



Cuckoo != Insanity

Why a little Cuckoo should be part of your
Incident Response Process

Chris Henderson

- ❖ @enruhe_
- ❖ Security Engineer
- ❖ Arctic Wolf Networks
- ❖ Focus on Blue Team for Small to Medium size companies
- ❖ GNFA - GCIH - GCIA - GSEC



Projects

- OpenVAS
- UserAgent String Baselining
- The Hive
 - Incident Response Platform
- [https://arcticwolf.com/
company/careers/](https://arcticwolf.com/company/careers/)

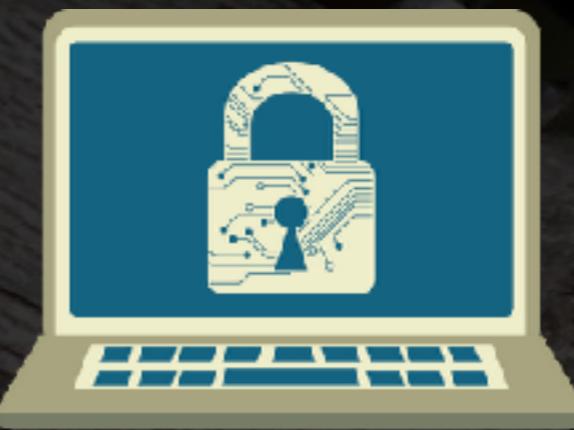


Community - CFP



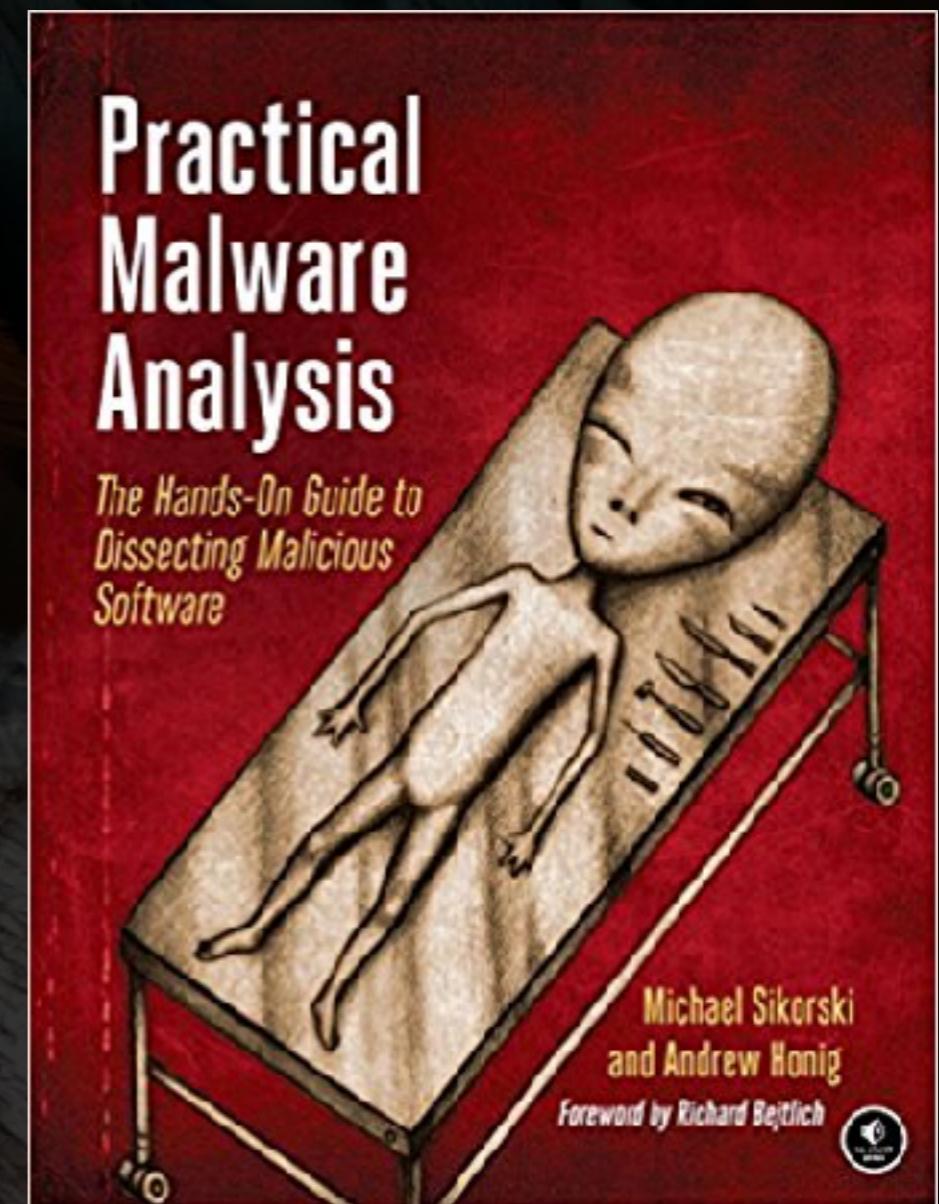
HACKWEST™

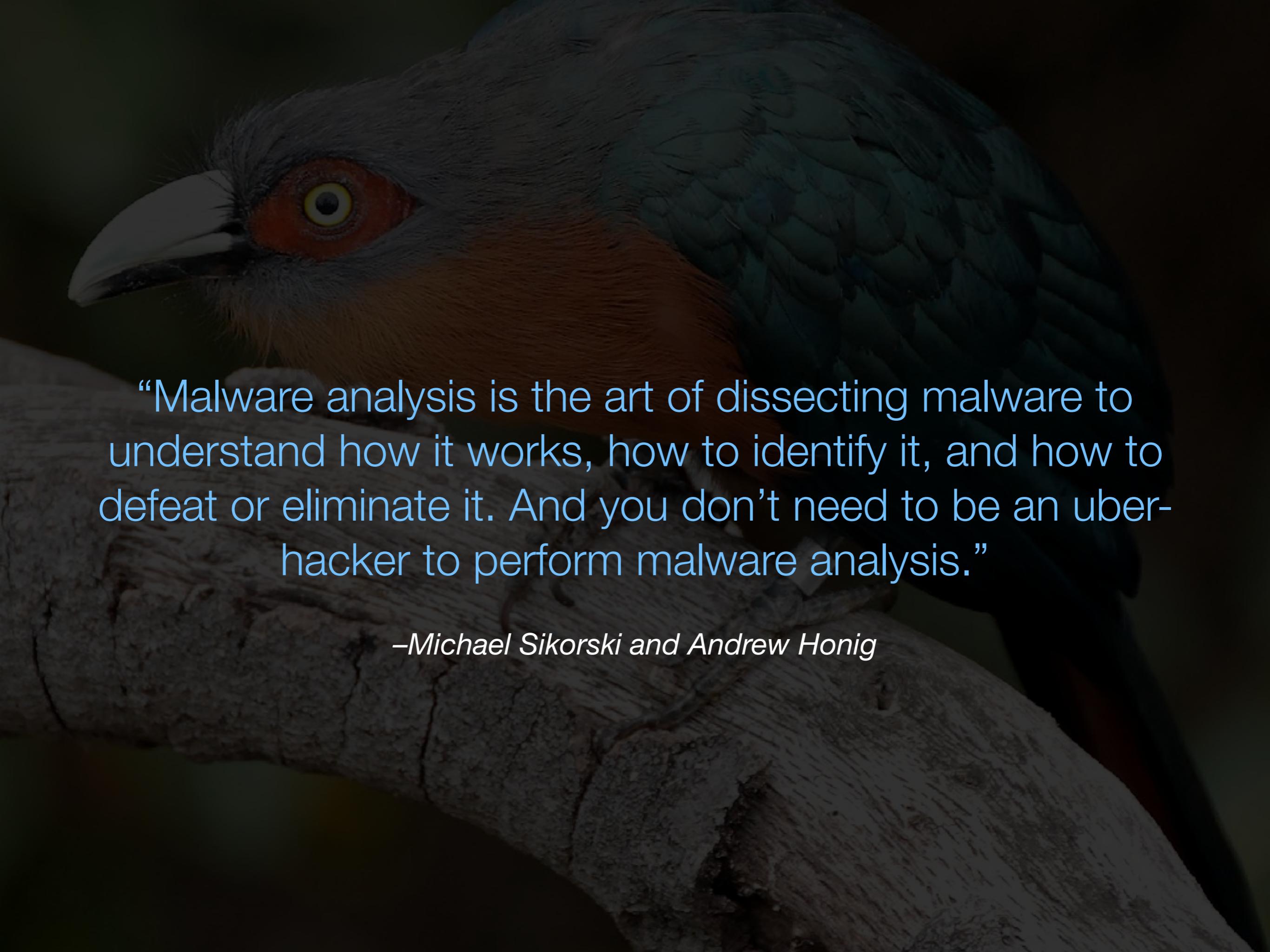
SALT LAKE CITY
BSIDES



Malware

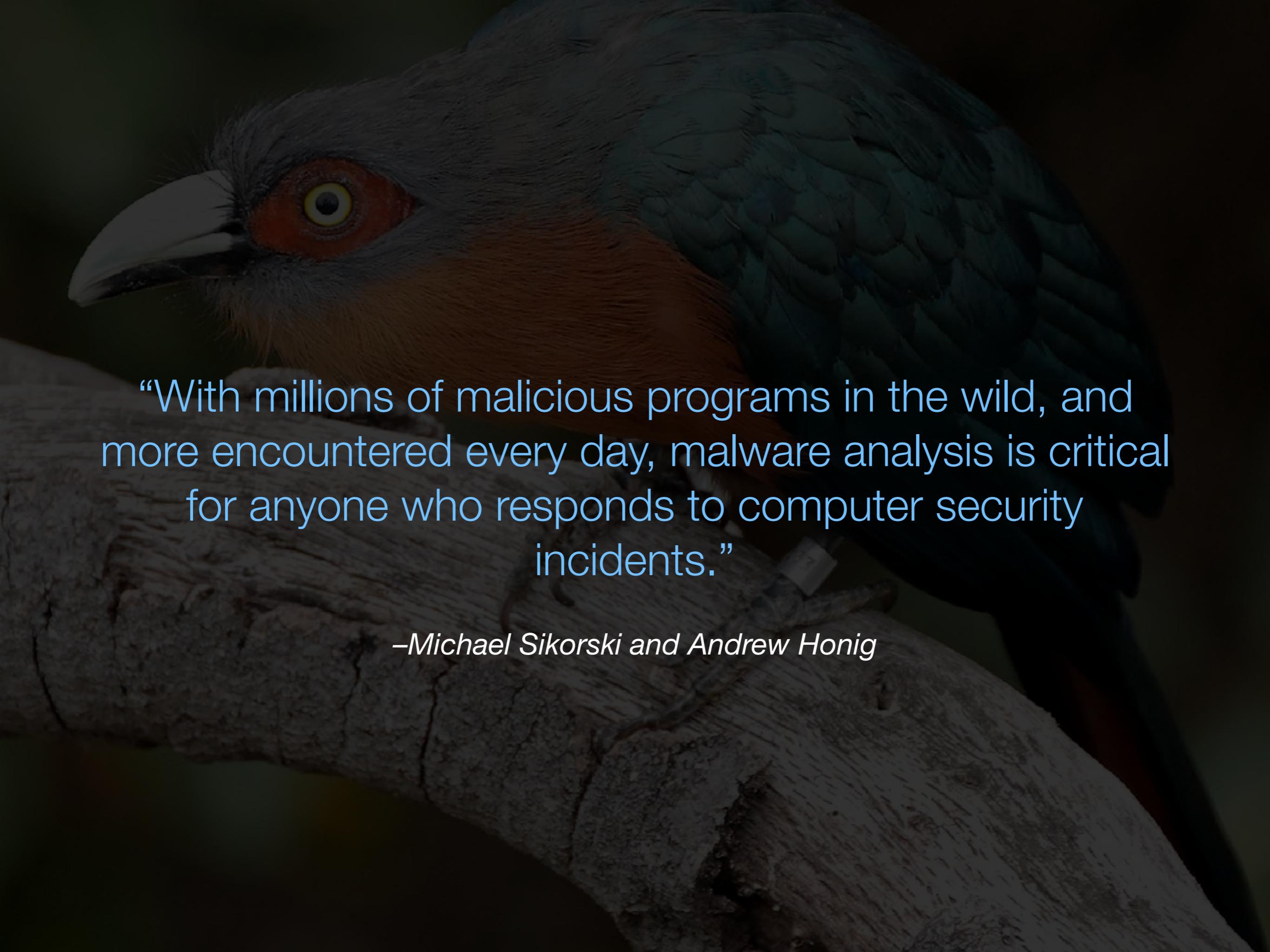
Any software that does something that causes harm to a user, computer, or network can be considered malware, including viruses, trojan horses, worms, rootlets, scareware and spyware.





“Malware analysis is the art of dissecting malware to understand how it works, how to identify it, and how to defeat or eliminate it. And you don’t need to be an uber-hacker to perform malware analysis.”

–Michael Sikorski and Andrew Honig



“With millions of malicious programs in the wild, and more encountered every day, malware analysis is critical for anyone who responds to computer security incidents.”

–Michael Sikorski and Andrew Honig

Analysis Techniques

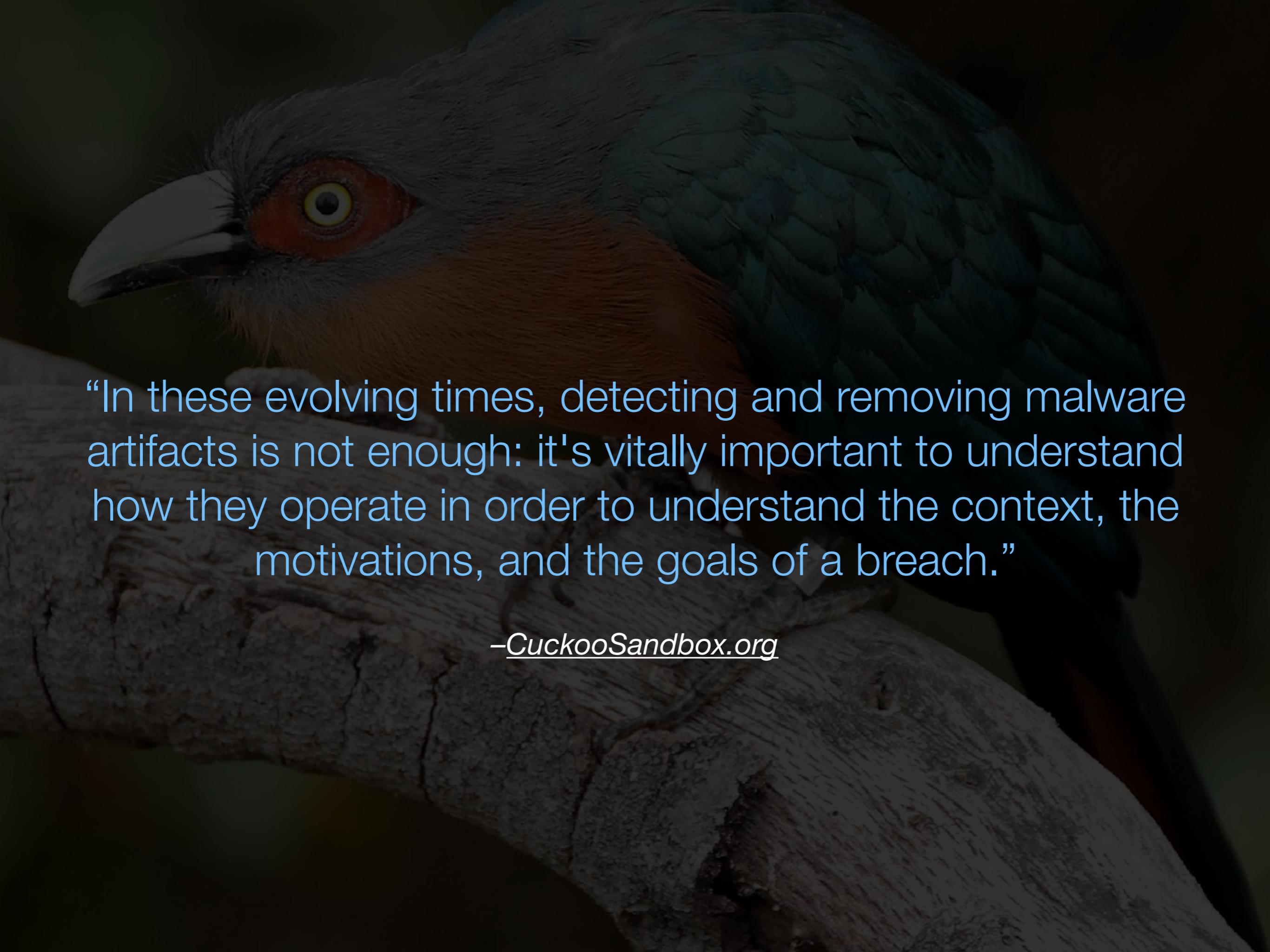


- ***Static Analysis*** - involves examining the malware without running it.
- ***Dynamic Analysis*** - involves executing the malware



- ❖ Began as a Google Summer of Code Project in 2010
- ❖ Part of the Honeynet Project
 - ❖ <http://www.honeynet.org/>
- ❖ Originally designed by Claudio Guarnieri
 - ❖ @botherder
- ❖ Open Source





“In these evolving times, detecting and removing malware artifacts is not enough: it's vitally important to understand how they operate in order to understand the context, the motivations, and the goals of a breach.”

-CuckooSandbox.org



a **sandbox** is a security mechanism for separating running programs, usually in an effort to mitigate system failures or software vulnerabilities from spreading. It is often used to execute untested or untrusted programs or code, possibly from unverified or untrusted third parties, suppliers, users or websites, without risking harm to the host machine or operating system.



Static Analysis

- File Information
 - Size, Type, MD5, SHA1, SHA256, SHA512, CRC32
- Version Info
- Strings Data



Dynamic Analysis

- ❖ Extracted Artifacts
- ❖ Behavioral Analysis
- ❖ Network Analysis
- ❖ Dropped Files
- ❖ Dropped Buffers
- ❖ Process Memory





You can throw any suspicious file at Cuckoo and in a matter of minutes Cuckoo will provide a detailed report outlining the behavior of the file when executed inside a realistic but isolated environment.



Cuckoo Sandbox is free software that automated the task of analyzing any malicious file under **Windows**, **OS X**, **Linux**, and **Android**.

Use Cases

- ❖ Generic Windows Executables
- ❖ DLL Files
- ❖ PDF Documents
- ❖ Microsoft Office Documents
- ❖ PHP Scripts
- ❖ Zip Files
- ❖ Java JAR
- ❖ Visual Basic (VB) Scripts
- ❖ Python Files
- ❖ CPL Files
- ❖ Websites



Trace API calls and general behavior of the file and distill
this into high level information and signatures
comprehensible by anyone.



Dump and analyze network traffic, even when encrypted
with SSL/TLS.



Perform advanced memory analysis of the infected virtualized system through Volatility as well as on a process memory granularity using YARA.



Version History



CUCKOO



April 26th
2.0.2



20 bugfixes



May 19th
2.0.3
25 bugfixes



August 31st
2.0.3
30 bugfixes

Cuckoo's release cycle since the 2.0.0 release in April 2017

2.0
.4

Sept 6, 2017

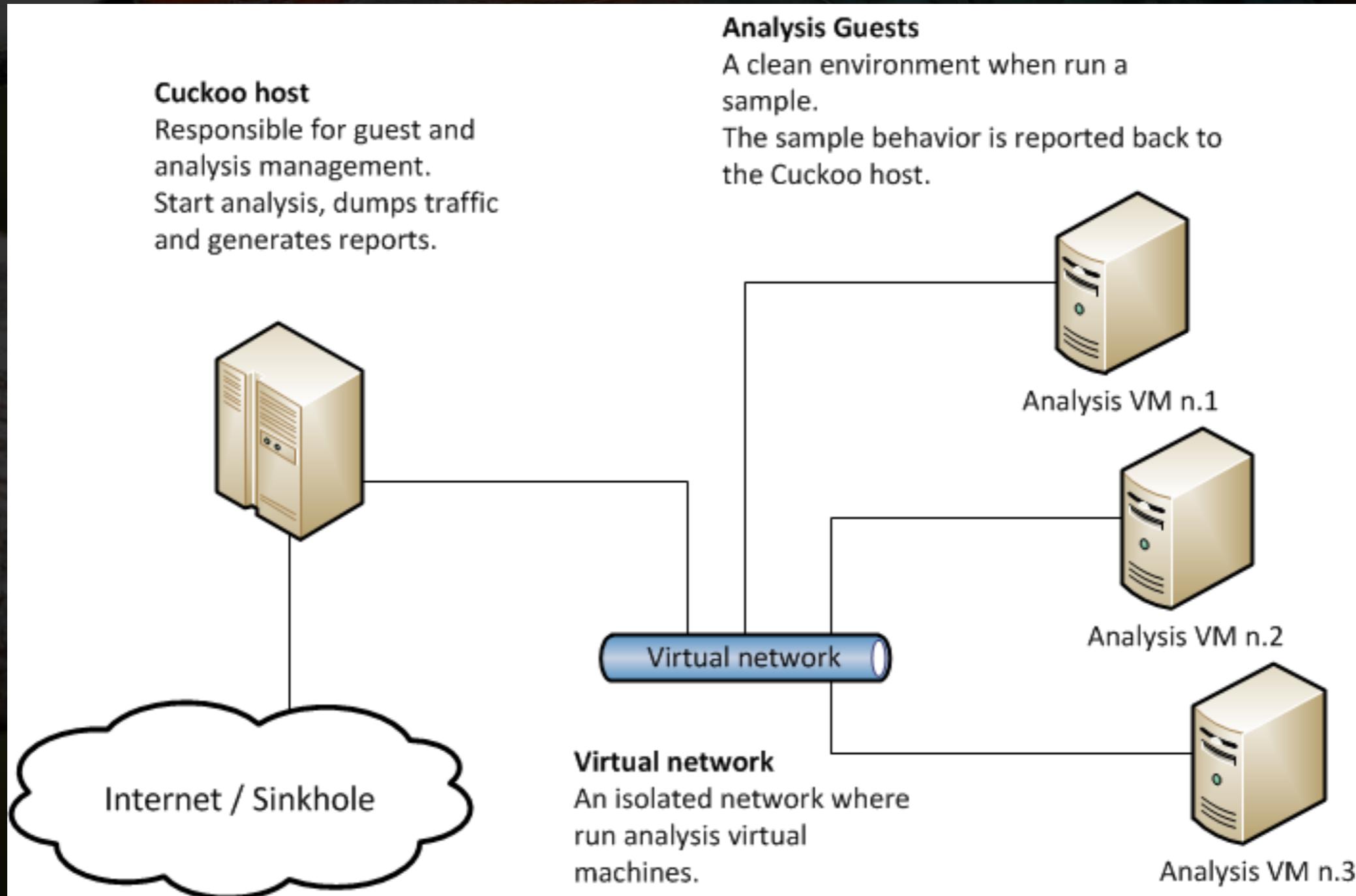


Smarter, faster, stronger.

Technology Stack

- ❖ Operating Systems
 - ❖ Linux, MacOS, Windows
- ❖ Python
- ❖ Virtualization Software
 - ❖ Virtual Box, KVM, XenServer
- ❖ tcpdump
- ❖ Volatility
- ❖ Yara
- ❖ mitmproxy
- ❖ Django
- ❖ MongoDB
- ❖ Virus Total

Architecture



cuckoo



- ❖ Python
- ❖ tcpdump
- ❖ Volatility
- ❖ Yara
- ❖ Virtualization Software



Cuckoo Sandbox



latest

Search docs

FAQ

Introduction

Installation

Preparing the Host

Requirements

Installing Python libraries (on
Ubuntu/Debian-based
distributions)

Installing Python libraries (on
Mac OS X)

Installing Python libraries (on
Windows 7)

Virtualization Software

Installing tcpdump

Installing Volatility

Installing M2Crypto

Installing Cuckoo

Per-Analysis Network Routing

Cuckoo Working Directory

Docs > Installation > Preparing the Host > Requirements

Edit on GitHub

Requirements

Before proceeding to installing and configuring Cuckoo, you'll need to install some required software packages and libraries.

Installing Python libraries (on Ubuntu/Debian-based distributions)

The Cuckoo host components is completely written in Python, therefore it is required to have an appropriate version of Python installed. At this point we only fully support Python 2.7. Older version of Python and Python 3 versions are not supported by us (although Python 3 support is on our TODO list with a low priority).

The following software packages from the apt repositories are required to get Cuckoo to install and run properly:

```
$ sudo apt-get install python python-pip python-dev libffi-dev libssl-dev  
$ sudo apt-get install python-virtualenv python-setuptools  
$ sudo apt-get install libjpeg-dev zlib1g-dev swig
```

In order to use the Django-based Web Interface, MongoDB is required:

```
$ sudo apt-get install mongodb
```



- ❖ `cuckoo.conf` * - for configuring general behavior and analysis options
- ❖ `auxiliary.conf` - for enabling and configuring auxiliary modules
- ❖ `machinery.conf` * - for defining the options for your virtualization software
- ❖ `memory.conf` - Volatility configuration
- ❖ `processing.conf` - for enabling and configuring processing modules
- ❖ `reporting.conf` - for enabling or disabling report formats

* required to get Cuckoo working



- ✿ `cuckoo.conf`
- ✿ **machinery** - defines which Machinery module you want Cuckoo to use to interact with your analysis machines. The value **must** be the name of the module without extension
- ✿ **ip & port** - define the local IP address and port that Cuckoo is going to try to bind the result server on
- ✿ **database** - connection string defines how Cuckoo will connect to the internal database



- `machinery.conf` - Machinery modules are scripts that define how Cuckoo should interact with your virtualization software of choice.
 - Virtualbox - `$CWD/conf/virtualbox.conf`
 - KVM - `$CWD/conf/kvm.conf`
 - XenServer - operates through an API, so to access it a URL and credentials are required



Community

- █ downloads Cuckoo Signatures, the latest monitoring binaries, and other goodies from the Cuckoo Community Repository and installs them
- █ `cuckoo community`

```
$ cuckoo community --help
Usage: cuckoo community [OPTIONS]

Utility to fetch supplies from the Cuckoo Community.

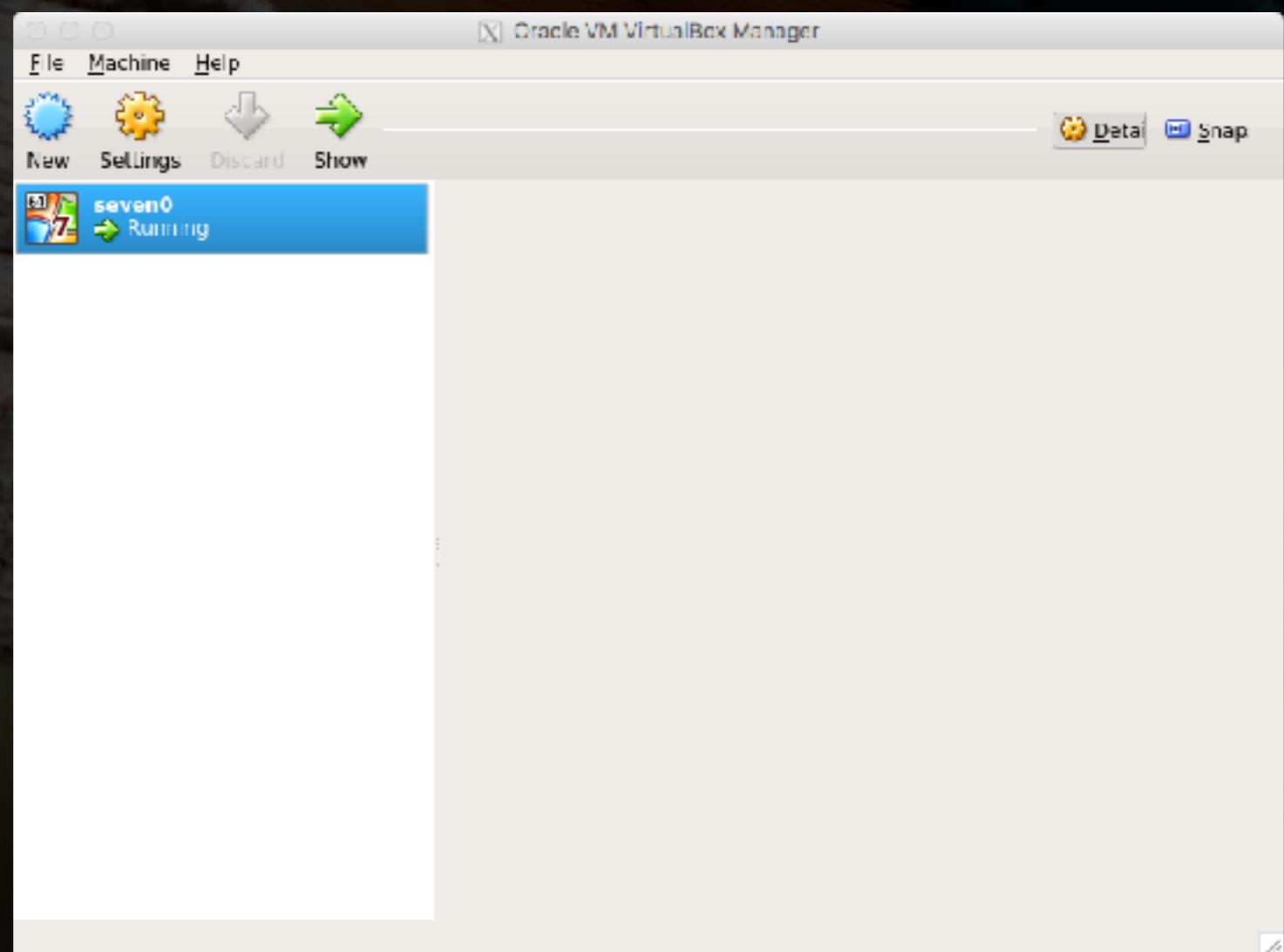
Options:
  -f, --force          Overwrite existing files
  -b, --branch TEXT   Specify a different community branch rather than
                      master
  --file, --filepath PATH  Specify a local copy of a community .tar.gz file
  --help                Show this message and exit.
```



- ❖ Guess OS Install (everything you need)
 - ❖ Windows 7 64bit
 - ❖ Yields much better results
 - ❖ Must disable User Access Control
 - ❖ Windows XP
 - ❖ Python
 - ❖ Python Pillow - Used for taking screenshots
 - ❖ Agent.py



- ❖ ssh -X ubuntu@192.168.0.10
- ❖ *run virtualbox*





- ❖ Additional Software
 - ❖ Depending on what kind of files you want to analyze and what kind of sandboxed Windows environment you want to run the malware samples in, you might want to install additional software such as browsers, PDF readers, office suites etc. Remember to disable the “auto update” or “check for updates” feature of any additional software.

VMCloak



- <http://vmcloak.org/>
- <https://github.com/jbremer/vmcloak>
- VMCloak is a tool for automatically creating, cloning, and cloaking Virtual Machines to be used for Cuckoo Sandbox
 - Microsoft Office Documents
- Utilizes the Cuckoo Agent

CUCKOO



```
01 # Install the latest vmcloak.  
02 sudo pip install vmcloak --upgrade  
03  
04 # Mount the Windows 7 Installer ISO.  
05 sudo mkdir -p /mnt/win7  
06 sudo mount -o loop,ro win7.iso /mnt/win7  
07  
08 # Ensure the hostonly adapter is up.  
09 vmcloak-vboxnet0  
10  
11 # Actually initialize the 64-bit Windows 7 VM.  
12 vmcloak init --win7x64 seven0
```

```
1 | vmcloak install seven0 office2007 \  
2 |   office2007.isopath=/path/to/a.iso \  
3 |   office2007.serialkey=ABC-DEF
```

```
1 | vmcloak install seven0 adobe9 wic pillow dotnet40 java7
```



API

Resource	Description
<code>GET /api/node</code>	Get a list of all enabled Cuckoo nodes.
<code>POST /api/node</code>	Register a new Cuckoo node.
<code>GET /api/node/<name></code>	Get basic information about a node.
<code>PUT /api/node/<name></code>	Update basic information of a node.
<code>POST /api/node/<name>/refresh</code>	Refresh a Cuckoo nodes metadata.
<code>DELETE /api/node/<name></code>	Disable (not completely remove!) a node.
<code>GET /api/task</code>	Get a list of all (or a part) of the tasks in the database.
<code>POST /api/task</code>	Create a new analysis task.
<code>GET /api/task/<id></code>	Get basic information about a task.
<code>DELETE /api/task/<id></code>	Delete all associated information of a task.
<code>GET /api/report/<id>/<format></code>	Fetch an analysis report.



CLI

```
$ cuckoo submit --help
Usage: cuckoo submit [OPTIONS] [TARGET]...

Submit one or more files or URLs to Cuckoo.

Options:
  -u, --url          Submitting URLs instead of samples
  -o, --options TEXT Options for these tasks
  --package TEXT     Analysis package to use
  --custom TEXT      Custom information to pass along this task
  --owner TEXT       Owner of this task
  --timeout INTEGER  Analysis time in seconds
  --priority INTEGER Priority of this task
  --machine TEXT    Machine to analyze these tasks on
  --platform TEXT   Analysis platform
  --memory           Enable memory dumping
  --enforce-timeout Don't terminate the analysis early
  --clock TEXT       Set the system clock
  --tags TEXT        Analysis tags
  --baseline          Create baseline task
  --remote TEXT      Submit to a remote Cuckoo instance
  --shuffle          Shuffle the submitted tasks
  --pattern TEXT     Provide a glob-pattern when submitting a
                    directory
  --max INTEGER      Submit up to X tasks at once
  --unique           Only submit samples that have not been
                    analyzed before
  -d, --debug        Enable verbose logging
  -q, --quiet        Only log warnings and critical messages
  --help             Show this message and exit.
```

Free Malware Analysis Tools



- <https://malwr.com/>
- <https://www.virustotal.com/>
- <https://www.hybrid-analysis.com/>

Things to Consider

- Tipping Your Hat
 - Monitored by Malware Authors
- Data Leakage
 - <https://thehackernews.com/2017/08/fortune-1000-data-leak.html>
- Availability
- File Size
- Typically Not Up-to-date



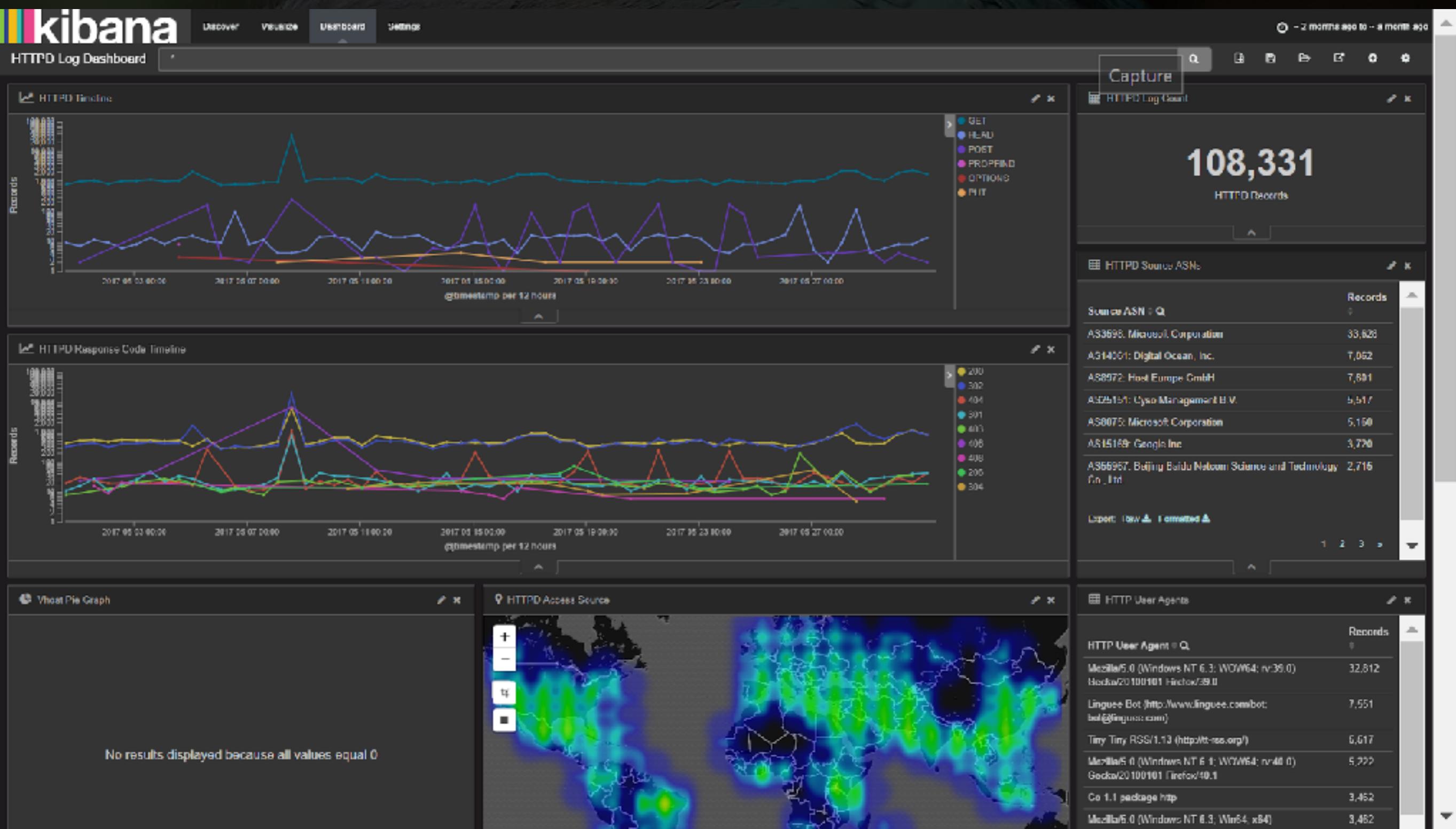
WARNING

**ASSUMPTIONS
A H E A D**

Tools



- SANS 572 - Advanced Network Forensics and Analysis
- Phil Hagen - @PhilHagen
- <https://github.com/philhagen/sof-elk>
- Mantella (utahsaint.org)
- Unsure if its an active project?





DEMO



Malware Samples



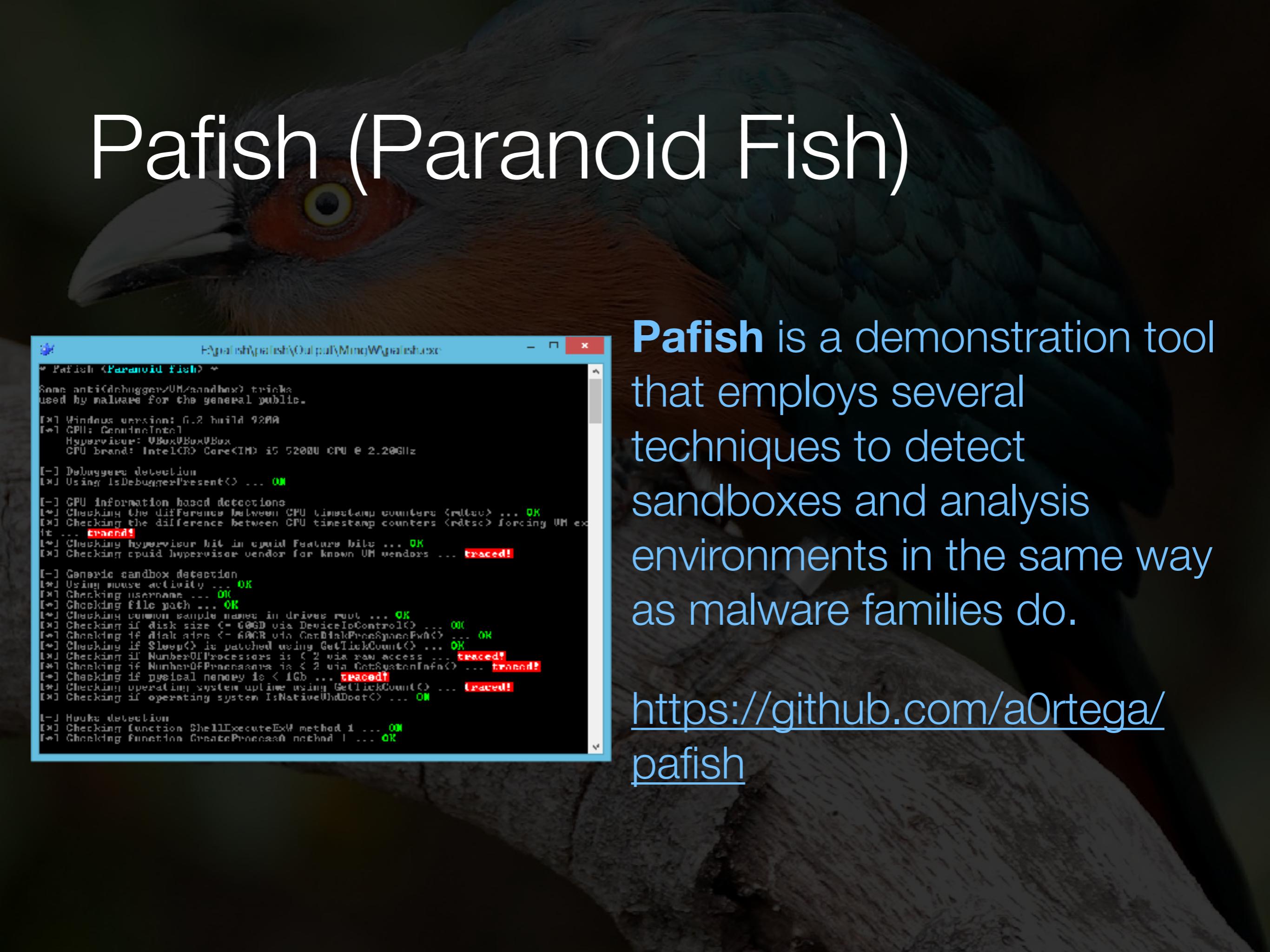
- theZoo - <https://github.com/ytisf/theZoo/tree/master/malwares/Binaries>
- ObjectiveSee - <https://objective-see.com/malware.html>
- <http://dasmalwerk.eu/>
- <http://www.virusign.com/>

Warning

- Virtualization Detection
 - Changing MAC Address
- Sandbox Bypassing
 - 2017 - BSidesSF - Bypassing Malware Analysis Sandboxes is Easy by Michael Gough
@HackerHurricane



Pafish (Paranoid Fish)



```
Pafish<Paranoid fish> ~
Some anti-debugger/VM/analysis tricks
used by malware for the general public.

[+] Windows version: 6.2 build: 9200
[+] CPU: GenuineIntel
  Hypervisor: VBoxVBoxVBox
  CPU brand: Intel(R) Core(TM) i5 5200U CPU @ 2.20GHz

[-] Debuggers detection
[+] Using IsDebuggerPresent() ... OK

[-] CPU information based detections
[+] Checking the difference between CPU timestamp counters (rdtsc) ... OK
[+] Checking the difference between CPU timestamp counters (rdtsc) forcing VM exit ... traced!
[+] Checking hypervisor bit in cpuid Feature Bits ... OK
[+] Checking cpuid hypervisor vendor for known VM vendors ... traced!

[-] Generic sandbox detection
[+] Using mouse activity ... OK
[+] Checking username ... OK
[+] Checking file path ... OK
[+] Checking common sample names in drives root ... OK
[+] Checking if disk size <= 60GB via DeviceIoControl() ... OK
[+] Checking if disk size <= 60GB via GetDiskFreeSpaceEx() ... OK
[+] Checking if Sleep() is patched using GetTickCount() ... OK
[+] Checking if NumberOfProcesses is < 2 via raw access ... traced!
[+] Checking if NumberOfProcesses is < 2 via GetSystemInfo() ... traced!
[+] Checking if physical memory is < 1Gb ... traced!
[+] Checking operating system uptime using GetTickCount() ... traced!
[+] Checking if operating system IsNativeUhdDxgi() ... OK

[-] Hook detection
[+] Checking function ShellExecuteExW method 1 ... OK
[+] Checking function CreateProcessW method 1 ... OK
```

Pafish is a demonstration tool that employs several techniques to detect sandboxes and analysis environments in the same way as malware families do.

<https://github.com/aOrtega/pafish>



@enruhe_