INSTITUTO SUPERIOR TÉCNICO

DEPARTAMENTO DE ENGENHARIA INFORMÁTICA

FORENSICS CYBER-SECURITY

MEIC, METI

Tutorial I

Introduction to Forensic Tools

2019/2020

nuno.m.santos@tecnico.ulisboa.pt

Introduction

This guide will introduce you to the Kali forensics toolkit. Among the available opensource tools, the Kali linux is a Live CD distro that has a large number of security tools already installed and configured. These tools can also be used for the purpose of collecting and examining digital forensic artifacts. Kali will be used throughout this course for all our lab assignments.

Figure 1: Kali Linux.

This tutorial is split into two parts. The first part will help you to set up an environment in which you can run forensic tools and analyze digital artifacts. To this end, you must create a clean slate virtual machine and prepare it for storing artifacts using the Kali Live CD. The second part aims to provide an overview of the forensic tools you will find in the Kali distribution and to introduce you to how some of these tools are used for forensic activities. In particular, you have to perform several activities: explore unknown file artifacts, audit passwords using a password cracker, capture and examine network traffic, and audit existing public vulnerabilities of a given network.

Note that while some of the forensic tools contained in the Kali distribution can be used for post-mortem analysis, e.g., to examine file artifacts in the aftermath of a given incident, others can be used for live forensics, e.g., to collect evidence while a network penetration is taking place. For live forensics, in particular, an important step of every analysis process is to be prepared. There is often little time available to capture evidences before they disappear. For instance, open browse sessions, network sockets connections, executing processes, etc. Preparing your toolkit is important to enable you to act fast.

1 Preparing the environment

For the propose of this class we will emulate a real forensic analysis process using a virtual environment. This virtual environment can be reproduced using your own laptops. However, is it possible that some of the steps explained here need minor adjustments given the specific characteristics of your installation.

Before we begin, you need to obtain Kali. If you are executing this tutorial from the Lab facilities, download Kali linux from: `http://turbina.gsd.inesc-id.pt/csf1920/resources/kali-linux-2017.2-amd64.iso`. Otherwise, download it from one of the official mirrors.

1.1 Configure the forensic (virtual) station

We will be using VirtualBox for virtualizing Kali. You may download VirtualBox from `https://www.virtualbox.org/`. To configure the forensic virtual station, proceed as indicated in the following screenshots.

Start by clicking on the *New* button at the toolbar.

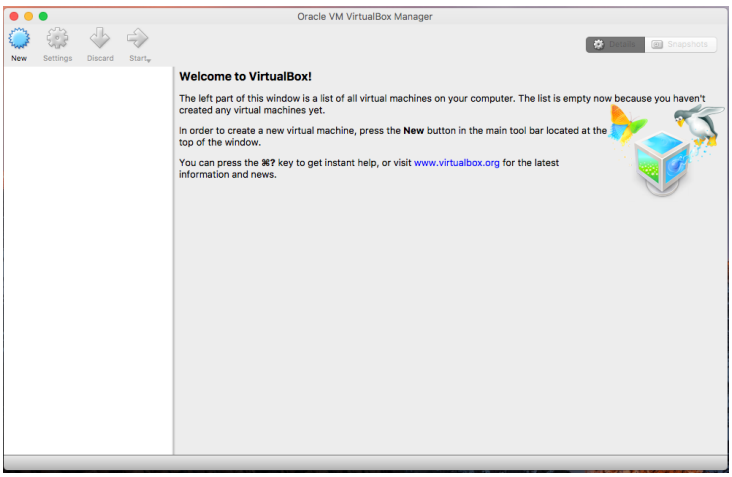

Figure 2: Create a new VM

Next, customize the VM. Set at least 2GB of memory. For the workstations at the lab, choose 4GB. Adjust the other parameters according to the Figure below.

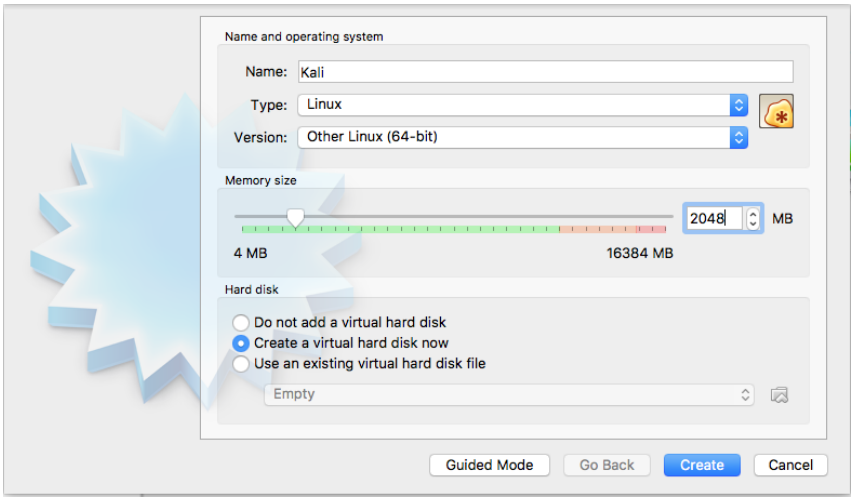

Figure 3: Customize the VM.

Adjust the storage size to at least 20GB. In real world scenarios, we often require 100s of GB, due to the need of saving disk images and analysis/extractions of evidences in these disks.

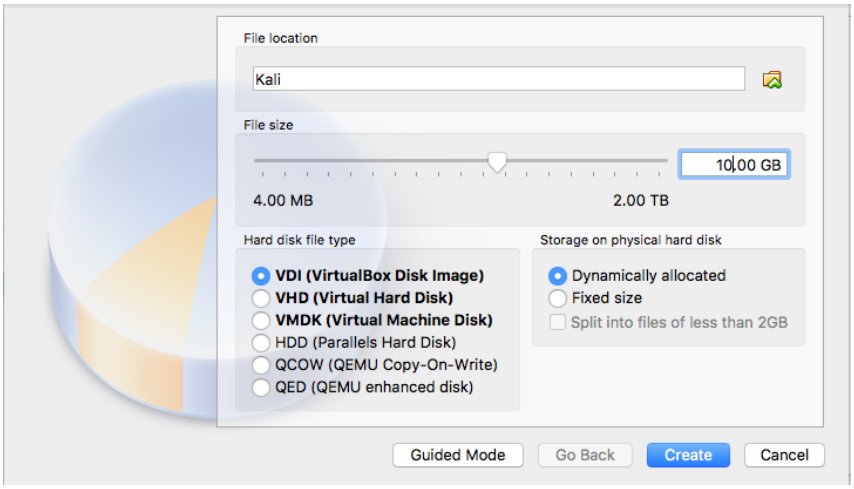

Figure 4: Define the Storage size

Finally, associate the downloaded image to the virtual machine. Mark also the **Live CD/DVD** option.

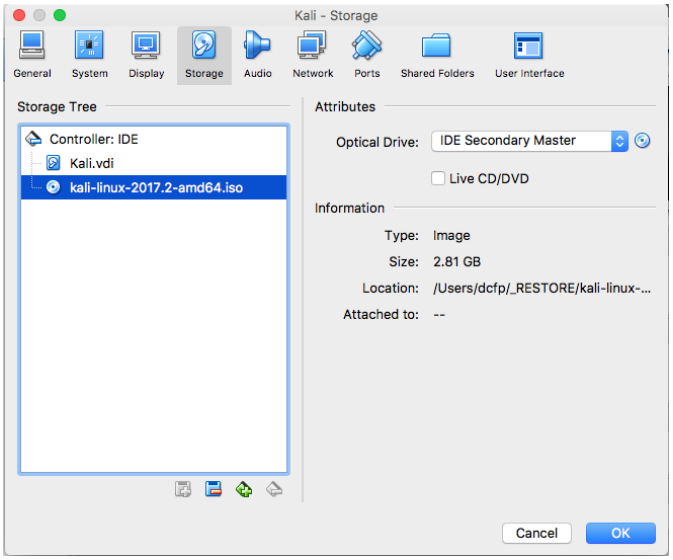

Figure 5: Associate the Kali image.

1.2 Starting Kali linux

Start the virtual machine selecting the VM at the Virtualbox side pane and then choose *start*. Once the VM boots up, select "**Forensic mode**" and then press *enter*.

Figure 6: Kali Linux - Boot screen

1.3 Preliminaries: preparing the storage of artifacts

Throughout this class we will find and analyze many important forensic artifacts. Some of these artifacts may become important evidence of an attack. Thus, we will create a repository in which we can save and organize all information collected during the analysis. The disk used to store digital evidence must be clean (zeroed) or new. Thus, in practice the following are steps often executed by the forensic analyst in the beginning of every analysis.

Once Kali is started (from the live CD), open a terminal using the button on the vertical toolbar as shown in Figure 7. We will execute all commands from this window.

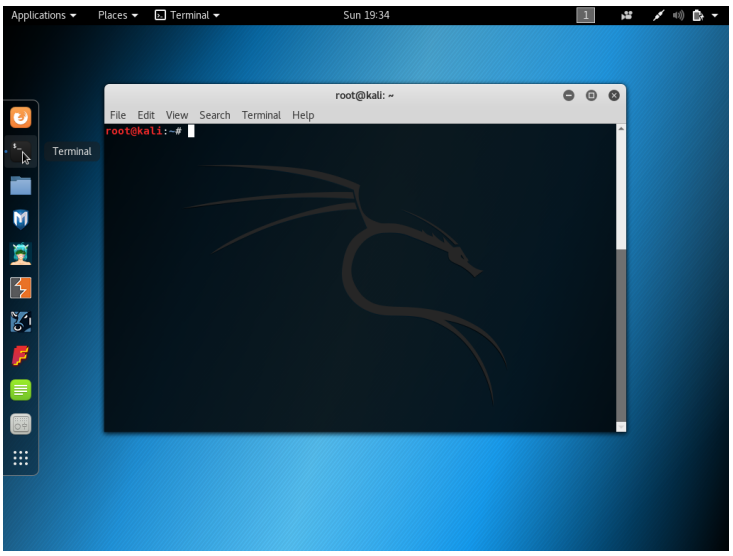

Figure 7: Open a Terminal

If necessary, configure your keyboard. The default is US layout. For setting the Portuguese layout use:

```
root@kali:~# setxkbmap -model latin9 pt
```

Use Google to find out more about your specific layout.

Next, you can list the storage devices connected in the system using *cat /proc/partitions* command as shown in Figure 8.

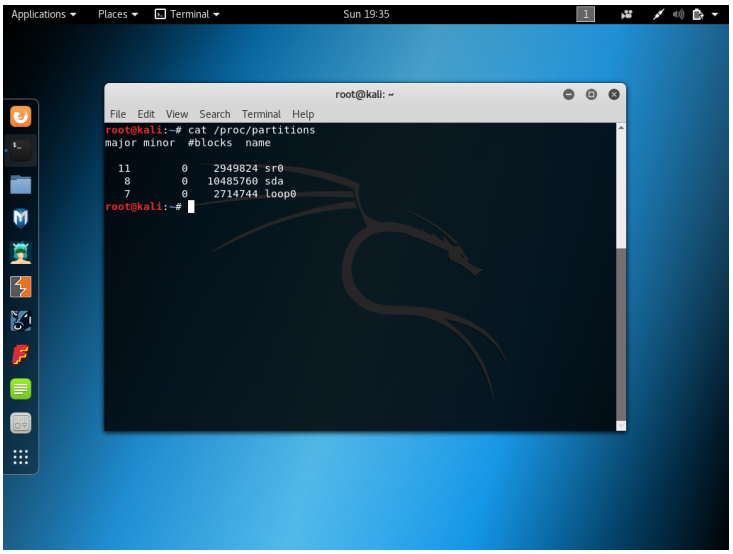

Figure 8: List connected storage devices.

Note that the disk – created during the VM configuration – is connected as *sda*. However there are no partitions. To make the disk usable, let's start by creating a partition in which we can store our artifacts using the *cfdisk* command as shown below.

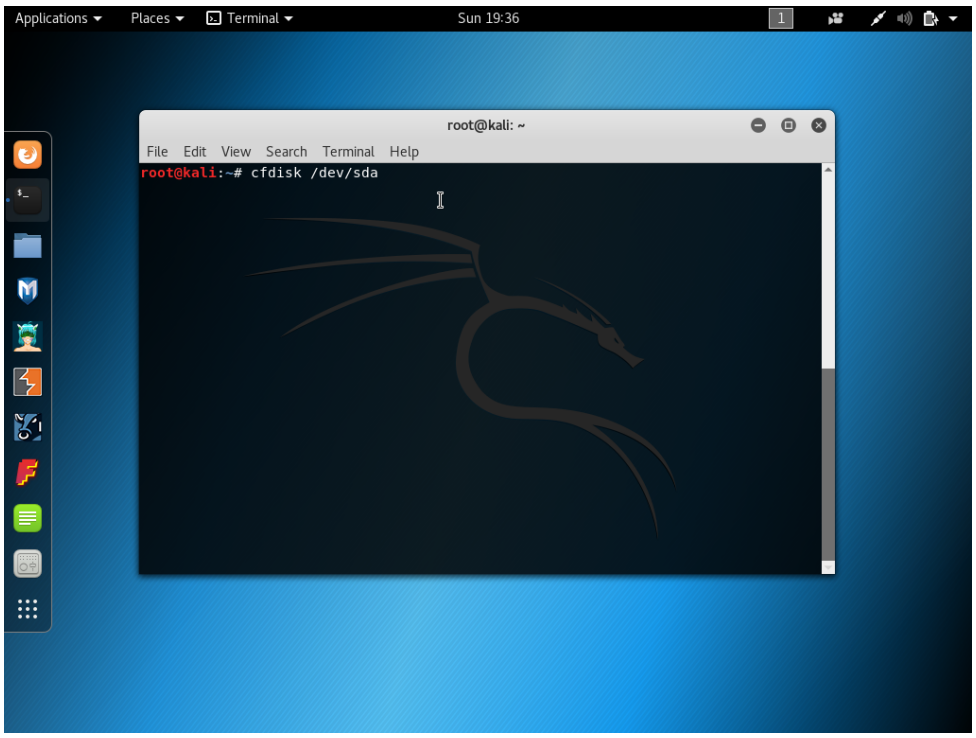

Figure 9: Run cfdisk to edit the partition layout.

Because the newly created disk was never formatted, we need to choose a partition scheme as shown in Figure 10. Choose DOS partition layout as it is compatible with the most popular operating systems.

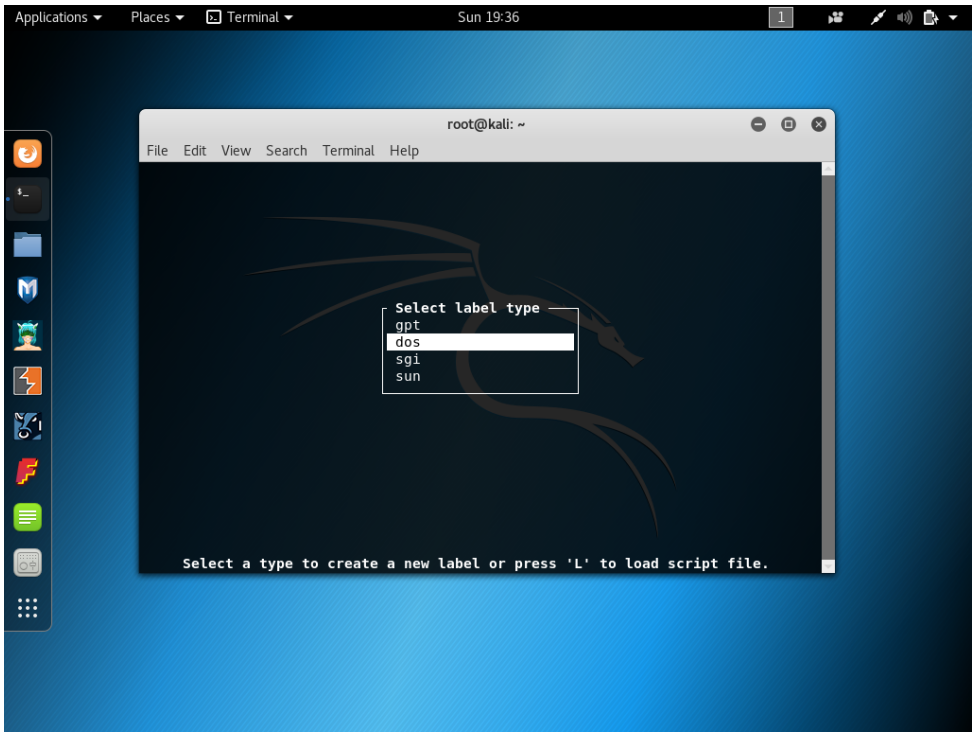

Figure 10: Choose the partition layout.

Next, in the *cfdisk* interface, choose *NEW* to create a new partition, described in Figure 11.

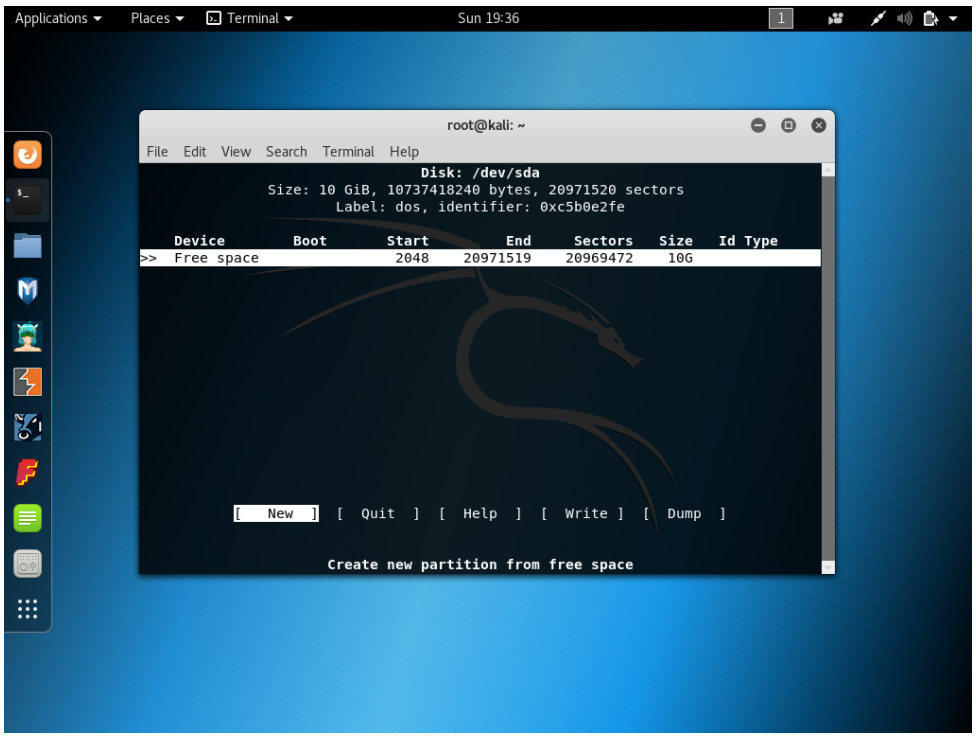

Figure 11: Create a new partition.

Then, choose *primary* as the type of the partition (Figure 12.

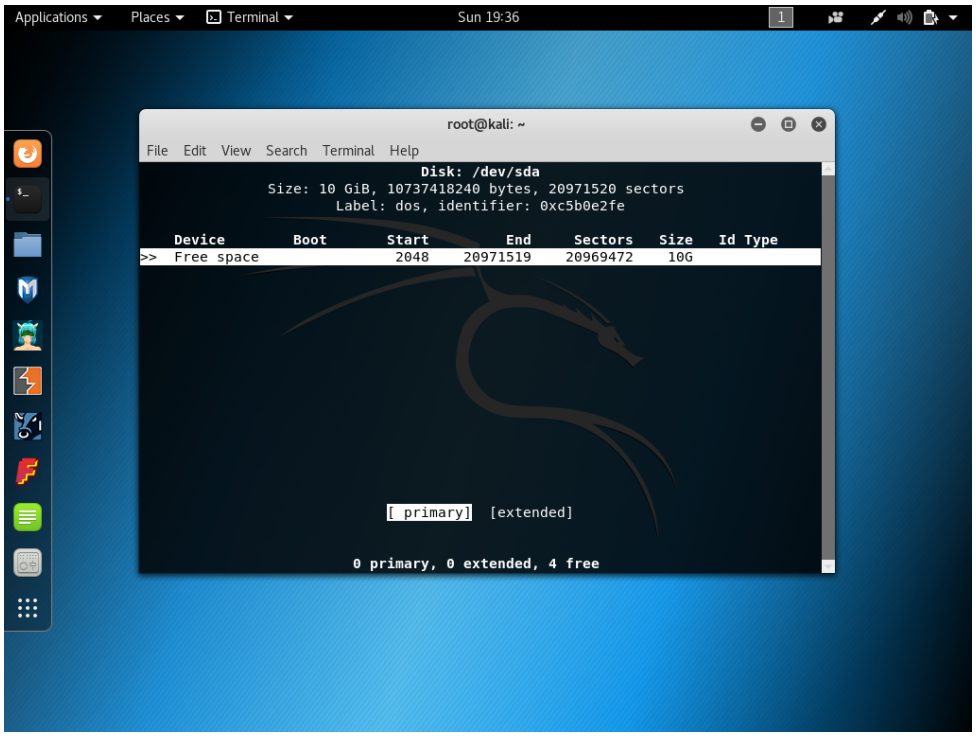

Figure 12: Type of the new partition.

Mark the new partition as bootable.

Figure 13: Mark the new partition bootable

Write all changes to the new disk using the *Write* option and type *yes* as shown in Figures14 and 15.

Figure 14: Write the new partition layout into the disk.

Figure 15: Confirm to write the new layout

Next, list the partitions again. If everything is done right, you should see an output similar to one shown in Figure 16. Note the new partition created (*sda1*).

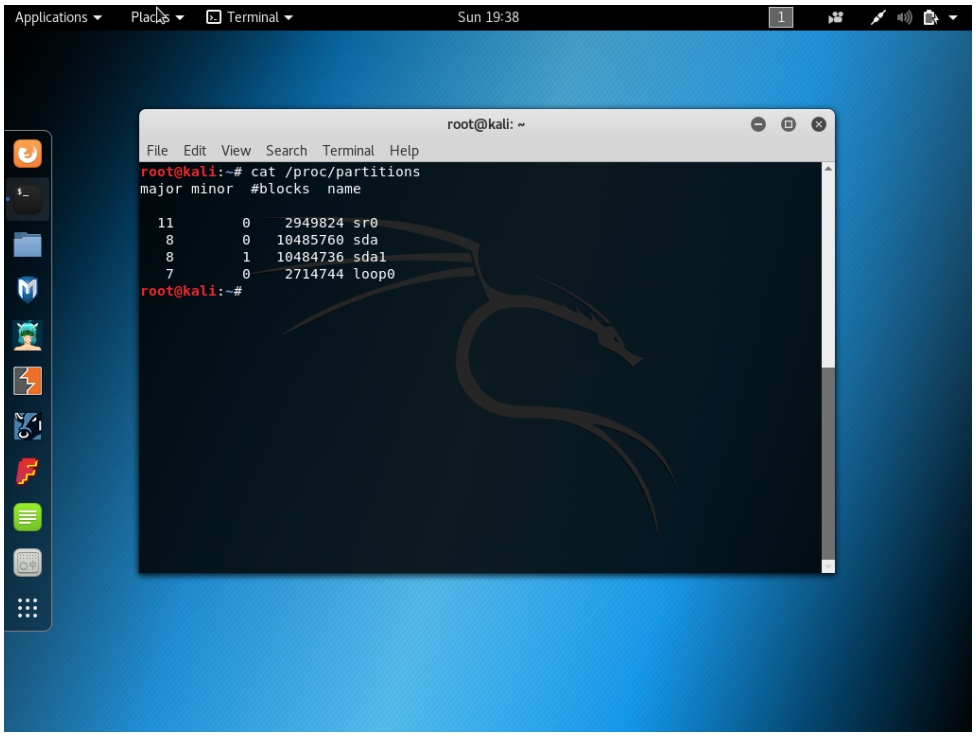

Figure 16: List the new partition layout.

Next, we need to format the disk according to the layout defined in the partition table. We can achieve that by executing the *mkfs -t ext4 /dev/sda1* command as instructed in Figures 17.

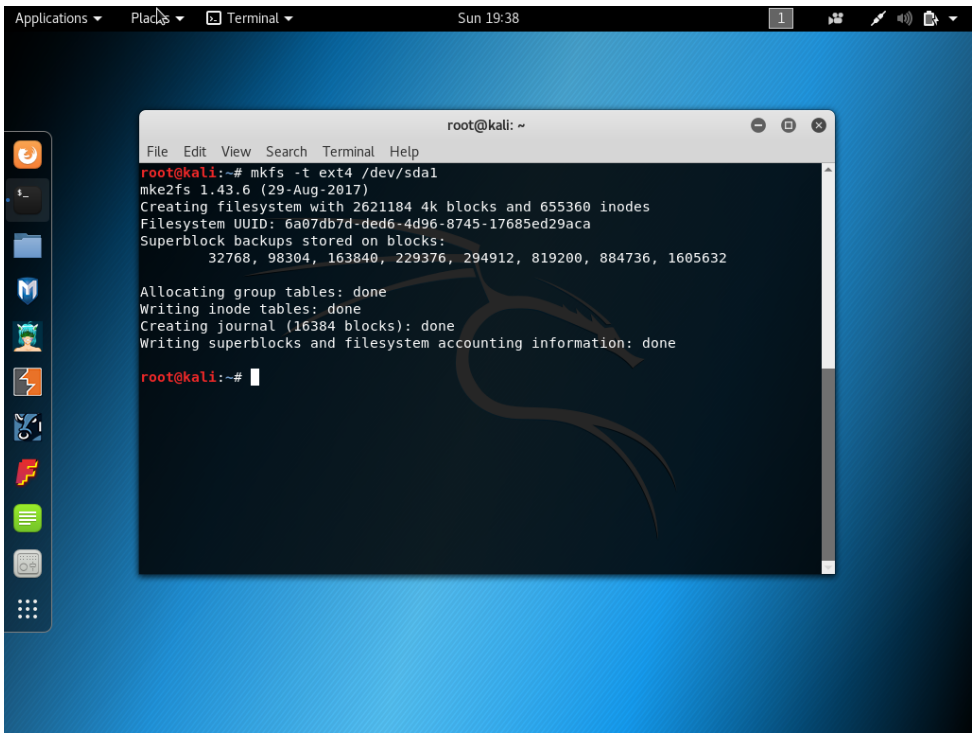

Figure 17: Formatting the new storage.

Finally we are ready to use the new disk. Create a mount point and mount the disk into the system as shown in Figure 18.

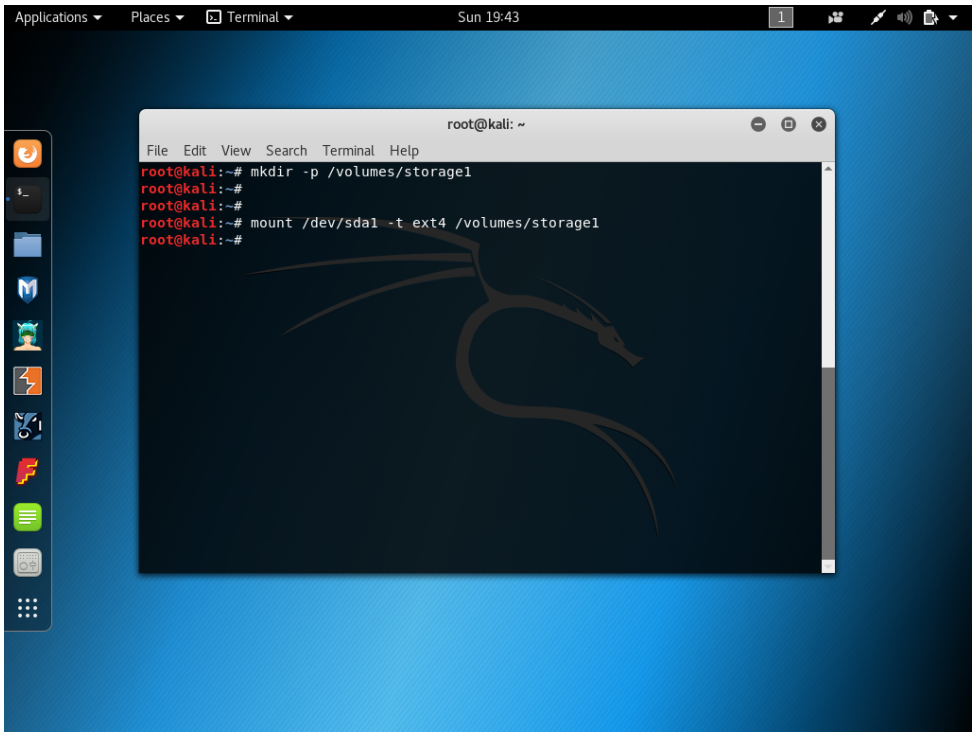

Figure 18: Mounting the storage.

Note: validate that you actually are using the new disk using the command *mount*.

1.4 Explore the tools available in Kali

The Kali website has a description of all tools installed into the distro. Tools are organized in several sections according to the kind of analysis they are designed to do. Navigate the menus as shown in Figure 20 and check the website `https://tools.kali.org/tools-listing` for more details.

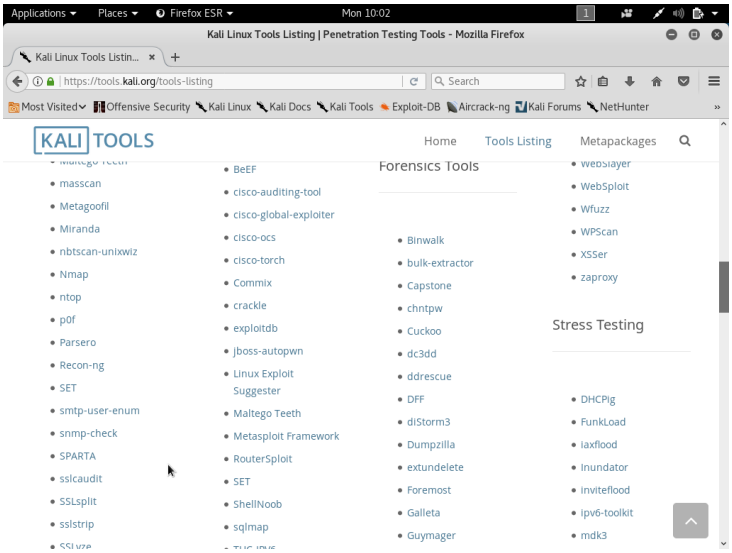

Figure 19: List of forensic tools.

Figure 20: Forensic tools accessible from the start menu.

2 Tools for exploring unknown artifacts

We now begin several exercises that illustrate how to use some of these tools for specific forensic purposes. Some files found during the analysis may not be exactly what they look like. For instance, an attacker can rename a malware with the name of the utility *ls* used to list files. To make things worse, sometimes the modified program still does the same operations as the original one but can also do more such as transfer files through the network.

In this exercise we will try to figure out information about artifacts without knowing exactly what it is. To do that we will use three tools. One is *hexdump*, which prints the content of the file in hexadecimal, as the name indicates. Also, the command *ldd* list the external libraries to which a binary is connected.

It can give us a hint about strange binaries. And finally, the command *strings*, which converts and prints all the readable strings in a chuck of data. You can combine these command with other known tools such as *grep*, *head*, *tail*, *awk*, *cut*, etc... To find out more about these commands use the manual: **man command**. Ex. *man hexdump*; *man ldd*....

Open a Terminal in Kali, access the *storage* you created and mounted earlier, and download the following artifacts:

```
root@kali:~# cd /volumes/storage1

root@kali:~# wget http://turbina.gsd.inesc-id.pt/csf1819/unknown_file

root@kali:~# wget http://turbina.gsd.inesc-id.pt/csf1819/unknown_file2
```

What do these downloaded artifacts actually do?

3 Tools for auditing passwords

Weak passwords are a major and common vulnerability. To audit weak passwords, we will use a tool named "John the ripper". This tool operates in three modes; we show the simple (but powerful) mode.

Specifically, we want to demonstrate this mode by finding out the system password of the newly configured VM. Recall that after booting, the system started up and gave a prompt with administrative rights. We never had to type the actual root password. Assume that we now need to figure out what the password is. For that, open a terminal. Examine the files in which Linux stores passwords as shown in listings below:

```
# look at the content of the passwd file
% root@kali:~# less /etc/passwd
```

We can see all existing users in the system but not their passwords. Next, we can explore the *shadow* file. Where the passorwds are stored but encrypted.

```
# look at the content of the shadow file
% root@kali:~# less /etc/shadow
```

Next, let's audit our system to find weak passwords. Copy both files to the storage we have just created and mounted:

```
# first copy the files
% root@kali:~# cp /etc/passwd /etc/shadow /volumes/storage1
# switch to the storage directorty for starting the analysis
% root@kali:~# cd /volumes/storage1
```

Combine those two files (the tool plugin requires that the file is compatible with an old format - for backward compatibility).

```
% root@kali:~## unshadow  passwd shadow > mypasswd
```

Finally run "John the ripper" as follows:

```
% root@kali:~# john mypasswd
```

When it finishes, you can see all cracked passwords using the following command at the same directory in which you run John for cracking the password file:

```
% root@kali:~# john mypasswd --show
```

4 Tools for network analysis

In this exercise, we will introduce network analysis tools and show how traffic analysis can be used by a hacker to gather sensitive information from our network. We deployed a simple (although common) installation of a popular blog platform (Wordpress)[1].

Inside Kali, open the web browser and go to the URL: `http://turbina.gsd.inesc-id.pt:8081/`. You should see a page like in Figure 21. Scroll down the page and open the link "login" at the bottom right side of the page to open the login page.

Fill in the form with the following credentials: (adminCSF1920/#csf1920#). **Do not click in login button after filling the form**. We only want to let it prepared.

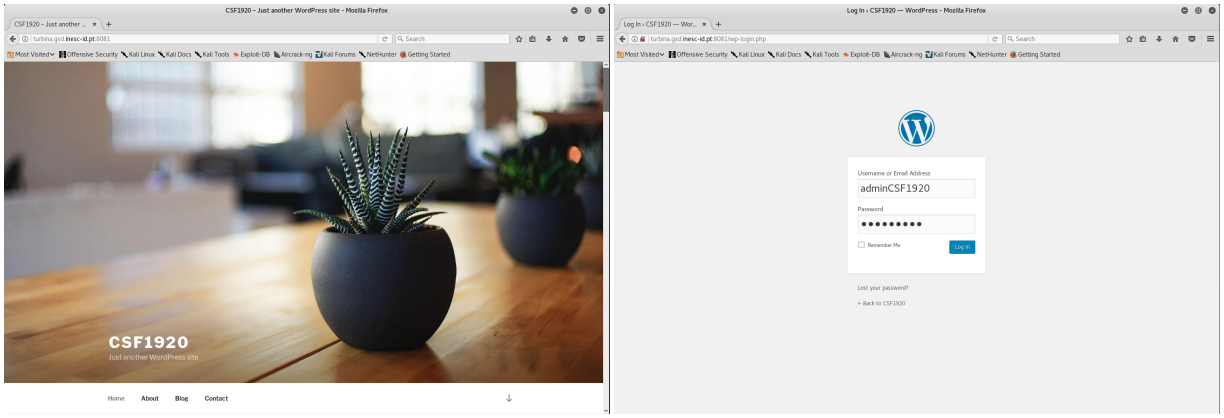

Figure 21: Preliminaries - prepare to capture traffic.

Next, open Wireshark as shown in Figure 22. Ignore the warning message that appears and advice us to not run it with *root* privileges. Wireshark is a *network sniffer*. It passively captures and analysis all packets arriving at the network interface. Combined with switch attacks, it can effectively capture the traffic of all devices connected to the same subnet.

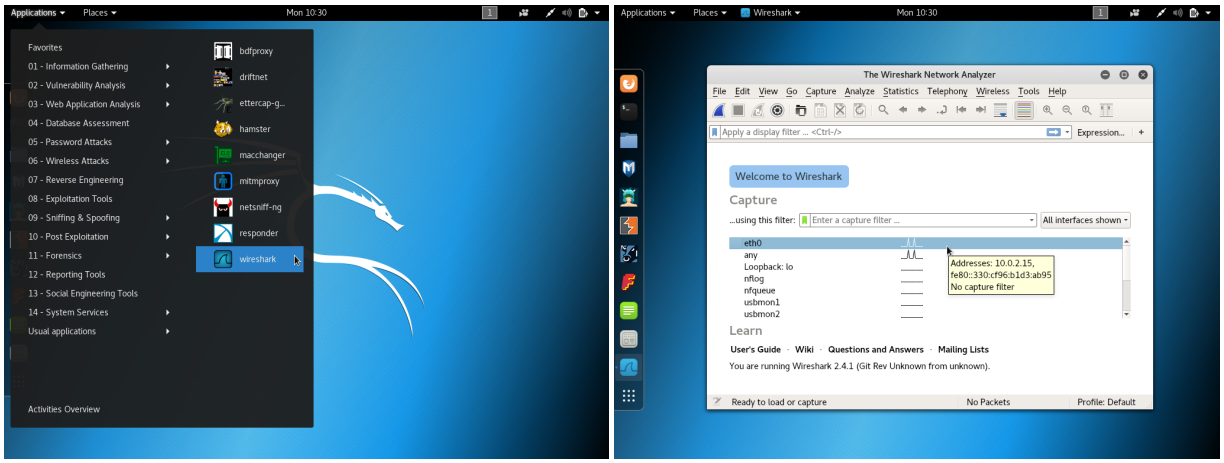

Figure 22: Opening Wireshark Graphic Interface

In Wireshark, navigate to the menu *Capture* → *Start* to initiate the capture. You should start seeing some packets being captured. Then, go back to firefox (alt+tab) and now click on the login button. Once logged in, go back to wireshark and stop it in the red button at the toolbar as shown in Figure 23.

[1] If the service is no longer online you still can analyze the packages captured earlier in this activity. Please download `http://turbina.gsd.inesc-id.pt/csf1920/wireshark.zip`, extract the content, open the "pcap" file in wireshark and jump to Section 4.1

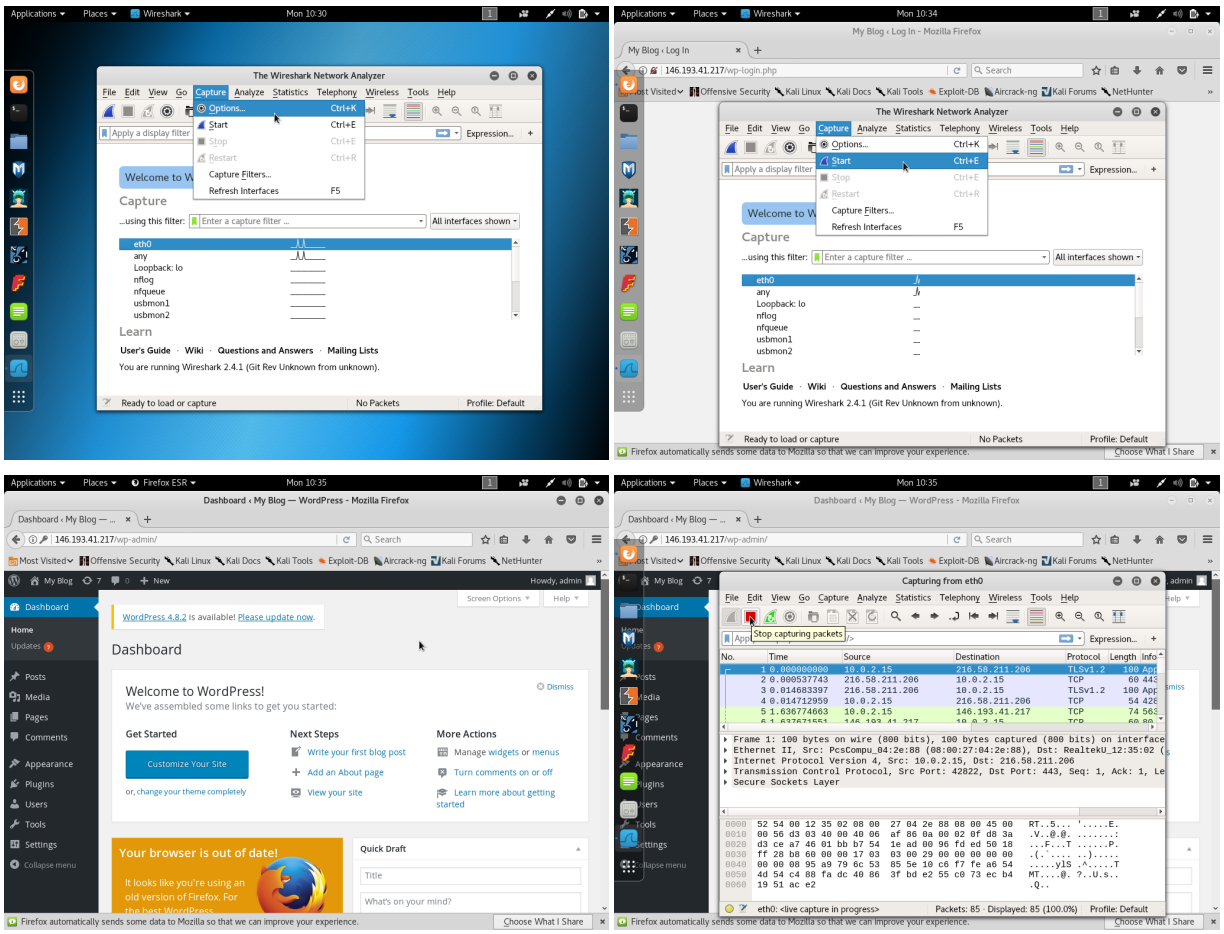

Figure 23: Wireshark capture configuration.

4.1 Analyzing the traffic

Now, let's analyze what has being captured. Find the first packet with **HTTP** traffic to our server (146.193.41.153). Then, do a right-click on it and then point to *Follow* menu option. Next, choose "*HTTP Stream*". You should see the HTTP conversation between the two hosts as shown in Figure 24. Can you spot the password?

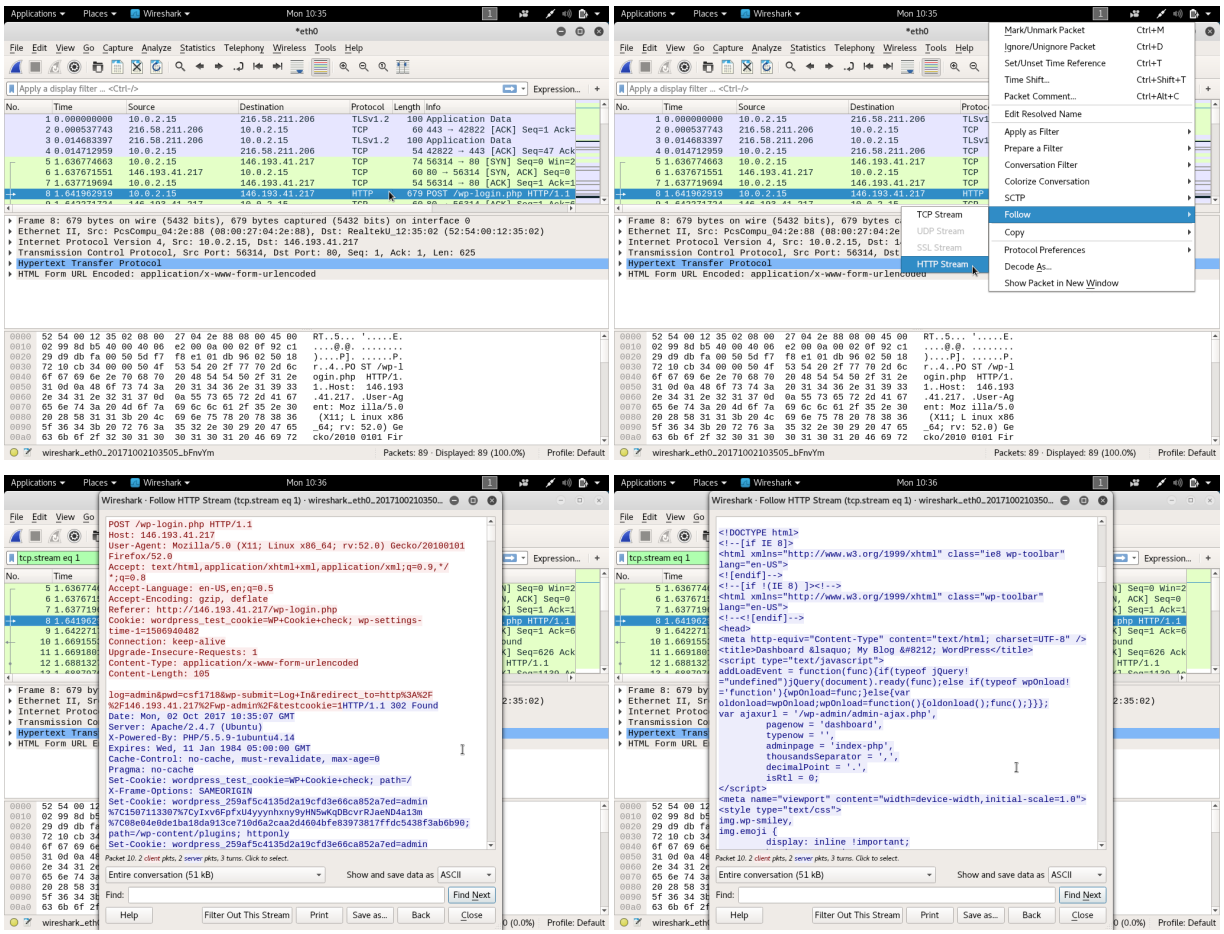

Figure 24: Wireshark capture

Finally, it is worth noticing that using a sniffer although difficult to discover it is not untraceable. As we can see in Figure 25 when we start the capture, wireshark puts the network interface into a special mode, named "*promiscuous mode*" to gather the packets. However this require administrative permissions and the kernel automatically outputs an alert of such behavior. While detecting a sniffer locally is simple, finding sniffer running in the network is challenging but possible. You can google for it to find out more.

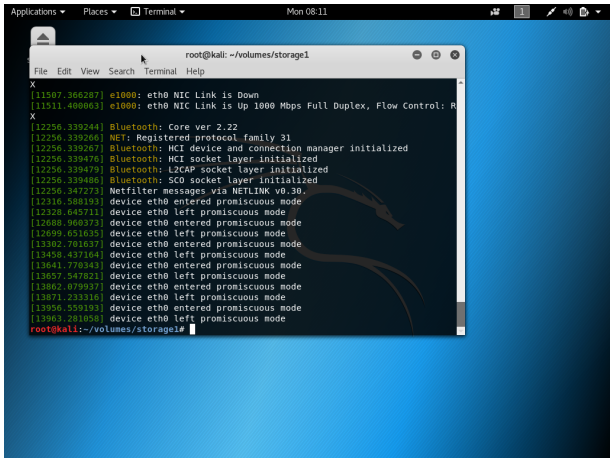

Figure 25: Kernel messages.

5 Tools for auditing public network vulnerabilities

The previous example might be seem artificial to begin with because we disabled the encryption in http communication (https/ssl). Nevertheless, many services are actually poorly configured like that.

To make matters worse, embedded devices and IoT devices, such as webcams, network home appliances, mobile hot-spots, often use web configuration interface and due to the high cost of having valid certificates (and the usability challenge that comes with invalid certificates normally blocked by all popular web browsers), use simple, non secure, web interfaces through which user passwords goes in plain text, unprotected.

One popular tool used by hackers (that can be also used to audit our managed machines) is Shodan – a search engine for connected devices `https://www.shodan.io/`. Recall that, unless specially motivated, most hackers goes to the easiest targets often indexed in Shodan, as seen in Figures 26 and 27.

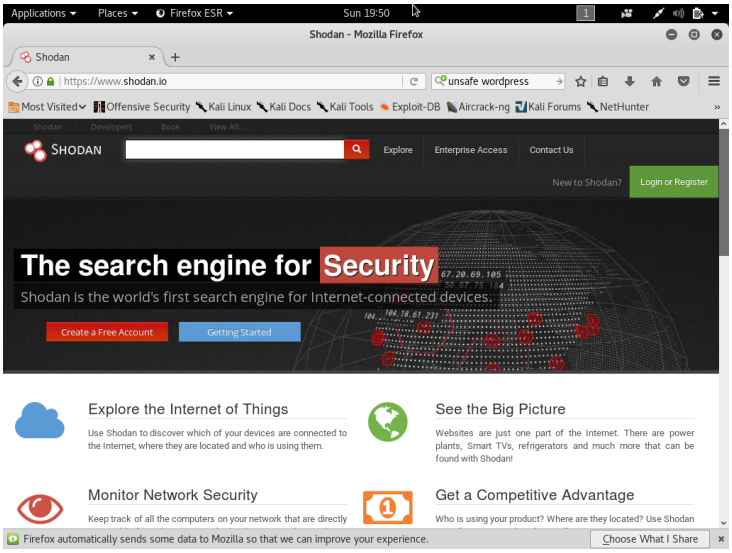

Figure 26: Shodan Search Engine.

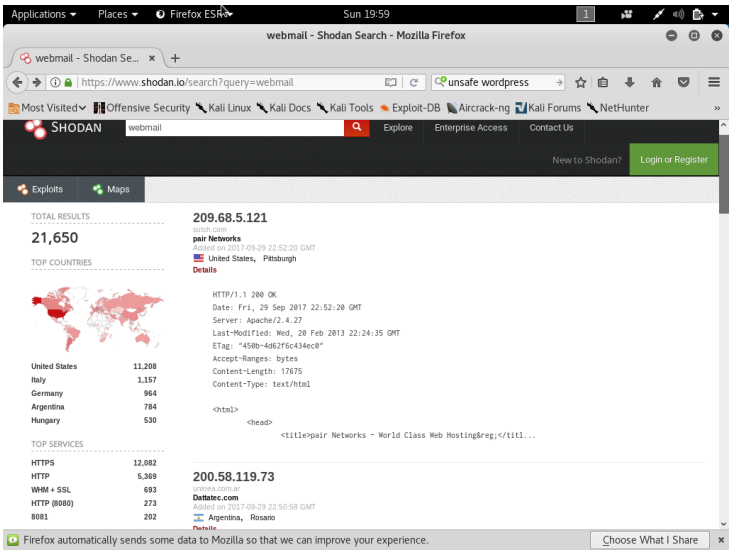

Figure 27: Shodan vulnerable .