

# Docs on SwarmLab.io !

## Πίνακας περιεχομένων

1. Prepare your development and test environment.....	1
1.1. <a href="#">Install docker</a> .....	2
1.2. <a href="#">Install dual</a> .....	2
2. LINUX.....	2
2.1. <a href="#">An Introduction to Linux</a> .....	2
2.2. <a href="#">The proc filesystem</a> .....	2
2.3. <a href="#">VIM - A simple workflow example with the vim editor</a> .....	2
3. Internet of things (IoT).....	2
3.1. <a href="#">Intro IoT</a> .....	2
3.2. IoT swarm implementation.....	2
4. Security.....	3
4.1. <a href="#">Scan and network statistics</a> .....	3
4.2. <a href="#">Network analysis</a> .....	4
4.3. <a href="#">Network Filter</a> .....	4
4.3.1. <a href="#">Traversing of tables and chains</a> .....	4
4.4. <a href="#">Iptables with shorewall! - Three-Interface Firewall</a> .....	4
4.5. <a href="#">virtual private network (VPN)</a> .....	4
4.6. <a href="#">SSH Tunneling</a> .....	5

### Lab docs

**Internet of Things.** Is as a networked interconnection of devices in everyday use that are often equipped with ubiquitous mechanism.

**Security.** Is a set of mechanisms to protect sensitive data from vulnerable attacks and to guarantee confidentiality, integrity and authenticity of data.

## 1. Prepare your development and test environment

## 1.1. Install docker



**Docker** is a set of platform as a service (PaaS) products that use OS-level virtualization to deliver software in packages called containers.

Containers are isolated from one another and bundle their own software, libraries and configuration files; they can communicate with each other through well-defined channels

## 1.2. Install dual

# 2. LINUX

## 2.1. An Introduction to Linux



Absolute basics

## 2.2. The proc filesystem

## 2.3. VIM - A simple workflow example with the vim editor

# 3. Internet of things (IoT)

## 3.1. Intro IoT



How It Works, Apps, Swarm: The Five Principles of Swarm Intelligence

## 3.2. IoT swarm implementation



A swarm implementation that will allow communication between all of the members/nodes.

- [source code](#)
- [info in adoc](#)
- [info in pdf](#)

## 4. Security

**Security** is defined as a set of mechanisms to protect sensitive data from vulnerable attacks and to guarantee confidentiality, integrity and authenticity of data.

**Network security**, in a cloud environment (**IaaS, PaaS, and SaaS**) **OR Cloud of Things** consists of the security of the underlying **physical environment** and the **logical security** controls that are inherent in the service or available to be consumed as a service.

- Physical environment security ensures access to the cloud service is adequately distributed, monitored, and protected by underlying physical resources.
- Logical network security controls consists of link, protocol, and application layer services.

In a **cloud environment**, a major part of network security is likely to be provided by virtual security devices and services, alongside traditional physical network devices.

Typically, the inspection and control of network traffic do not pass through physical interfaces where classical control devices can analyze or block them.

This is the reason why effective controls require the integration with the software layer - *network security architecture, security gateways (firewalls, WAF, SOA/API), Security Products (IDS/IPS, Sub Tier Firewall, Security Monitoring and Reporting, Denial of Service (DoS) protection/mitigation, and secure "base services" like DNSSEC and NTP.*

### 4.1. Scan and network statistics



This tutorial demonstrates some common **nmap** port scanning scenarios and explains the output.

## 4.2. Network analysis



**tcpdump** is a common packet analyzer that runs under the command line. It allows the user to display TCP/IP and other packets being transmitted or received over a network to which the computer is attached. Distributed under the BSD license, tcpdump is free software.

## 4.3. Network Filter



Netfilter offers various functions and operations for **packet filtering**, **network address translation**, and **port translation**, which provide the functionality required for **directing packets** through a network and **prohibiting packets** from reaching sensitive locations within a network.

**iptables** is a command line utility for configuring Linux kernel firewall implemented within the Netfilter project. The term "iptables" is also commonly used to refer to this kernel-level firewall. It can be configured directly with iptables, or by using one of the many

### 4.3.1. Traversing of tables and chains



When a packet first enters the firewall, it hits the hardware and then gets passed on to the proper device driver in the kernel.

Then the packet starts to **go through a series of steps in the kernel**, before it is either **sent to the correct application** (locally), or **forwarded to another host** - or whatever happens to i

## 4.4. Iptables with shorewall! - Three-Interface Firewall



**Shorewall** is an open source firewall tool for Linux that builds **upon the Netfilter (iptables/ipchains)** system built into the Linux kernel, making it easier to manage more **complex configuration schemes** by providing a higher level of abstraction for describing rules using text files.

## 4.5. virtual private network (VPN)



A **virtual private network (VPN)** extends a private network across a public network, and enables users to send and receive data across shared or public networks as if their computing devices were **directly connected to the private network**.

## 4.6. SSH Tunneling



**SSH Tunneling**, is the ability to use ssh to **create a bi-directional encrypted network connection** between machines over which data can be exchanged, typically TCP/IP.