, for example, that different databases are in the same host.

The main objectives for the development of this scenario are as follows:

- In-depth study of the main services currently used, since it will be necessary to install and configure them properly.
- Creation of a scenario (basic) with the different configurations of the default services except for the logs, which will be sent to a centralized log server, which will allow us to position ourselves from two different points of view:
  - Red Team: whose objective is to carry out a study of these configurations looking for security flaws in them in order to later carry out attacks taking advantage of these default configurations.
  - Blue Team: At the same time, from this point of view we will perform an analysis of the logs to see what traces are left during the attack.
- Creation of a scenario (advanced) to which appropriate configurations will be applied according to the needs of each deployed service. At this stage we will also test different types of attack and analyze the logs to see the logs generated.
- Creation of a scenario (hardening) in which all machines and services will have more robust configurations applied following standards of guides such as CIS or NIST.

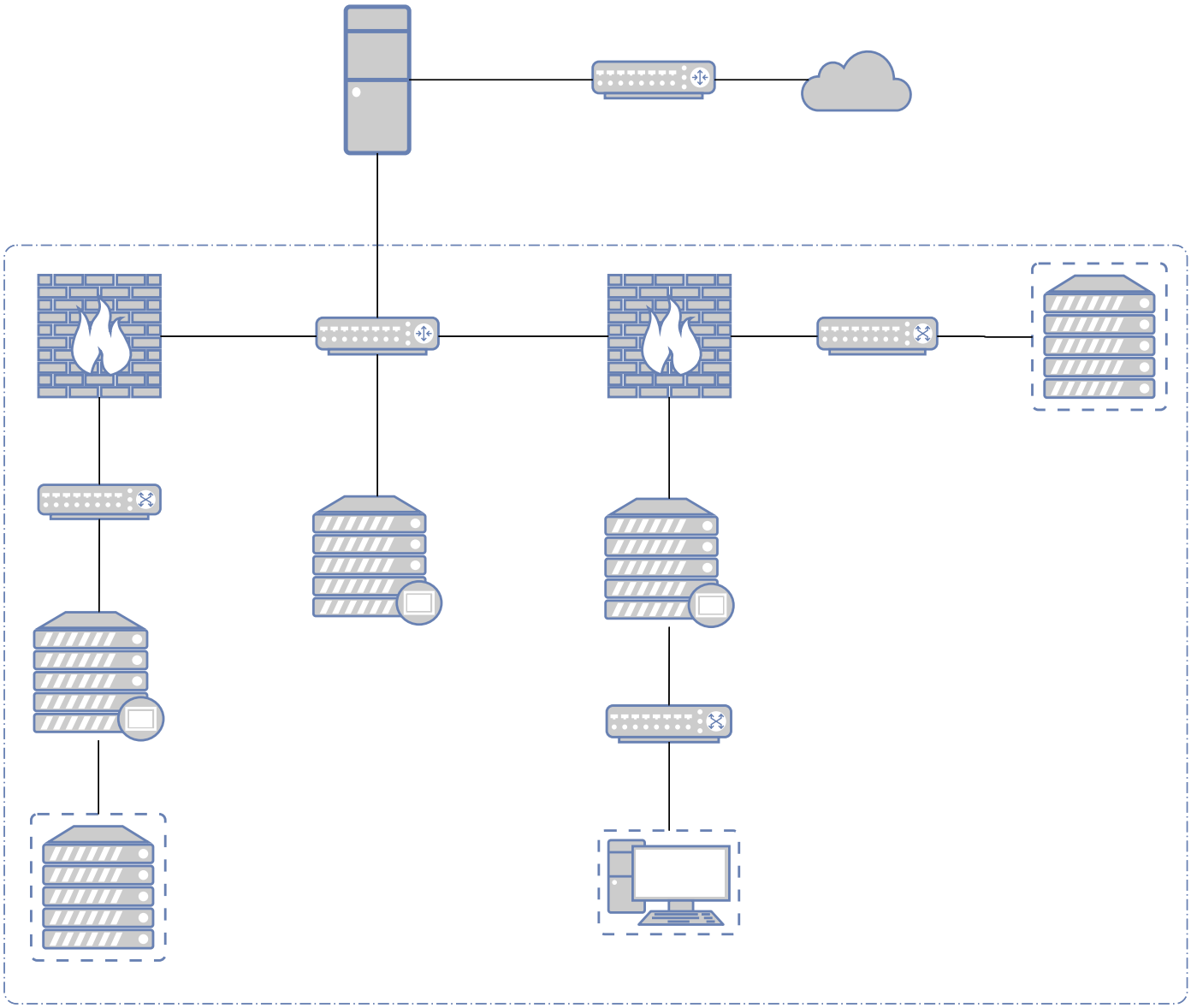
The main tools to be used for scenario deployment are as follows:

- **Proxmox**: Proxmox Virtual Environment, or Proxmox VE, is an open source server virtualization environment. It is on Debian-based GNU/Linux distributions with a modified version of the RHEL Kernel and allows the deployment and management of virtual machines and containers. Proxmox VE includes a Web console and command-line tools, and provides a REST API for third-party tools. Two types of virtualization are supported: LXC-based containers and KVM virtualization. It comes with an installer and includes a Web site-based administration interface.
- **Terraform**: Terraform is an infrastructure-as-code (IaC) software developed by HashiCorp. It allows users to define and configure a data center infrastructure in a high-level language, generating an execution plan to deploy the infrastructure on Proxmox, OpenStack, AWS, Azure, etc.
- **Bash**: For the automation of configuration and deployment of Proxmox, hosts and the different containers we are going to use scripts made in bash that perform an initial configuration.
- Ansible___: For the installation and configuration of services on all hosts. The objective is to create different environments being Ansible the one in charge of applying the more or less robust configurations according to the environment.

The operating systems to be used to deploy the different services are:

- Linux: Different Linux distributions will be used to make a more heterogeneous scenario, these Operating Systems are based on the main Linux distributions and are the following:
  - Debian 10
  - CentOS 7
- Windows: Windows based Operating Systems will be used for the user hosts and for two servers that have an Active Directory, these are as follows:
  - Windows Server 2019
  - Windows Server 2012
  - Windows 10
  - Windows 7
  - Windows XP

The following is a simplified diagram of how the architecture will be deployed, the computers inside the square are those that will be virtualized.



{:height="100%" width="100%"}