# Cuckoo Sandbox 3.0

certEE

Hatching™

Co-financed by the Connecting Europe Facility of the European Union

# What is Cuckoo 3?

Complete rewrite of Cuckoo sandbox in Python 3, with a focus on:

- Improved maintainability
- Support for Windows 10 analysis
- Easy integration with MISP and IntelMQ platforms
- Increased performance and dependencies updated/rewritten where necessary
- Greater support for scalability
- Open source code

Co-financed by the Connecting Europe Facility of the European Union

# Current Project Status

Core features of Cuckoo 3 in place and working:

- Machinery to manage virtual machines, including multi-environment support
- New kernel-mode agent
- Signature mechanism and updated analysis scoring
- Functional web interface and API for interaction
- Modular static analysis support
- Search engine for IoC filtering
- Mitre ATT&CK support

Current development focus on improving reporting of data to the user, including:

- UI improvements
- Porting/updating signatures

certbee

Hatching™

Co-financed by the Connecting Europe Facility of the European Union

# Cuckoo 2 vs Cuckoo 3

**Kernel-mode vs User-mode agent**

New Cuckoo uses new Threemon kernel agent instead of Cuckoo 2's user-mode agent

Allows more advanced behavioural monitoring and improved resistance to anti-analysis techniques

Enables memory/process dumping

Logs all relevant system actions to be passed to the reporting phase for processing with signatures

- All 'standard' IoCs such as file and registry actions, network activity, new processes etc.
- Also logs suspicious system activity and API calls related to known-malicious behaviour
  - Debugger checks, process injection techniques, loading drivers, executing dropped file etc.

Co-financed by the Connecting Europe Facility of the European Union

# Cuckoo 2 vs Cuckoo 3

## Updated signature engine

- Yaml-style signature format
- Support for file, registry, command-line and mutex events
- Automatic normalization of some file/registry paths to ease signature writing
- Support for Mitre ATT&CK v8

```
name: antivm_virtualbox
signatures:
  registry_antivm_virtualbox:
    short_description: Detects VirtualBox through registry keys
    description: Enumerates registry keys generally present in VirtualBox VMs. Common anti-VM/anti-analysis technique.
    score: 7
    ttps:
      - T1497.001
      - T1012
    triggers:
      - registry read:
        - ^(HKLM|HKCU)\\SOFTWARE\\Oracle\\VirtualBox Guest Additions
      - registry read:
        - ^(HKLM|HKCU)\\SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Uninstall\\Oracle VM VirtualBox Guest Additions
      - registry read:
        - ^(HKLM|HKCU)\\ControlSet001\\Services\\VBox*
      - registry read:
        - ^(HKLM|HKCU)\\SYSTEM\\ControlSet001\\Enum\\PCI\\VEN_80EE&DEV_(BEEF|CAFE)&SUBSYS_00000000&REV_00
      - registry read:
        - ^(HKLM|HKCU)\\SYSTEM\\ControlSet001\\Control\\VirtualDeviceDrivers
      - registry read:
        - ^(HKLM|HKCU)\\HARDWARE\\ACPI\\(DSDT|FADT|RSDT)\\VBOX__.*
```

certSEE

Hatching™

Co-financed by the Connecting Europe Facility of the European Union

# Cuckoo 2 vs Cuckoo 3

## Updated signature engine

Manual API hooks in Cuckoo 2 replaced with predefined events reported by Threemon agent

Includes direct API hooks, e.g.

- NtCreateThreadExHideFromDebugger
- SetWindowsHookAW

Also includes known bad actions which include multiple events, e.g. process injection, loads dropped DLL, etc.

```
18  class VBoxDetectProvname(Signature):
19      name = "antivm_vbox_provname"
20      description = "Detects VirtualBox using WNetGetProviderName trick"
21      severity = 3
22      categories = ["anti-vm"]
23      authors = ["Optiv"]
24      minimum = "2.0"
25      evented = True
26
27      filter_apinames = "WNetGetProviderNameW",
28
29      def on_call(self, call, process):
30          if call["arguments"]["net_type"] == "0x00250000":
31              self.mark_call()
32
33          return self.has_marks()
```

Example Cuckoo 2 API hook

Co-financed by the Connecting Europe Facility of the European Union

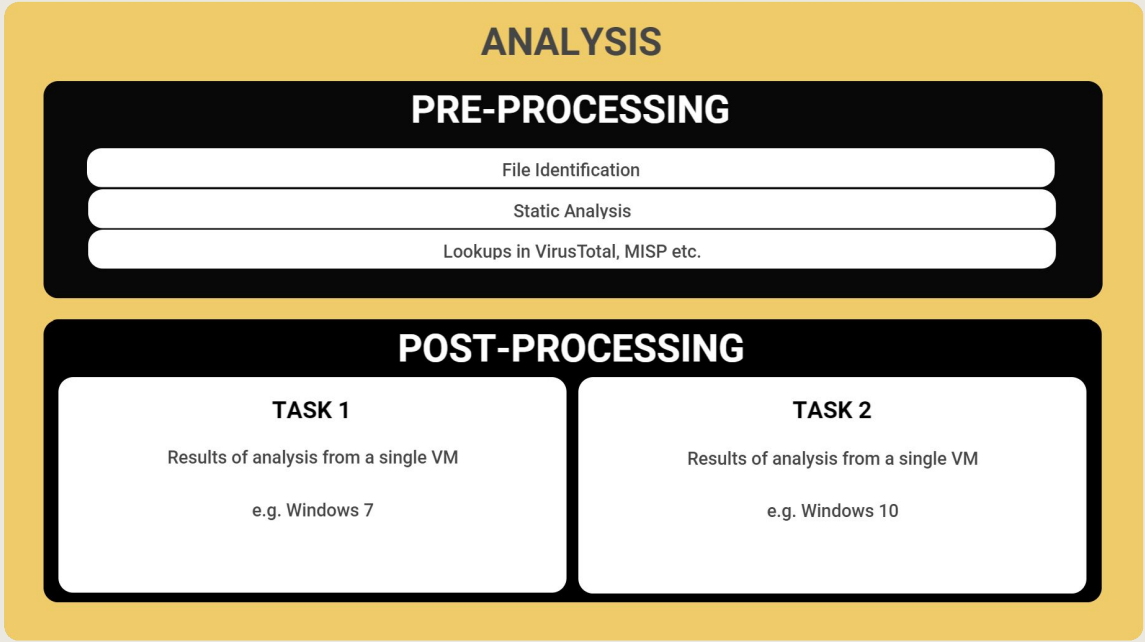
# Cuckoo 2 vs Cuckoo 3

## New analysis structure

Cuckoo 2 treated each VM task as an individual analysis

Cuckoo 3 can run multiple behavioural tasks from a single submission, supporting analysis on different platforms simultaneously

Static and other pre-processing actions are made once per sample before starting behavioural analysis




Co-financed by the Connecting Europe Facility of the European Union

# Cuckoo 2 vs Cuckoo 3

**Reliable file type identification**

Cuckoo 2 only performed automatic file identification for submissions made through the web UI or specific API endpoints

Cuckoo 3 carries this out for all submissions, API or UI. Helps with:

- Smoother automation of sample submission
- Reduces confusion and errors for users

certee

Hatching™

Co-financed by the Connecting Europe Facility of the European Union

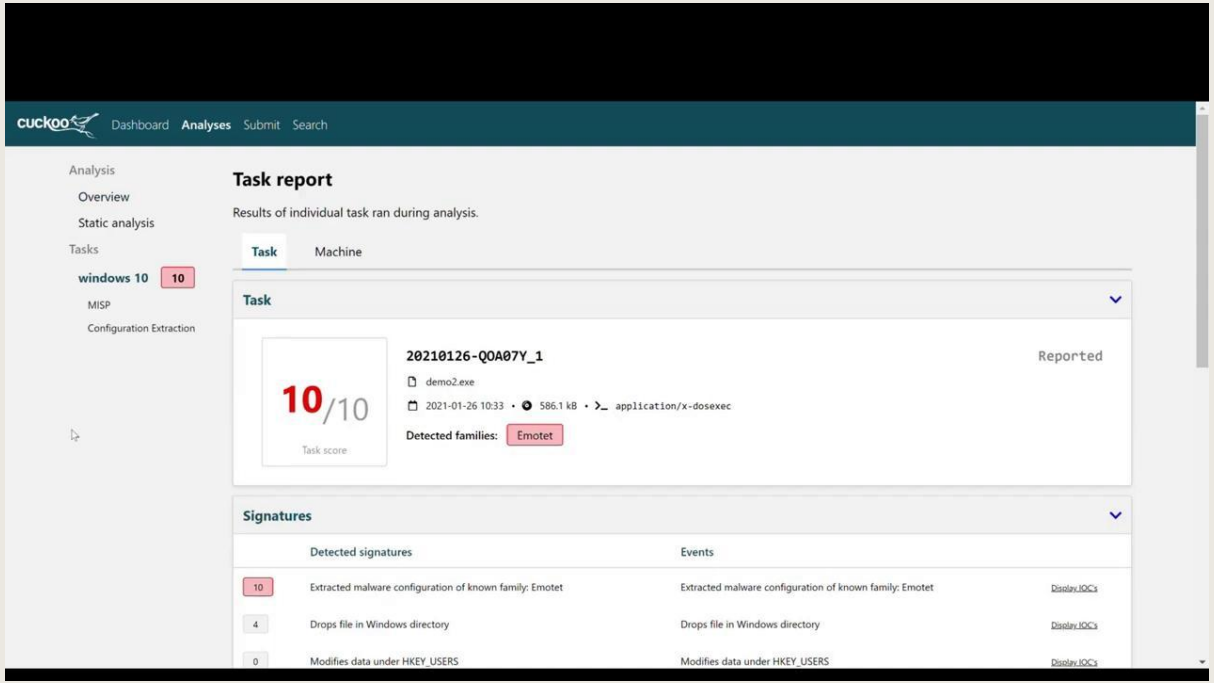
# Demos

certhee

Hatching™

Co-financed by the Connecting Europe Facility of the European Union

# Demos




certbee

Hatching™

Co-financed by the Connecting Europe Facility of the European Union

# Next Steps

Focus is now on the reporting mechanisms:

- Extending signatures beyond current baseline
- Improving report view in the web interface
- Multiple trainings in April
- Distributed analysis
  - Host VMs across multiple servers managed by a single scheduler

certee

Hatching™

Co-financed by the Connecting Europe Facility of the European Union

# Questions?

Reach us at:

**info@hatching.io**

Thanks for listening!

certbee

Hatching™

Co-financed by the Connecting Europe Facility of the European Union