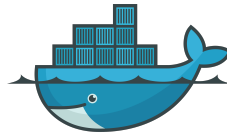# PiTest
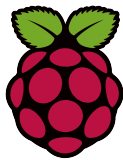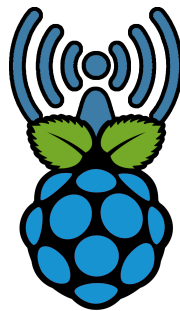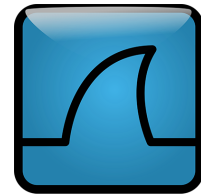*Penetration Testing on a mobile platform*

University of Padua

Project documentation

filippo.berto.4@studenti.unipd.it, berto.f@protonmail.com

Version	1.0.0
Radaction date	2017-06-30
Distribution	*Prof. Armir Bujari*

Indice

Elenco delle tabelle

Elenco delle figure

1 Introduction

1.1 Purpose of the document

This document describes the project, the analysis of the problem, its design, the choices taken, its use and how to extend its functionality.

1.2 Purpose of the product

This product has the purpose of simplify the use of security research software on a mobile context, allowing the user to access a complete set of tools from an easy to use interface on a mobile phone.
The final product is going to offer the following functionality:

- Secure login through a REST API;

- Obtain the list of allowed commands;

- Execute common tasks, as network and WiFi scanning;

- Execute predefined scripts;

- Execute arbitray commands, give by the user;

- Connect to the device through a WiFi hotspot;

- Controll the device using an Android App client;

- Show the results of the commands to the user in a simple GUI.

1.3 Usefull references

- **Project presentation**: www.google.com/drive;

- **GitLab documentation**: www.gitlab.com;

- **Git documentation**: www.git.org

- **Android documentation**: developers.android.com

- **NodeJS documentation**: TODO

- **Docker documentation**: TODO

- **Arch Linux Wiki**: wiki.archlinux.org

- **Black Arch documentation**: TODO

- **Raspberry Pi documentation**: TODO

- **Aircrack-ng documentation**: TODO

- **Wireshark documentation**: TODO

2   Problem analysis

2.1   Problematic factors

The activity of penetration testing, or security auditing, pose difficulties in various in both hardware and software fields, partilcularly the following ones:

Penetration testing tools   Most penetration testing tools only work on a Linux environment and some may require dependencies wich can conflic with eachother.

Penetration testing in mobility   Some tasks need to be executed on the field, and carrying a laptop could be difficult in some situation. A small and portable device is necessary to accomplish the work.

Penetration testing and computational power   The common tasks of security auditing can be splitted in two generes: the ones which need lot of computational power, such as hash cracking, and the ones which need less, such as wifi auditing and network traffic sniffing. Since the platform chosen is mobile, its calculational power and it battery life are limited, so the project will focus mostly on the second ones.

Penetration testing hardware   Most tasks need some specific hardware to be executable, such as Software Defined Radios (SDR) or WiFi adapters. The device must have the general connectivity needed to allow the use of those tools.

2.2   Solution proposed

The following solution has been chosen as the most adeguate to mitigate or completelly solve all the problems metioned:

Hardware   The platform choosen for the project is a Rapsberry Pi B+, a System On a Chip board, featuring a single core ARM v6 processor, 512MB of RAM memory, 4 USB ports and a Ethernet adapter, plus 40 GPIO expansion pins for extra expansion. The board is powerfull enough to accomplish common tasks and its power consumption is low enough to be suxcessfully powered my a USB powerbank. Also the board consts about 35$, meaning the total cost is very contained.

Penetration testing hardware   For WiFi analysis I choose to use an ALFA AWS051NH v2 adapter, capable of both monitor mode and packet injection, necessary in most WiFi attacks.

Operatig system   The operationg system is Arch Linux a lightweight GNU Linux distribution that is easy to modify for the user needs.

Architecture   The architecture choosen is a simple server-client, since the network used to communicate between the Raspberry Pi and the client is self contained and most of the time the server will have only one client connected at a time.

network   Since the necessity of a simple way to connect to the board and transfer any kind of data between the server and the client, I've choose to generate a WiFi hotspot on the board; this way the connection is wireless and fast and the ethernet port is free to use. The 3 USB ports remaining can still be used to control directly the board or to plug in other devices.

Software   The main difficulti with software is compatibility, so the most effective solution is to virtualize a minified system with only the applications needed for the task. The most effective way to achieve this is using a Docker container, specifically built for that single application. This solution allows controlled transparence of both hardware and software, excluding the possibility of dependency collision and incompatibility.

As base to the container software I've chosen to use a minified version of Black Arch, an Arch Linux based distro with built in support for a variety of penetration testing software, still mantaining small size and ease of use.

The main application, which exposes a REST API and allows to execute given commands, scripts and tasks is a NodeJS server. This choice allows a simple development, given the modularity of the framework, and cros-platform compatibility.

The network is controlled by a hostapd daemon on the main operati system and the network is managed by both a dhcpcd and a dnsmasq services, to allow network NAT.

3 Project definition

The project definition will be splitted in two sections, describing the server and on the client.

3.1 Server

The server application, from now on *PiTest*, is installed on the Operating System of the board and needs NodeJS to work.

3.1.1 Class diagram

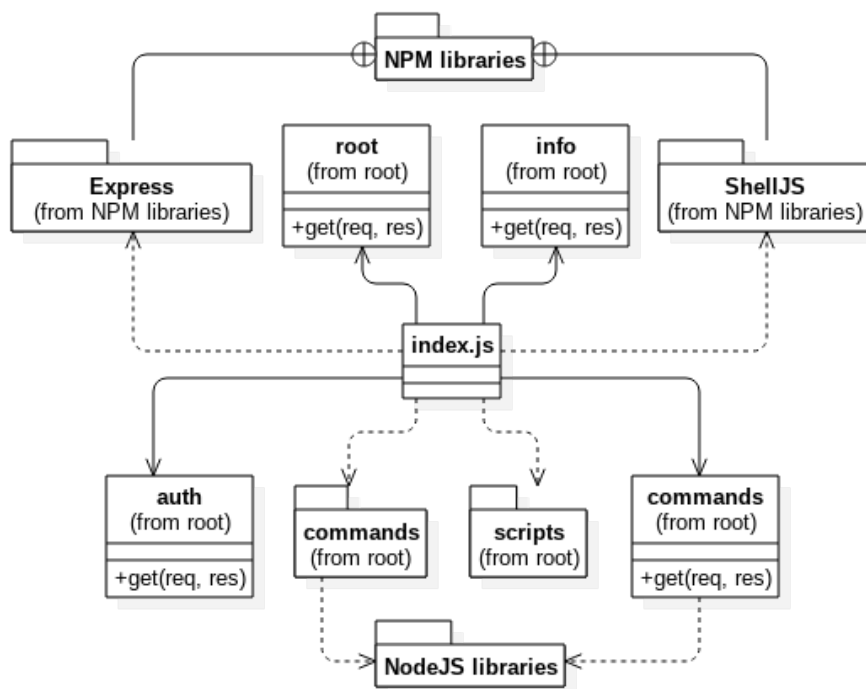The following is an high level diagram of the architecture of *PiTest*.

Figura 1: High level architectural diagram

3.2 Client