M.Tech Cybersecurity (MIT)

School of Computer Engineering, MIT

Cuckoo Sandbox

Comprehensive Configuration, Modules, and Installation Guide

Submitted By:

Yadunandan N - 251091010013

MadhavJee - 251091010023

M.Tech – Cybersecurity

Under the Guidance of:

Dr. Manoj T

Assistant Professor

School of Computer Engineering, MIT

Date: November 4, 2025

Contents

1. Introduction

Cuckoo Sandbox is an open-source automated malware analysis system that runs suspicious files in an isolated virtualized environment (sandbox). It observes and reports behavioral patterns such as network activity, file system modifications, and process creation.

This documentation serves as a professional README and configuration manual for setting up and customizing a Cuckoo Sandbox instance. It includes configuration files, scripts, and example modules to extend analysis capabilities.

2. Repository Structure

The repository is organized to support modular customization and easy deployment.

File / Directory	Description
README.md	Overview and usage guide
install.sh	Automated installation script
conf/cuckoo.conf	Core configuration file
modules/processing/custom_analysis.py	Custom analysis module
modules/signatures/custom_signature.py	Custom malware signature
scripts/setup_db.sh	Database initialization script
examples/tasks/sample_task.json	Example task submission file

Table 2.1: Repository Contents

3. Installation and Setup

3.1 System Requirements

- Ubuntu 20.04 or Debian-based OS
- Python 3.10+
- Virtual environment (`venv` or `virtualenv`)
- SQLite or PostgreSQL database

3.2 Installation Script

The installation script automates environment preparation.

Listing 3.1: install.sh

```
#!/usr/bin/env bash
set -euo pipefail

echo "Updating package lists..."
sudo apt update
sudo apt install -y python3 python3-venv python3-pip git libffi-dev \
libssl-dev sqlite3 build-essential

PROJECT_DIR="$HOME/cuckoo-local"
mkdir -p "$PROJECT_DIR" && cd "$PROJECT_DIR"

python3 -m venv venv
source venv/bin/activate
pip install -U pip setuptools wheel
pip install requests flask sqlalchemy yara-python pefile pydeep python-magic

echo "Clone your preferred Cuckoo fork into $PROJECT_DIR/cuckoo"
echo "git clone https://github.com/cuckoosandbox/cuckoo.git cuckoo"
```

4. Configuration File

The `cuckoo.conf` file defines local system settings, storage paths, enabled modules, and reporting options.

Listing 4.1: conf/cuckoo.conf

```
[local]
host = 127.0.0.1
storage_path = ./storage

[database]
uri = sqlite:///./cuckoo.db

[processing]
enabled = pe,behavior,network,custom_analysis

[reporting]
enabled = json,file

[logging]
level = INFO
```

5. Custom Analysis Module

```
"""
Custom post-processing module for analyzing network and behavior logs.
"""

import os, json

MODULE_NAME = "custom_analysis"

def run(results_path):
    summary = {"module": MODULE_NAME, "files": []}
    net_file = os.path.join(results_path, "network.json")

    if os.path.exists(net_file):
        try:
            with open(net_file, "r", encoding="utf-8") as f:
                net = json.load(f)
            summary["files"].append({"network_events": len(net.get("
   connections", []))})
        except Exception as e:
            summary["error"] = str(e)

    output_path = os.path.join(results_path, "custom_analysis.summary.json")
    with open(output_path, "w", encoding="utf-8") as f:
        json.dump(summary, f, indent=2)
    return summary
```

6. Custom Signature Module

```
"""
Simple signature for detecting keyboard hook usage in malware.
"""

def check_keyboard_hooks(results):
    calls = results.get("api_calls", [])
    for call in calls:
        if call.get("api") == "SetWindowsHookExA":
            return True
    return False

def run(results):
    if check_keyboard_hooks(results):
        return {"name": "keyboard_hook", "malicious": True,
                "desc": "Detected SetWindowsHookExA usage."}
    return {"name": "keyboard_hook", "malicious": False}
```

7. Database Setup

```
#!/usr/bin/env bash
DBFILE="./cuckoo.db"
if [ ! -f "$DBFILE" ]; then
  echo "Creating database file..."
  sqlite3 "$DBFILE" "VACUUM;"
else
  echo "Database already exists."
fi
```

8. Sample Task File

Listing 8.1: sample.json

```
{
  "package": "exe",
  "timeout": 120,
  "options": {"enforce_timeout": true},
  "file_path": "/path/to/sample.exe",
  "tags": ["example", "test"]
}
```

9. Running Cuckoo Sandbox

1. Activate the virtual environment: `source venv/bin/activate`
2. Start Cuckoo: `cuckoo -d`
3. Submit a file: `cuckoo submit /path/to/sample.exe`
4. View reports in `storage/analyses/`

10. Conclusion

This repository provides a modular foundation for deploying and extending Cuckoo Sandbox. It supports the creation of custom analysis modules, signature extensions, and automated setups suitable for malware behavior research and cybersecurity labs.