# Integration

## Device Integration

The program code will be contained within a Raspberry Pi. Raspberry Pi is a low-cost computer of small size developed by the Raspberry Pi Foundation in the United Kingdom with the purpose of helping promote the study of computer science and bringing programming to developing countries, but it also sees extended use in other fields such as robotics[[1]](#footnote-1).

Part of the success of the Raspberry Pi is due to the wide variety of models that exist, as well as their accessible price. By default the Raspberry Pi doesn’t include peripherals such as mice, keyboards or even a screen, but most models include several ports that allow such devices to be connected externally. Some models include an Ethernet port that allows the Raspberry Pi to access the Internet.

Raspberry Pi can work with many different operating systems, but this project will use Raspbian, a Debian-based Linux distribution and the primary operating system officially provided by the Raspberry Pi Foundation.

Note that this program can be executed from any laptop, and does not strictly require a Raspberry Pi to run. A Raspberry Pi is chosen for the purposes of these examples because of its small size and portability, as well as its low cost.

## Dependences

In order for the program code to run, a series of requirements or dependences must be installed beforehand in the device that will be used to run the scan, whether it is a Raspberry Pi, a laptop or any other compatible device.

First, the appropriate repositories must be cloned to the device. This can be easily achieved through the command line in any operating system that accepts orders via bash shell such as GNU/Linux or OS X. If prompted for a username and password while cloning from GitBook, those fields might be left blank.



Installing GitBook is also required, assuming it’s not installed already:

The Python code requires the following network manipulation libraries:



Nmap and its Python API can be installed with the following commands:



Finally, it’s necessary to install OWASP ZAP and its Python API. It might be downloaded and installed from the official page, or directly via shell like this:

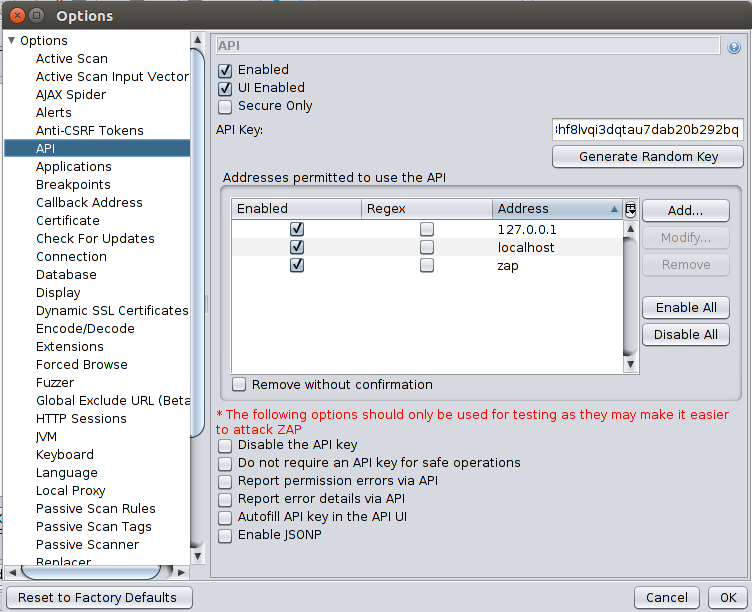


The target containing directory might be changed at will, remembering to change the appropriate *owasp\_location* variable inside the code to reflect that change. It is recommended to always install the newest version available.

In order to solve possible conflicts between different versions of urllib3, running the following command is recommended:



As a last step before the code can be executed, OWASP ZAP must be opened once (zap.sh file) in order to configure the API key. The API key is an optional security measure that isn’t strictly required to run OWASP ZAP, but is nonetheless recommended since it helps prevent malicious sites from accessing the ZAP API. Its configuration can be found under Tools → Options → API:



15 API key configuration in OWASP ZAP

The “Generate Random Key” button will, as its name implies, generate a new API key that can be copied to the *api\_key* variable used in the code. Alternatively, the “Disable the API key” option may be ticked and the *api\_key* variable left blank, but this is again not recommended.

## Remote Control

In order to avoid the need for a keyboard and a screen, it’s convenient to be able to control the Raspberry Pi remotely. This can be achieved via Virtual Network Computing (VNC).

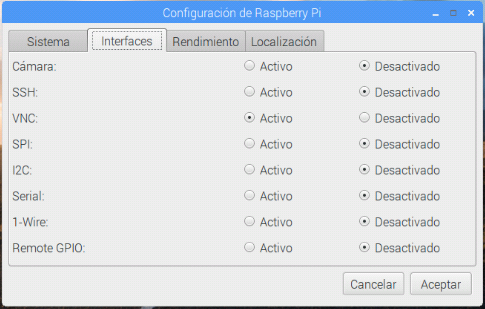
VNC is a graphical desktop sharing system consisting of a server and a viewer[[2]](#footnote-2). The VNC server (Raspberry Pi in this case) will receive any keyboard or mouse input coming from the VNC viewer, which will also act as a screen. The VNC viewer can be anything from a mobile phone to a tablet, or even a laptop.

By default, Raspbian includes VNC Connect from RealVNC, which includes both a VNC server and a viewer. It will be used as the VNC server for the Raspberry Pi, although any other VNC server could be used in its place. There is a wide variety of VNC viewers available for different platforms.

First, while VNC Connect is included by default in Raspbian installations, it’s still a good idea to run the following commands to ensure the latest version is installed:

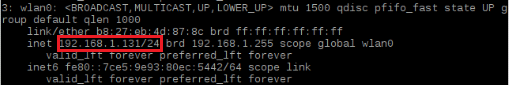


Next, it is necessary to enable the VNC server to allow external connections to the Raspberry Pi. This can be done graphically by selecting Menu > Preferences > Raspberry Pi Configuration > Interfaces and then setting the VNC option to Enabled. Alternatively, the VNC server can be enabled via command line using raspi-config and navigating to the Interfacing Options section.



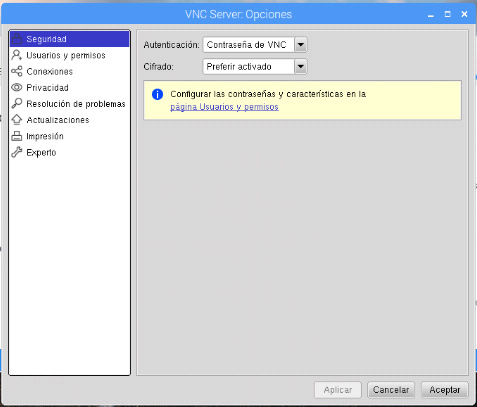
16 Configuration to enable VNC in Raspberry Pi

In order for the VNC viewer to connect to the enabled VNC server in the Raspberry Pi, it first needs to know its IP address. The IP address can be located by running *ip addr* on the Raspberry Pi and then checking the *inet* parameter in the appropriate section (eth0 for Ethernet connections, wlan0 for LAN connections, usb0 for USB).



17 Locating the Raspberry Pi's IP address via "ip addr"

Other than the IP address, the VNC viewer will require a password to successfully connect to the Raspberry Pi. For VNC Connect, a password can be specified under the Menu > Options > Security section, choosing “VNC password” from the Authentication dropdown menu. Depending on the VNC viewer, it might also be necessary to change the encryption option to a lower security setting to avoid conflicts.



18 Security configuration for the VNC server

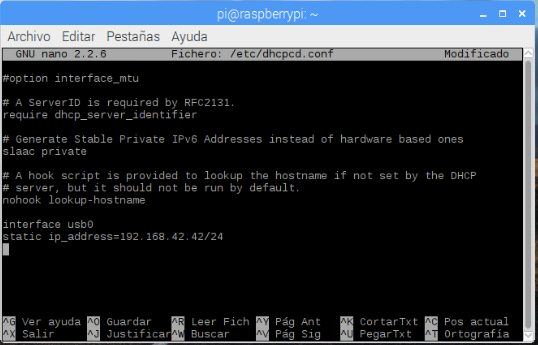
This method requires both the Raspberry Pi and the VNC viewer to be connected to the Internet. Connecting a Raspberry Pi to the Internet without a keyboard or a mouse can be tricky, so instead, it is advised to have the phone or tablet act as a Wi-Fi Hotspot for the Raspberry Pi to connect to. This only requires a conventional USB cable to connect the Raspberry Pi to the device that will act as the VNC viewer. The IP address to be supplied to the VNC viewer in this case is the one that corresponds to *usb0*, which can also be found by running *ip addr*.

By default, the Raspberry Pi gets assigned a dynamic IP address. This can be an issue, as the VNC viewer cannot connect if it doesn’t know the IP address of the VNC server, and finding the Raspberry Pi’s IP address requires a screen, which would defeat the purpose of using VNC in the first place. A simple solution is to assign a static IP address to the Raspberry PI. This requires editing the dhcpcd.conf file[[3]](#footnote-3):



The following lines should be added to the bottom of the file:





19 Configuration of the dhcpcd.conf file

*Interface* refers to the type of connection (such as usb0 for USB or wlan0 for LAN). The *ip\_address* will be the new static IP that will be assigned to the Raspberry Pi (the /24 at the end is necessary to specify a netmask of 255.255.255.0).

Note that it is necessary to run “*sudo service dhcpcd restart”* or reboot the Raspberry Pi for these changes to take effect. After this, it should be possible to control the Raspberry Pi from a phone or tablet even in absence of a keyboard or screen. This configuration requires only a power supply for the Raspberry Pi, a USB cable to connect it to the viewer device, and permission for this device to connect to the network.

From here, it should be easy to control the Raspberry Pi via phone or tablet to connect it to the enterprise network, and then run the appropriate scans.

1. http://elinux.org/RPi\_Hub [↑](#footnote-ref-1)
2. https://www.raspberrypi.org/documentation/remote-access/vnc/ [↑](#footnote-ref-2)
3. https://www.modmypi.com/blog/how-to-give-your-raspberry-pi-a-static-ip-address-update [↑](#footnote-ref-3)