Cuckoo Sandbox User's Guide

release 0.2

November 2, 2011

Abstract

This paper is the Cuckoo Sandbox user's Guide. Get the latest release in the project homepage: `http://www.cuckoobox.org`.

Contents

1 Introduction

This is the official Cuckoo Sandbox user's guide. This guide is designed to explain what Cuckoo is, how it works and what you can do with it.

The latest release of this document can be found on the official website at: `http://www.cuckoobox.org`.

Cuckoo is an Open Source dynamic malware analysis system which allows you to get information on suspicious files in a completely automated fashion.

1.1 Requirements

Cuckoo Sandbox makes only use of the following components:

1. Python for automating the analysis process and managing the tasks.
2. VirtualBox for managing virtualized analysis environments.

1.2 Use case

Cuckoo Sandbox is used to analyze Windows binaries and get comprehensive results describing their behavior while executed.

Its original purpose is to automate the analysis of malwares in a virtualized and isolated environment and provide all needed information to perform intelligence and forensic analysis over them.

Cuckoo can also process and analyze PDF documents, Office documents, scripts and any other file type for Microsoft Windows platforms.

Beside this, Cuckoo's use spectrum is much wider: it can be used to automate specific tasks on given files, perform customised analysis and anything you could possibly imagine thanks to its scripting capabilities.

1.3 How to get Cuckoo

You can get the latest Cuckoo release package from the official website: `http://www.cuckoobox.org.`.

For the most recent development version you can clone the Git repository at: `http://github.com/cuckoobox/cuckoo`.

Please consider that even if the Git version might include newer functionalities and bug fixes, it should be considered an "under development" branch, therefore it could be not completely stable.

2 About Cuckoo Sandbox

Cuckoo Sandbox is a dynamic malware analysis system which allows you to get informations on suspicious files in a completely automated fashion.

And it's free and Open Source.

2.1 Features

Cuckoo Sandbox provides you with a fully automated system able to fetch files, analyze them inside an isolated virtualized Windows system and return back results, such as:

- Network traffic dump generated during malware sample execution.
- Dump of dropped files.
- Screenshots taken during the whole malware analysis process.
- Relevant Windows API calls tracing of all recursively spawned processes.

2.2 Architecture

Cuckoo Sandbox consists of a central management software which handles sample execution and analysis. Each analysis is launched in a fresh and isolated virtual machine.

Cuckoo's infrastructure is composed by an Host machine (the management software) and a number of Guest machines (virtual machines for analysis).

The Host runs the core component of the sandbox that manages the whole analysis and execution process, while the Guests are the isolated environments where the malwares get actually safely executed and analyzed.

The following picture explains Cuckoo's architecture:

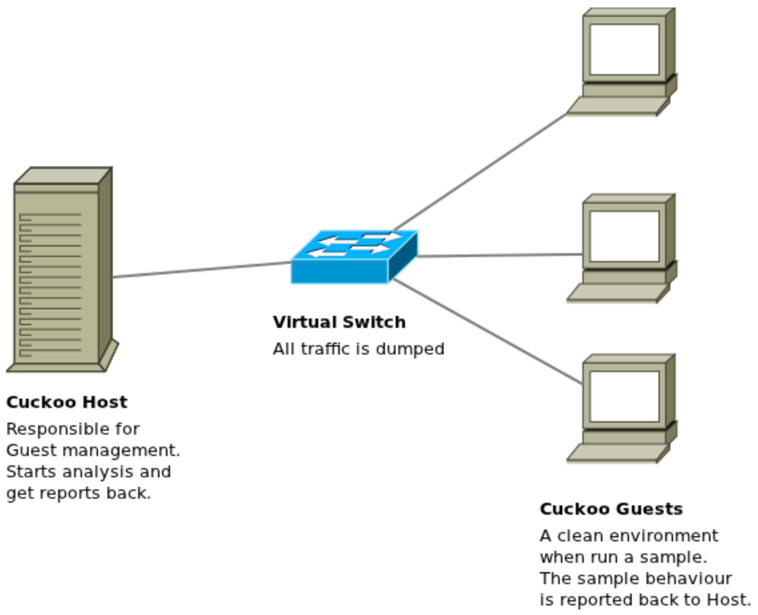


Figure 1: Cuckoo's architecture.

2.3 History

Cuckoo Sandbox started as a Google Summer of Code project in 2010 within The Honeynet Project.

It was designed and developed by Claudio "Nex" Guarnieri, who still maintains it and coordinates all efforts from joined contributors.

After initial work during the summer 2010, the first beta release was published on Feb. 5th 2011, when Cuckoo was publicly announced and distributed for the first time.

Since then it has been undergoing a lot of changes and improvements which lead to current stable release, 0.2.

In March 2011, Cuckoo as been selected again as a supported project during Google Summer of Code 2011 with The Honeynet Project during which Dario Fernandes contributed extending its functionalities.

3 Installation

3.1 Preparing the host

In this guide we are assuming that you are setting up Cuckoo Sandbox on a GNU/Linux host system.

Even if it has proven to work smoothly also on Mac OS X 10.6 Snow Leopard and Mac OS X 10.7 Lion, at current stage OS X should not be considered an officially supported platform.

3.2 Checking requisites

The following requisites must be met to setup a working Cuckoo Sandbox.

3.2.1 Python

Make sure that you correctly installed the appropriate version of Python: for current Cuckoo Sandbox you are supposed to use Python 2.7 (preferred) or 2.6.

To check if your current version is correct, launch the following command in your terminal:

```
$ python --version
Python 2.7.1+
```

3.2.2 VirtualBox

Cuckoo Sandbox makes extensive use of VirtualBox and all the features and capabilities that come with it.

Even if you will most likely find VirtualBox already packaged with your GNU/Linux distribution, it's preferable for you to download the latest version directly from the official website and choose an appropriate package.

VirtualBox comes with dedicated packages for the most popular GNU/Linux distributions, you can get yours at official VirtualBox website: `https://www.virtualbox.org/wiki/Linux_Downloads`.

3.3 Setup

If you downloaded the package from the Cuckoo website, as described in How to get Cuckoo, extract it in a path of your choice (in this guide we'll use /opt/cuckoo/).

If you cloned the Git repository, you'd probably want to copy it to another target path and keep the Git folder for fetching updates, in order to not rewrite potential critical files on your running setup.

3.4 Configuration

In order to adapt Cuckoo to your needs, you need to edit its configuration file.

Despite being extensively commented, we're going through all of its sections and see available options.

3.4.1 Logging

The first section defines Cuckoo's logging capabilities, including:

- The path to the main log file.
- The possibility to use UTC timezone for logging events.
- The possibility to enable or disable additional debug messages (useful for error handling).

Here is an example:

```
[Logging]
# Path to Cuckoo's log file.
path = log/cuckoo.log
# Set to "True" if you want to use UTC time for logging, turn to "False" if you
# want to use local time logging.
utc = False
# Enable/Disable additional debugging messages. This messages won't wrote to
# log file but just printed on screen.
debug = False
```

3.4.2 Analysis

The second section defines Cuckoo's general analysis behavior. Here you can define timeouts, path to the results and to the post-processing script (which is a custom Python script you can use to access each analysis' results and handle the produced data).

```
[Analysis]
# This is the actual analysis timeout (expressed in seconds). This represents
# the default timeout performed by analysis core if none is specified.
analysis_timeout = 120
# Watchdog timeout (expressed in seconds) for analysis execution to complete,
# when this timeout gets hit, current execution is aborted and virtual machine
# is restored and freed.
watchdog_timeout = 600
# Specify here the path where analysis results shall be stored.
results_path = analysis/
# Specify here the path to the postprocessing script.
postprocessing = processor.py
```

3.4.3 Database

The third section contains the path to the local SQLite database that Cuckoo Sandbox uses to handle pending tasks queue. You generally shouldn't edit this.

```
[LocalDatabase]
file = db/cuckoo.db
```

3.4.4 Sniffer

This section contains details on network traffic monitoring.

By default it disables use of the external sniffer (tcpdump), but in case you need it, turn it on and specify the proper Host's network interface to monitor. The sniffer will automatically filter traffic based on the current virtual machine's MAC address.

```
[Sniffer]
# Enable or disable the following option by assigning a True or False value.
# In case you decide to disable it, you're supposed to either not have any
# network dump or to used VirtualBox's (or any other virtualization engine
# you are using) to handle the network monitoring instead of using an external
# sniffer such as tcpdump.
sniffer = False
# This specifies the network interface where the sniffer will bind to in order
# to monitor virtual machines' generated traffic.
interface = eth0
```

3.4.5 Virtual Machines

This section contains information on the virtual machines that Cuckoo has at its disposal.

It's very important to properly list the available virtual machines in the "enabled" option, by listing IDs you're going to assign to each of them separated by commas.

Through the "mode" option you can choose between "headless" (generally used for servers) or "gui" (generally used for desktop setup, enabling the creation of the graphical interface of the virtual machine).

Make sure also to specify the exact path on Guest machines where you're going to install the Python interpreter for Windows.

```
[VirtualMachines]
# Virtualization product.
engine = VirtualBox
# List virtual machines IDs separated by commas.
enabled = cuckoo1
# Set to "gui" if you want Cuckoo to spawn virtual machines' GUIs or set to
# "headless" if you don't.
```

```
mode = headless
# Path to local Python installation on guest machines. Please be sure to have
# correctly set this value as it's critical to Cuckoo's proper execution.
python = C:\Python27\python.exe
```

3.4.6 Virtual Machine Details

For every machine you listed in the previous section, you should have a corresponding detailed section like the following:

```
[cuckoo1]
name = Cuckoo 1
username = Me
password = cuckoo
# Please notice that the shared folder name must coincide with the current
# virtual machine id, which is the name you assigned between the square
# brackets (e.g. [cuckoo1]).
share = shares/cuckoo1
```

Here you can define:

- Virtual machine's name in VirtualBox (which can obviously be different than the ID you assigned in Cuckoo's config file).
- The username of the Windows account you created on Guest.
- The password for such account.
- The path to its unique shared folder (IMPORTANT: the name of the shared folder must equals the ID you assigned to current virtual machine).

3.5 Preparing the Guest

Once you've installed VirtualBox you can proceed by creating your first virtual machine: this can easily be done through the Graphical User Interface or through the powerful command line utility "VBoxManage".

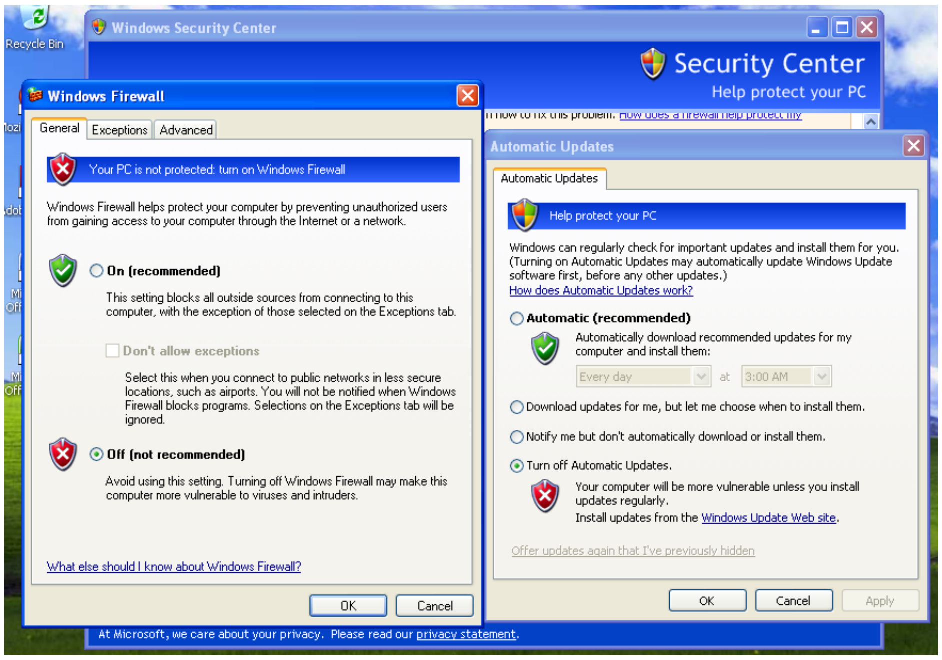
Since VirtualBox basic usage is beyond the scope of this guide, if you're not familiar with it, you're invited to first go through its official documentation at `https://www.virtualbox.org/wiki/Documentation`.

Cuckoo Sandbox doesn't require any particular configuration at virtual machine's creation time, so you can just go straight following the instructions and providing default values.

3.5.1 Windows XP

In order to make Cuckoo work properly, you need to perform some changes on your Windows XP guest environment.

1. Download (from official website at `http://python.org/download/`) and install Windows Python 2.7 interpreter.
2. Install VirtualBox Guest Additions.
3. Set a password to your current user.
4. If you want to have screenshots of Windows desktop during malware's execution, you need to install Python Imaging Library which is available from `http://www.pythonware.com/products/pil/`. Make sure to select the installer corresponding to your Python version.
5. Proceed by disabling Windows Firewall and Automatic Updates from Control Panel - Security Center, as shown in the picture:



Your PC is not protected: turn on Windows Firewall

Windows Firewall helps protect your computer by preventing unauthorized users from gaining access to your computer through the Internet or a network.

On (recommended)

This setting blocks all outside sources from connecting to this computer, with the exception of those selected on the Exceptions tab.

Don't allow exceptions

Select this when you connect to public networks in less secure locations, such as airports. You will not be notified when Windows Firewall blocks programs. Selections on the Exceptions tab will be ignored.

[✓ **Off (not recommended)**]

Avoid using this setting. Turning off Windows Firewall may make this computer more vulnerable to viruses and intruders.

What else should I know about Windows Firewall?

Help protect your PC

Windows can regularly check for important updates and install them for you. (Turning on Automatic Updates may automatically update Windows Update software first, before any other updates.)
How does Automatic Updates work?

Automatic (recommended)

Automatically download recommended updates for my computer and install them:

Every day at 3:00 AM

Download updates for me, but let me choose when to install them.

Notify me but don't automatically download or install them.

[✓ Turn off Automatic Updates.]

Your computer will be more vulnerable unless you install updates regularly.
Install updates from the Windows Update Web site.

Offer updates again that I've previously hidden

At Microsoft, we care about your privacy. Please read our privacy statement.

Figure 2: Disabling Windows Firewall and Automatic Updates.

This is not mandatory, but helpful to avoid malware execution being blocked and to not have the network traffic dump polluted with updates downloads.

Proceed with installing any other software you might want available in your sandboxed environment.

For example you might want to install:

- Mozilla Firefox

- Adobe Reader
- Adobe Flash Player
- Microsoft Office
- PHP
- Perl

Remember to disable auto-updates and update-checking features for each software you want to install.

Please consider that it's up to you on how to configure your sandboxed environment and which versions of such softwares to install, consequently which kind of exploits and malware you'll be able to analyze.

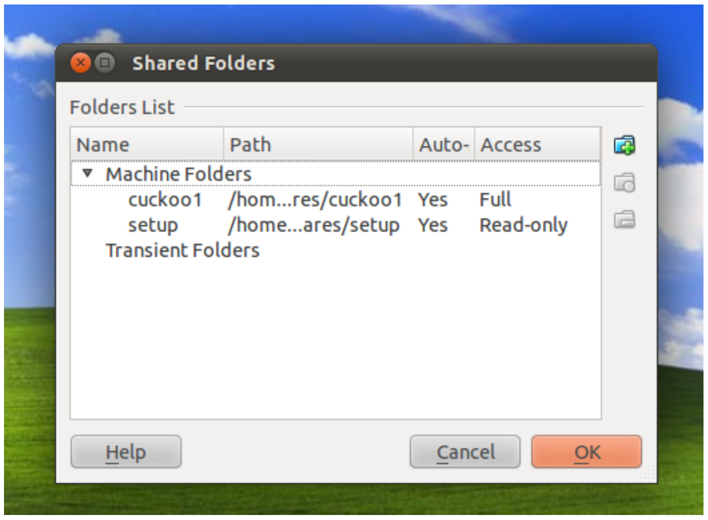
3.5.2 Shared Folders

Cuckoo Sandbox uses VirtualBox's Shared Folders to exchange data between the Host and the Guest.

Generally speaking, a Cuckoo virtual machine should be provided with two shared folders:

1. A "setup" folder used to get required programs and scripts.
2. A unique folder used to drop the analysis results.

In order to do so, open VirtualBox's shared folders configuration:



Name	Path	Auto-	Access
▼ Machine Folders			
cuckoo1	/hom...res/cuckoo1	Yes	Full
setup	/home...ares/setup	Yes	Read-only
Transient Folders			

Figure 3: Network shares configuration.

Make sure to add the two folders with the proper privileges:

1. A "setup" folder with read-only, auto-mount, permanent.
2. A unique folder (in the example being "cuckoo1") full access, auto-mount, permanent.

3.5.3 Network Configuration

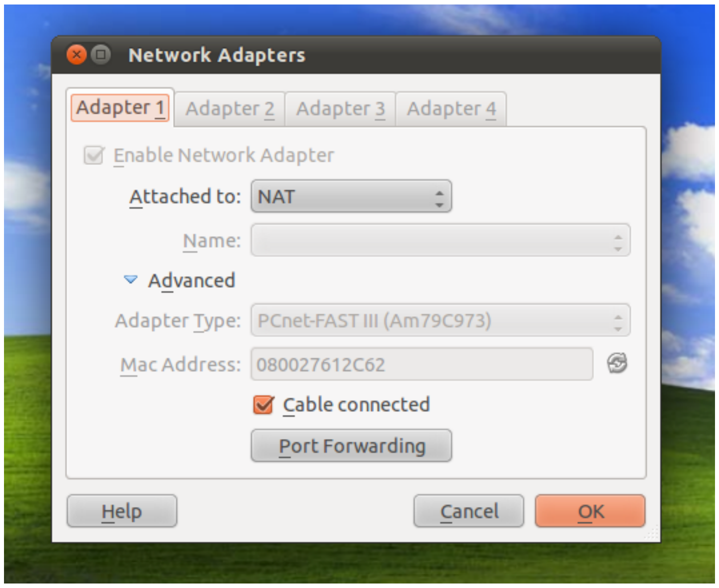
Now you have to decide how to proceed configuring the network for your virtual machine.

The two easiest choices are:

1. Use NAT and VirtualBox's embedded network trace functionality (suggested).
2. Use Bridged Network and adopt an external sniffer (i.e. tcpdump) to run on the host machine.

3.5.4 NAT Networking

In case you decided to go for NAT, just open the network configuration dialog of your virtual machine and choose NAT for your first adapter, as shown in figure:



Enable Network Adapter
Attached to: NAT
Name:
Adapter Type: PCnet-FAST III (Am79C973)
Mac Address: 080027612C62
Cable connected

Figure 4: Networking configuration.

Now, in order to modify virtual machine's configuration, power it off as you would normally do with a common Windows machine.

Open a terminal on your host machine and issue the following command:

```
$ VBoxManage modifyvm "Name of VM" --nictrace1 on \
--nictracefile1 /opt/cuckoo/shares/cuckoo1/dump.pcap
```

Specify the correct virtual machine's label name and the correct path to its own Cuckoo shared folder (it's importat to specify "dump.pcap" appended to such path).

3.5.5 Bridged Networking

The bridged network is a good alternative to NAT and allows you to use an external sniffer instead of VirtualBox's NIC trace.

As you would do configuring NAT, just open the network configuration dialog of your virtual machine and choose Bridged Network, select the host's network interface and press OK.

On the host, go to Cuckoo's root, open "cuckoo.conf" and edit the following section:

```
[Sniffer]
sniffer = True
interface = eth0
```

Be sure to choose the appropriate network interface.

3.5.6 Snapshot

After you completed all the previous steps, you can proceed taking a snapshot of the clean virtual machine.

Launch your virtual machine, wait for Windows to boot, log in and wait for the operating system to finish its startup.

Now you can take a snapshot.

You can do so through VirtualBox's GUI or through the command line with:

```
$ VBoxManage snapshot "Name of VM" take "Name of Snapshot"
```

Once the snapshot is created, you can poweroff the virtual machine and restore it with the following commands:

```
$ VBoxManage controlvm "Name of VM" poweroff
$ VBoxManage snapshot "Name of VM" restorecurrent
```

4 Usage

Cuckoo can be simply launched (preferably in a screen shell) with:

```
$ python cuckoo.py
```

Once started, this is similar to what you're going to see:

```
                     _
    ____ _   _  ____| |  _ ___   ___
   / ___) | | |/ ___) |_/ ) _ \ / _ \
  ( (___| |_| ( (___|  _ ( |_| | |_| |
   \____)____/ \____)_| \_)___/ \___/ v0.2

 www.cuckoobox.org
 Copyright (C) 2010-2011
 by Claudio "nex" Guarnieri

[2011-11-02 00:00:00] [Virtual Machine] [Check] Your VirtualBox version is:
                      "4.1.2", good!
[2011-11-02 00:00:00] [Start Up] Populating virtual machines pool...
[2011-11-02 00:00:00] [Virtual Machine] Acquired virtual machine with name
                      "Cuckoo_1".
[2011-11-02 00:00:00] [Virtual Machine] [Infos] Virtual machine "Cuckoo_1"
                      informations:
[2011-11-02 00:00:00]   \_| Name: Cuckoo_1
[2011-11-02 00:00:00]     | ID: xxxxxxx-xxxx-xxxx
[2011-11-02 00:00:00]     | CPU Count: 1 Core/s
[2011-11-02 00:00:00]     | Memory Size: 192 MB
[2011-11-02 00:00:00]     | VRAM Size: 16 MB
[2011-11-02 00:00:00]     | State: Saved
[2011-11-02 00:00:00]     | Current Snapshot: "Clean & Running"
[2011-11-02 00:00:00]     | MAC Address: 08:00:27:03:13:37
[2011-11-02 00:00:00] [Start Up] 1 virtual machine/s added to pool.
[2011-11-02 00:00:00] [Database] [Init] Generated database "db/cuckoo.db"
                      which didn't exist before.
```

As you can see, at first run Cuckoo generates a SQLite database file located at db/cuckoo.db. This is used to manage the analysis tasks queue, therefore your malware submissions end up there and you should directly interface your feeds to it.

For manual submissions there's a simple script you can use:

```
$ python submit.py --help
Usage: submit.py [options] filepath
Options:
  -h, --help                           show this help message and exit
  -t TIMEOUT, --timeout=TIMEOUT       Specify analysis execution time limit
  -p PACKAGE, --package=PACKAGE       Specify custom analysis package name
  -r PRIORITY, --priority=PRIORITY    Specify an analysis priority expressed
                                      in integer
  -c CUSTOM, --custom=CUSTOM          Specify any custom value to be passed
                                      to postprocessing
```

For example, in order to submit an executable you could simply issue:

```
$ python submit.py /path/to/binary.exe --priority=5 --timeout=100
```

Once added, Cuckoo will start analyzing the submitted file as displayed by its logs:

```
[2011-11-02 00:00:10] [Core] [Dispatcher] Acquired analysis task for
                      target "/path/to/binary.exe".
[2011-11-02 00:00:10] [Database] [Lock] Locked task with id 1.
[2011-11-02 00:00:10] (Task #1) [Analysis] [Generate Config] Config file
                      successfully generated at "shares/cuckoo1/analysis.conf".
[2011-11-02 00:00:10] [Virtual Machine] Acquired virtual machine with name
                      "Cuckoo 1".
[2011-11-02 00:00:13] [Virtual Machine] [Start] Virtual machine "Cuckoo 1"
                      starting in "headless" mode.
[2011-11-02 00:00:14] [Virtual Machine] [Execute] Cuckoo executing with
                      PID 1560 on virtual machine "Cuckoo 1".
[2011-11-02 00:01:21] [Virtual Machine] [Execute] Cuckoo exited with
                      code 0 on virtual machine "Cuckoo 1".
[2011-11-02 00:01:21] [Virtual Machine] [Stop] Virtual machine "Cuckoo 1"
                      powered off successfully.
[2011-11-02 00:01:27] [Virtual Machine] [Restore] Virtual machine "Cuckoo 1"
                      successfully restored to current snapshot.
[2011-11-02 00:01:28] (Task #1) [Analysis] [Save Results] Analysis results
                      successfully saved to "analysis/1".
[2011-11-02 00:01:28] (Task #1) [Analysis] [Clean Share] Shared folder
                      "shares/cuckoo1" cleaned successfully.
[2011-11-02 00:01:28] [Database] [Complete] Task with id 1 updated in
                      the database with status "1".
[2011-11-02 00:01:28] (Task #1) [Analysis] [Free VM] Virtual machine
                      "cuckoo1" released.
[2011-11-02 00:01:28] (Task #1) [Analysis] [Core] Postprocessing script
                      started with pid "6845".
```

The results of the analysis are then stored under analysis/X where X is the numeric ID of the task in the SQLite database, and they will contain:

- analysis.conf: the config file used to perform the analysis.
- analysis.log: a log file generated by Guest's Cuckoo component.

- dump.pcap: the generated network traffic dump.
- files/: the directory containing all dumped files.
- logs/: the raw analysis log files generated by Cuckoo.
- report.txt: an example of text report generated by the default post-processing script out of the raw logs.
- shots/: all the screenshots taken during the analysis.

4.1 Feedback and bug reports

If you want to report bug findings on the software, you're invited to either use our official GitHub repository's ticketing system at `https://github.com/cuckoobox/cuckoo` or report on the official forums located at `http://forums.malwr.com`.

Please remember that we can also be reached on IRC at irc.freenode.net #cuckoobox.

5 Copyright and license

5.1 License

5.2 Disclaimer

6 FAQ

6.1 General questions

6.1.1 Can I redistribute Cuckoo Sandbox?

Yes, you can. Cuckoo Sandbox is distributed under the GNU General Public License version 3. See License.

6.1.2 Can I include Cuckoo Sandbox in my closed source commercial product?

Generally no, you can't. Cuckoo Sandbox is distributed under the GNU General Public License version 3. See License.

6.1.3 I want to help Cuckoo, what can I do?

Your help is very appreciated, you can help Cuckoo Sandbox in several ways, from coding to send bug reports. See Feedback and bug reports and Feedback questions.

6.2 Usage questions

6.2.1 How to start an analysis?

You can simply start an analysis via command line. Check Usage.

6.2.2 How to change Cuckoo default behaviour?

You can edit the Cuckoo configuration file. Check Configuration.

6.3 Feedback questions

6.3.1 I found a bug or I want to suggest some features

Your help is really appreciated to improve Cuckoo Sandbox. Take a look at Feedback questions.

6.3.2 I want to help but I haven't time

There are many ways to help Cuckoo: coding, testing, reporting bugs, donating money or hardware, reviewing code and documentation or submitting feature requests or feedback.

To do this you don't need to spend a lot of time. Simply reading this guide and submitting bugs related to this is a great help.

7 Authors

Claudio “nex” Guarnieri – Lead Developer – nex@cuckoobox.org

Dario Fernandes – Contributor – dariosfernandes@gmail.com

Alessandro “jekil” Tanasi – Contributor – alessandro@lonerunners.net