

# IRMA

*Incident Response & Malware Analysis*

QUARKSLAB  
INNOVATIVE SECURITY

*Hack in the Box - Amsterdam - 2015*

*Guillaume Dedrie - Alexandre Quint - Fernand Lone Sang*

# Agenda

1. Problematic
2. Internals and results
3. A community project
4. Workshop
5. Conclusion

# Agenda

1. Problematic
2. Internals and results
3. A community project
4. Workshop
5. Conclusion

# Problematic

4

De: admin@chat-k.cat  
À: me  
Sujet: Try this one !!!

<3 cats



BestCatScreensaverEver.exe

*Is BestCatScreensaverEver.exe clean?*

**Solution #1** : scan it with your antivirus.

*Is BestCatScreensaverEver.exe clean?*

**Solution #1** : scan it with your antivirus.

+ easy

*Is BestCatScreensaverEver.exe clean?*

**Solution #1** : scan it with your antivirus.

- + easy
- + quick (well... often)

*Is BestCatScreensaverEver.exe clean?*

**Solution #1** : scan it with your antivirus.

- + easy
- + quick (well... often)
- all the security based on one vendor



*Is BestCatScreensaverEver.exe clean?*

**Solution #1** : scan it with your antivirus.

- + easy
- + quick (well... often)
- all the security based on one vendor

Good but not enough

*Is BestCatScreensaverEver.exe clean?*

**Solution #2** : send it to a website for scanning

*Is BestCatScreensaverEver.exe clean?*

**Solution #2** : send it to a website for scanning

- + many sites freely available:
  - virustotal.com
  - avcaesar.malware.lu
  - metascan.com

*Is BestCatScreensaverEver.exe clean?*

**Solution #2** : send it to a website for scanning

- + many sites freely available:
  - virustotal.com
  - avcaesar.malware.lu
  - metascan.com
- + many antivirus supported

*Is BestCatScreensaverEver.exe clean?*

**Solution #2** : send it to a website for scanning

- + many sites freely available:
  - virustotal.com
  - avcaesar.malware.lu
  - metascan.com
- + many antivirus supported
- one file at a time

*Is BestCatScreensaverEver.exe clean?*

**Solution #2** : send it to a website for scanning

- + many sites freely available:
  - virustotal.com
  - avcaesar.malware.lu
  - metascan.com
- + many antivirus supported
- one file at a time
- files are sent on the Internet

*Is BestCatScreensaverEver.exe clean?*

**Solution #2** : send it to a website for scanning

- + many sites freely available:
  - virustotal.com
  - avcaesar.malware.lu
  - metascan.com
- + many antivirus supported
- one file at a time
- files are sent on the Internet
- scan settings are unknown

*Is BestCatScreensaverEver.exe clean?*

**Solution #2** : send it to a website for scanning

- + many sites freely available:
  - virustotal.com
  - avcaesar.malware.lu
  - metascan.com
- + many antivirus supported
- one file at a time
- files are sent on the Internet
- scan settings are unknown

Good but not enough



*Is BestCatScreensaverEver.exe clean?*

**Solution #3** : Open the file #YOLO

*Is BestCatScreensaverEver.exe clean?*

**Solution #3** : Open the file #YOLO



*Is BestCatScreensaverEver.exe clean?*

**Solution #3** : Open the file #YOLO

- + opportunity to test your backup/restore procedures

*Is BestCatScreensaverEver.exe clean?*

**Solution #3** : Open the file #YOLO

- + opportunity to test your backup/restore procedures

No comment

# New threats → New tools

21

Companies and public CERT share the same analysis:

Use of a single antivirus is not enough, but antivirus cannot be avoided.

Antivirus are a source of information, among other ones, in the incident response process.

# New threats → New tools

22

Companies and public CERT share the same analysis:

Use of a single antivirus is not enough, but antivirus cannot be avoided.

Antivirus are a source of information, among other ones, in the incident response process.

To handle all these sources and gather the most information, a **modular, scalable** tool which can rely on a **community** of users/contributors is needed.

# Joint initiative





## *Incident Response & Malware Analysis*

- **Private** file analysis platform
- **Open source** (Apache V2 license)
- **Customisable**



- **Private** platform: no data ever leaves your network
- **Analyze** files, and not only with antivirus  
(24 analyzers available)
- **Several** files simultaneously analyzed
- **Open source** (code hosted on GitHub)
- **Customizable** (API, plugins)

# Analysis modules

26

AVIRA  
GDATA  
MCAFEE  
SYMANTEC

EMSIOSFT  
KASPERSKY  
SOPHOS



*ANTIVIRUS*

AVAST  
BITDEFENDER  
COMODO  
ESETNOD32  
FPROT  
MCAFEE

AVG  
CLAMAV  
DrWEB  
ESCAN  
FSECURE  
SOPHOS

VIRUSBLOKADA  
ZONER



*ANTIVIRUS*

PEiD  
YARA  
PE STATIC ANALYSIS

*METADATA*

NSRL

*DATABASE*

VIRUSTOTAL

*EXTERNAL*

# Other usage examples

27

- Web API

**IRMA API**

[Apache 2.0](#)

**Scans** Show/Hide List Operations Expand Operations

GET	/scans	List all scans
POST	/scans	Create a scan
GET	/scans/{scanId}	Retrieve a scan
POST	/scans/{scanId}/launch	Launch a scan
POST	/scans/{scanId}/cancel	Cancel a scan
POST	/scans/{scanId}/files	Create a file upload
GET	/scans/{scanId}/results	List all results from a scan
GET	/scans/{scanId}/results/{resultId}	Retrieve a result for a specific scan

# Other usage examples

28

- Web API
- Any client can access it
- New usages!

# Other usage examples

29

Cleaning kiosk for USB keys



# Other usage examples

30

Cleaning kiosk for USB keys



Filter for mail attachments



# A few figures

31

- Project started in November 2013.
- 3 Quarkslab engineers.
- 1 Orange intern for 6 months.

Total: 680 days at the end of 2014 (3 man-years).

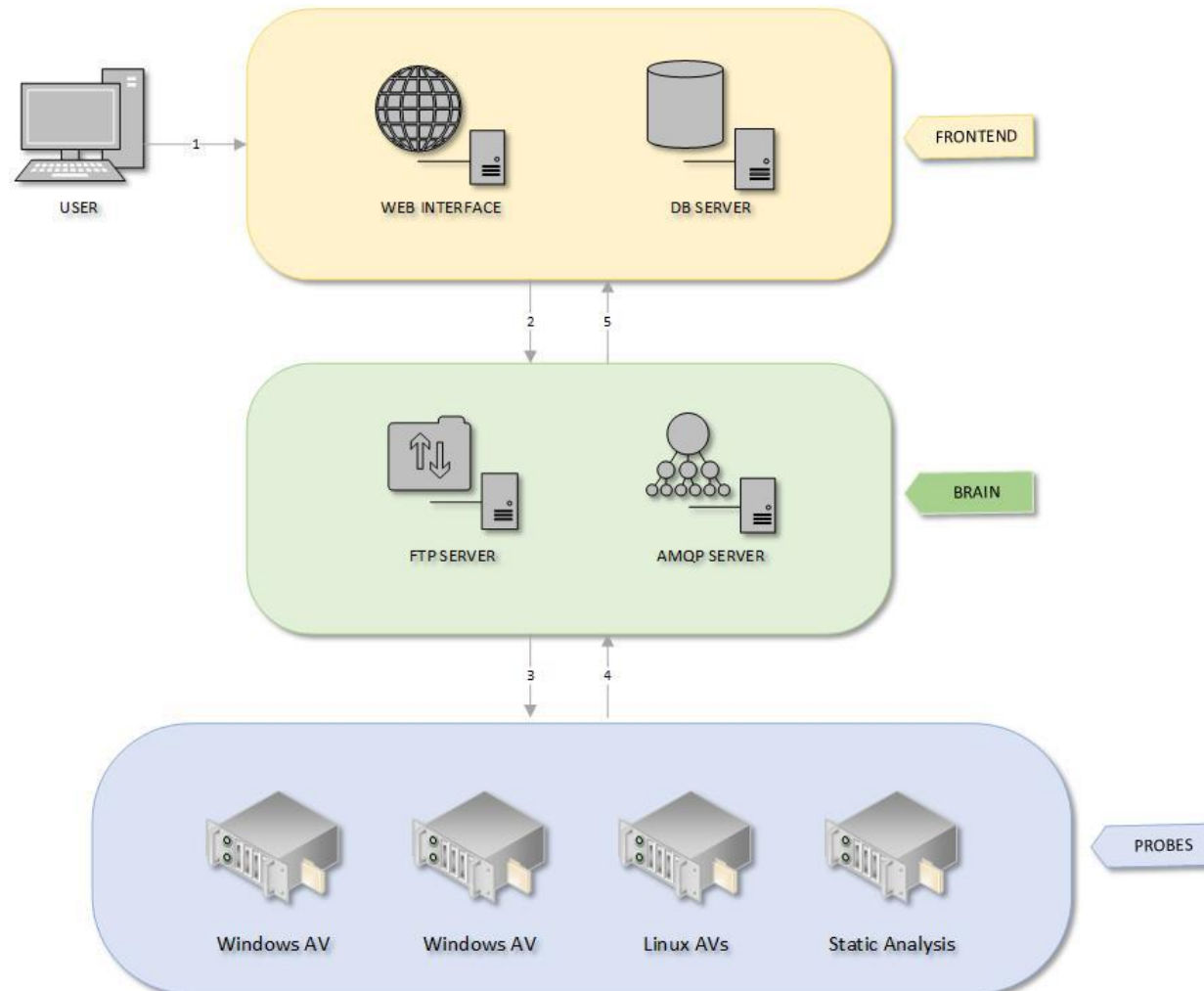
# Agenda

1. Problematic
2. Internals and results
3. A community project
4. Workshop
5. Conclusion



# Global architecture

33



# Adding analysers

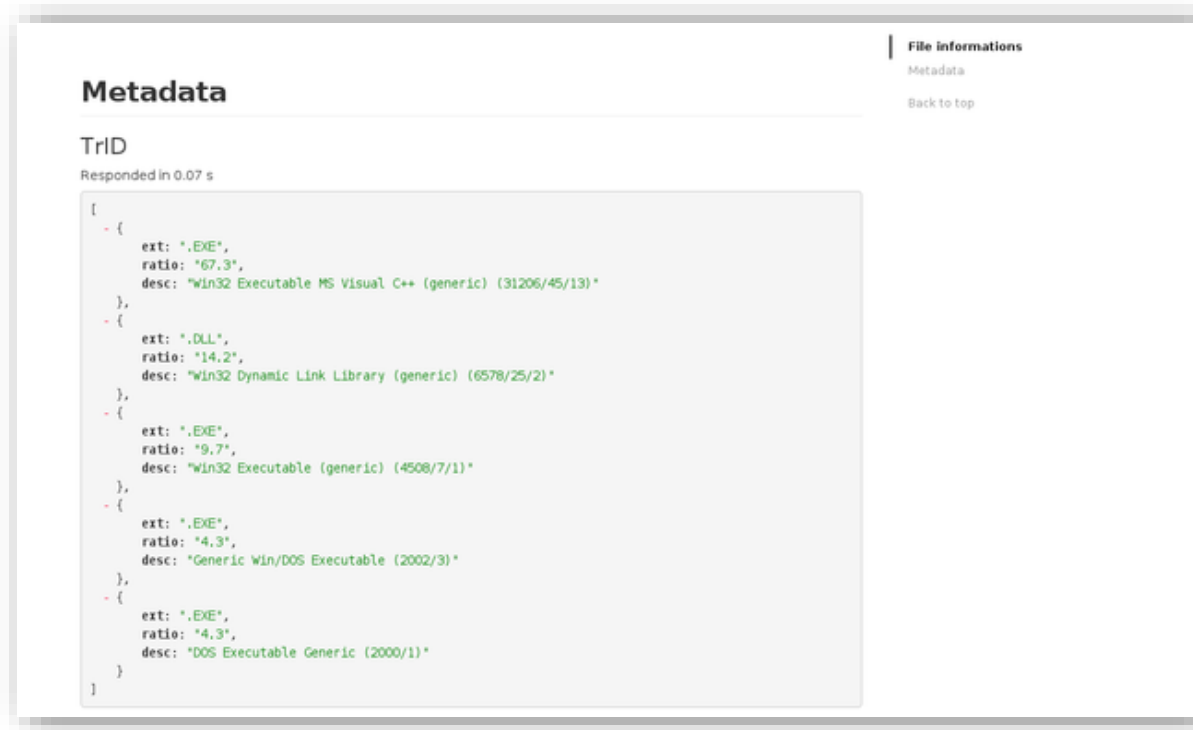
34

- Each analysis module is a plugin.
- Separated in two parts:
  - Interface, specific to IRMA
  - The processing part, which analyses the file. It is independant from IRMA and can be reused in another project.
- Plugins are automatically discovered when a probe is started.

# Customizing the results

35

- Each analysis result can be independently filtered.
- Plugins are dynamically discovered when the *frontend* is started.
- Results are kept in raw form in the database.



The screenshot displays a web interface for file analysis. On the right side, there is a sidebar with the heading "File informations" and two links: "Metadata" (which is active) and "Back to top". The main content area is titled "Metadata" and shows a "TrID" analysis result. Below the title, it states "Responded in 0.07 s". The analysis results are presented as a JSON array of objects, each representing a file type identified by TrID. The objects include fields for extension (ext), ratio, and description (desc).

```
[
  - {
    ext: ".EXE",
    ratio: "67.3",
    desc: "Win32 Executable MS Visual C++ (generic) (31206/45/13)"
  },
  - {
    ext: ".DLL",
    ratio: "14.2",
    desc: "Win32 Dynamic Link Library (generic) (6578/25/2)"
  },
  - {
    ext: ".EXE",
    ratio: "9.7",
    desc: "Win32 Executable (generic) (4508/7/1)"
  },
  - {
    ext: ".EXE",
    ratio: "4.3",
    desc: "Generic Win/DOS Executable (2002/3)"
  },
  - {
    ext: ".EXE",
    ratio: "4.3",
    desc: "DOS Executable Generic (2000/1)"
  }
]
```

# Customizing the results

36

- Each analysis result can be independently filtered.
- Plugins are dynamically discovered when the *frontend* is started.
- Results are kept in raw form in the database.

## Metadata

TrID  
Responded in 0.07 s

Description	File Extension	Ration (in %)
Win32 Executable MS Visual C++ (generic) (31206/45/13)	.EXE	67.3
Win32 Dynamic Link Library (generic) (6578/25/2)	.DLL	14.2
Win32 Executable (generic) (4508/7/1)	.EXE	9.7
Generic Win/DOS Executable (2002/3)	.EXE	4.3
DOS Executable Generic (2000/1)	.EXE	4.3

**File informations**  
Metadata  
[Back to top](#)



# Agenda

1. Problematic
2. Internals and results
- 3. A community project**
4. Workshop
5. Conclusion

# Building a community

39

Creating an open source project is good

If the project has **users**, it is better.

If it has **contributors**, it is even better.

Creating an open source project is good

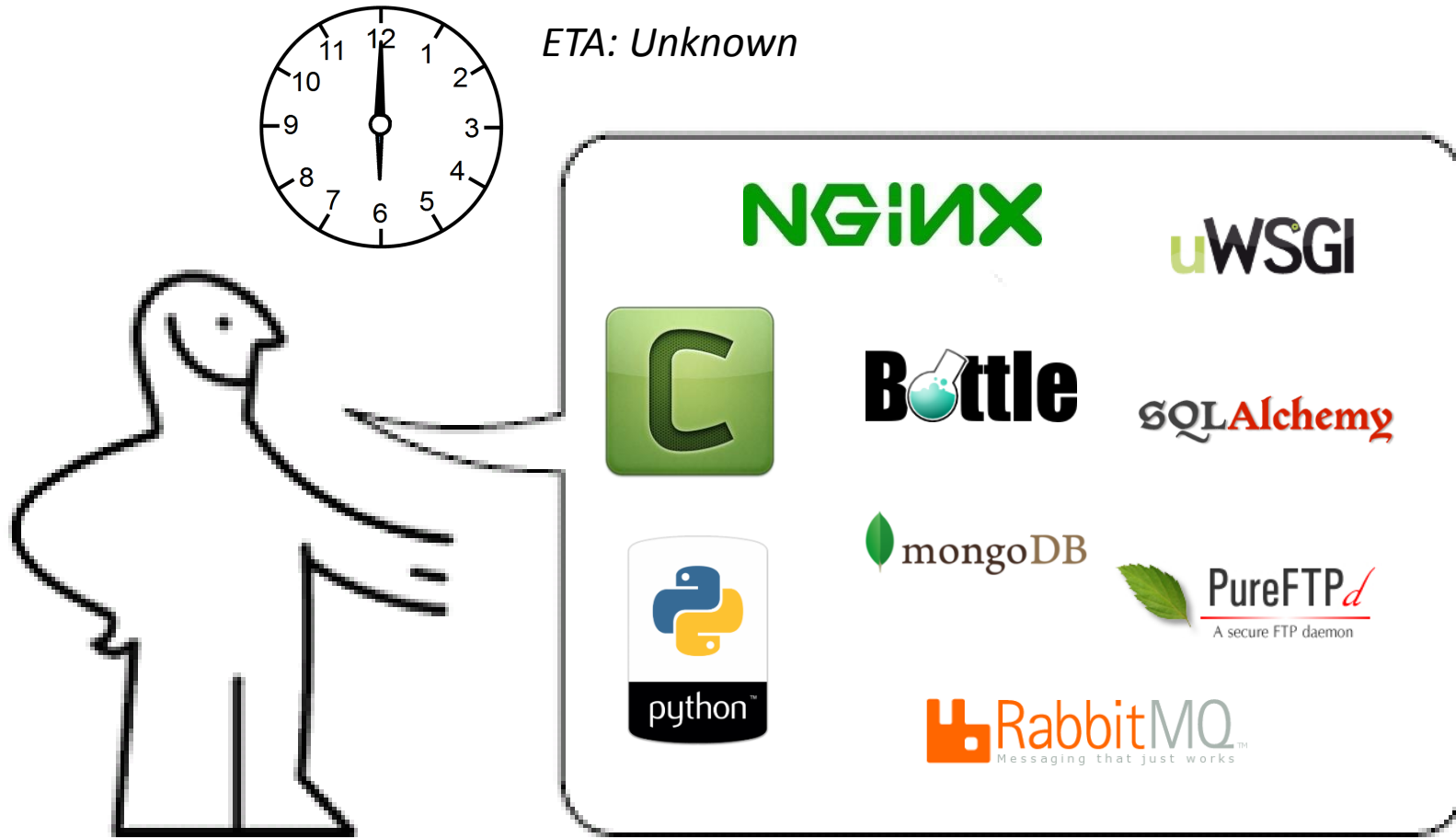
If the project has **users**, it is better.

If it has **contributors**, it is even better.

**Need for a simple, deterministic installation system**



# Installation v1.0



ETA: Unknown

NGINX

uWSGI

C

Bottle

SQLAlchemy

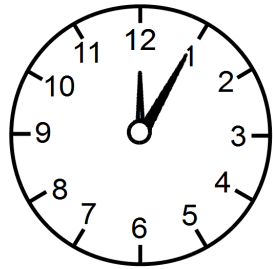
python

mongoDB

PureFTPd  
A secure FTP daemon

RabbitMQ  
Messaging that just works

# Installation v1.1.0



*ETA: 5 minutes*



A rounded rectangular box containing two logos. On the left is the VAGRANT logo, which is a blue, 3D-style letter 'V' with the word 'VAGRANT' in blue capital letters below it. On the right is the ANSIBLE logo, which is a black circle with a white letter 'A' inside, and the word 'ANSIBLE' in black capital letters below it.

# Installation v1.1.0

43

Installing Vagrant :

```
https://www.vagrantup.com/downloads.html
```

Installing Ansible :

```
$ sudo pip install ansible
```

Installing IRMA:

```
$ git clone https://github.com/quarkslab/irma-ansible  
$ cd irma-ansible  
$ ansible-galaxy install -r ansible-requirements.yml  
$ vagrant up
```

# The birth of a community

44

2 contributors, 3 new probes:

- YARA
- GDATA for Windows
- AVIRA for Windows

HITB challenge:

- Outlook submitter (scan all attachments)
- ICAP probe

# Agenda

1. Problematic
2. Internals and results
3. A community project
- 4. Workshop**
5. Conclusion

# Workshop agenda

46

- PROBE - Create your own probe
- PROBE - Integrate it in IRMA
- FRONTEND - Add a formatter to customize its output
- FRONTEND - API 101

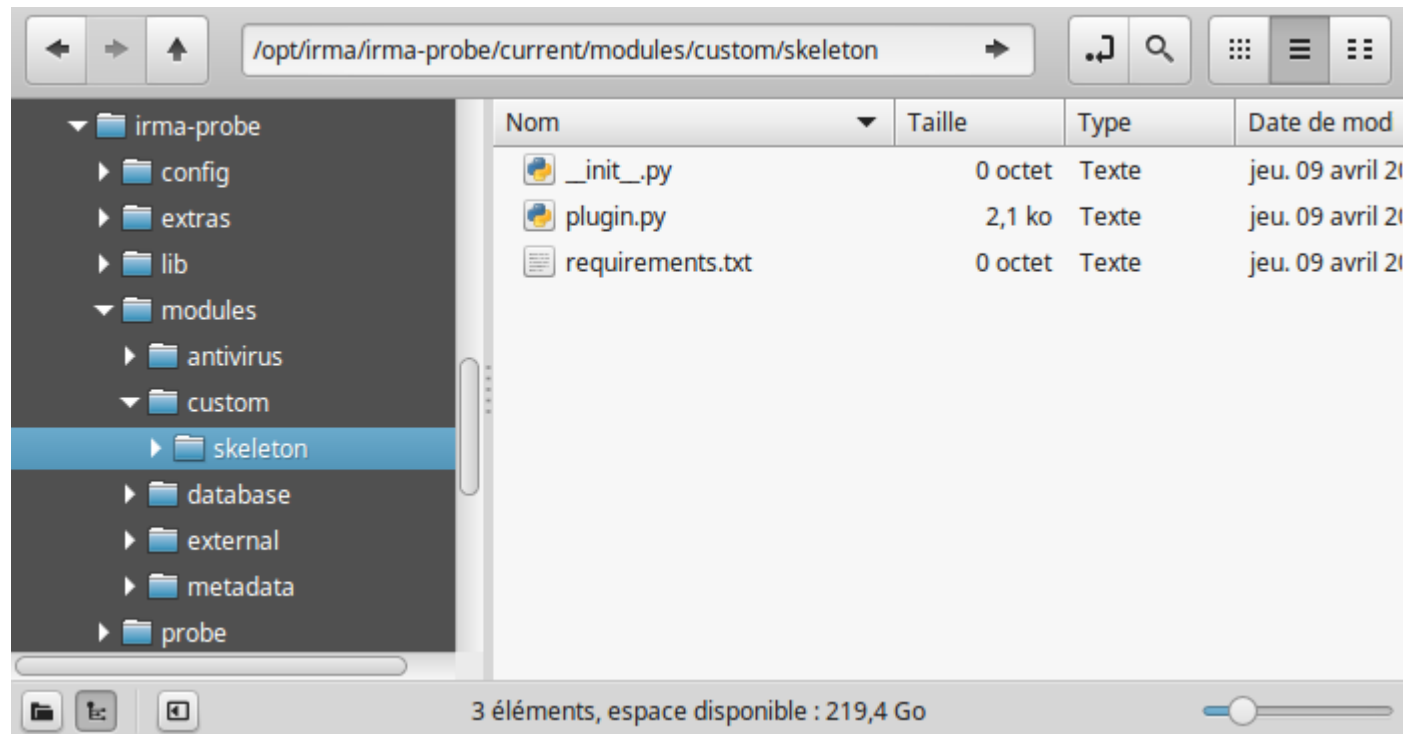
# Workshop agenda

47

- PROBE - Create your own probe
- PROBE - Integrate it in IRMA
- FRONTEND - Add a formatter to customize its output
- FRONTEND - API 101

# Probe skeleton

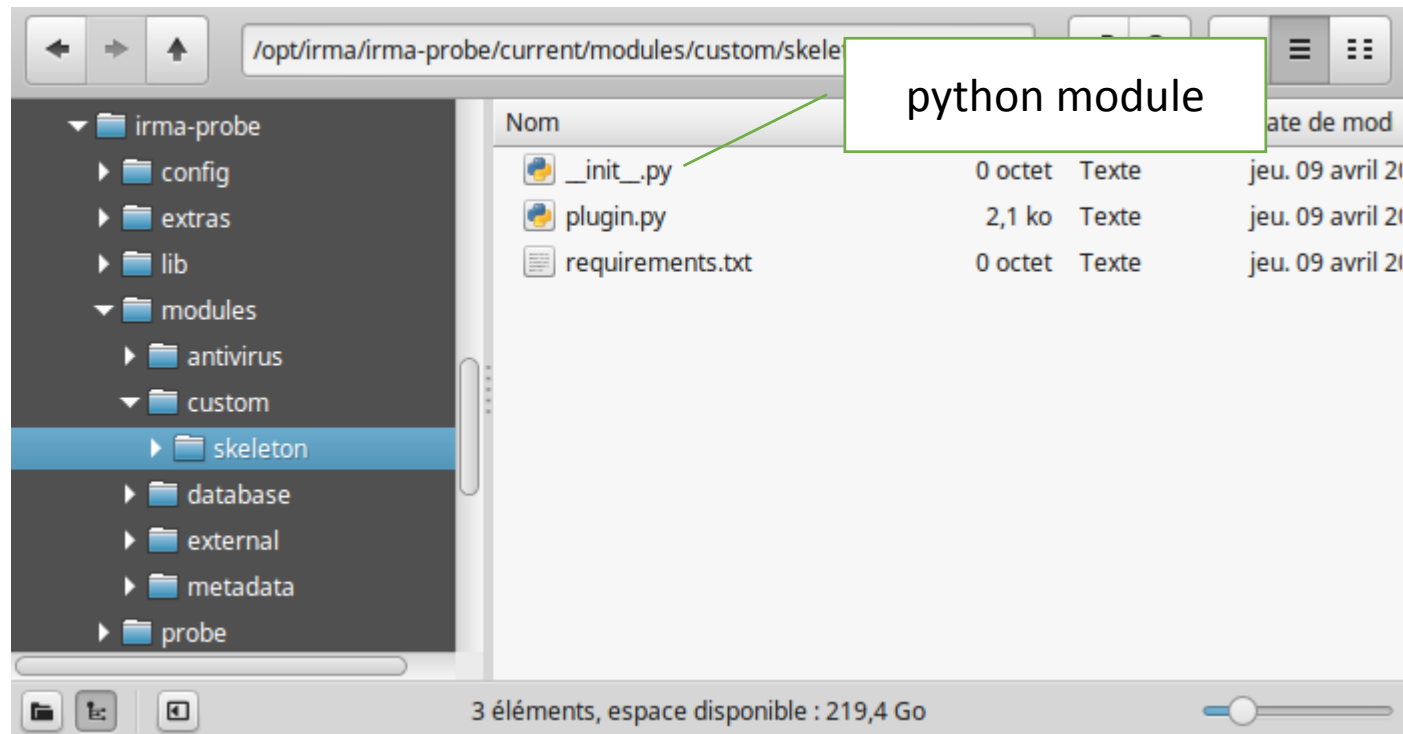
48





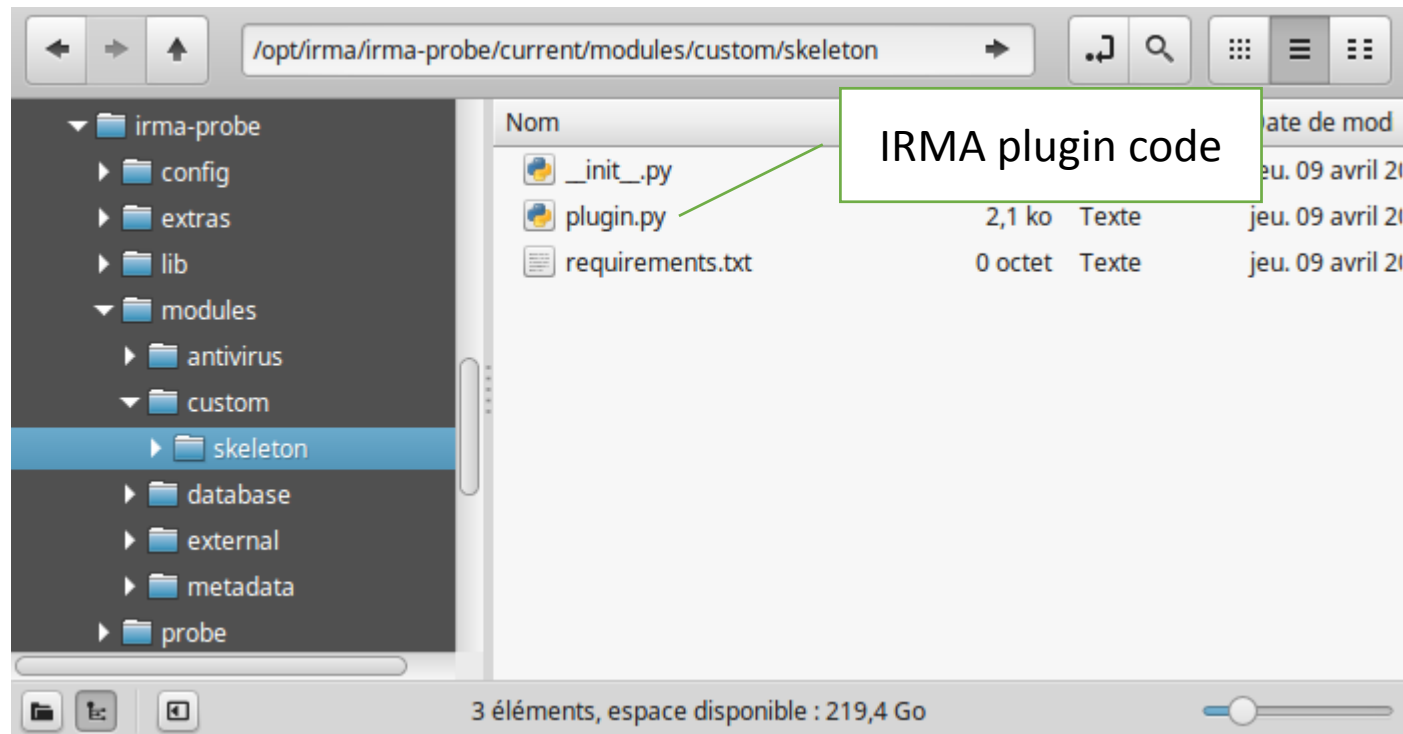
# Probe skeleton

49



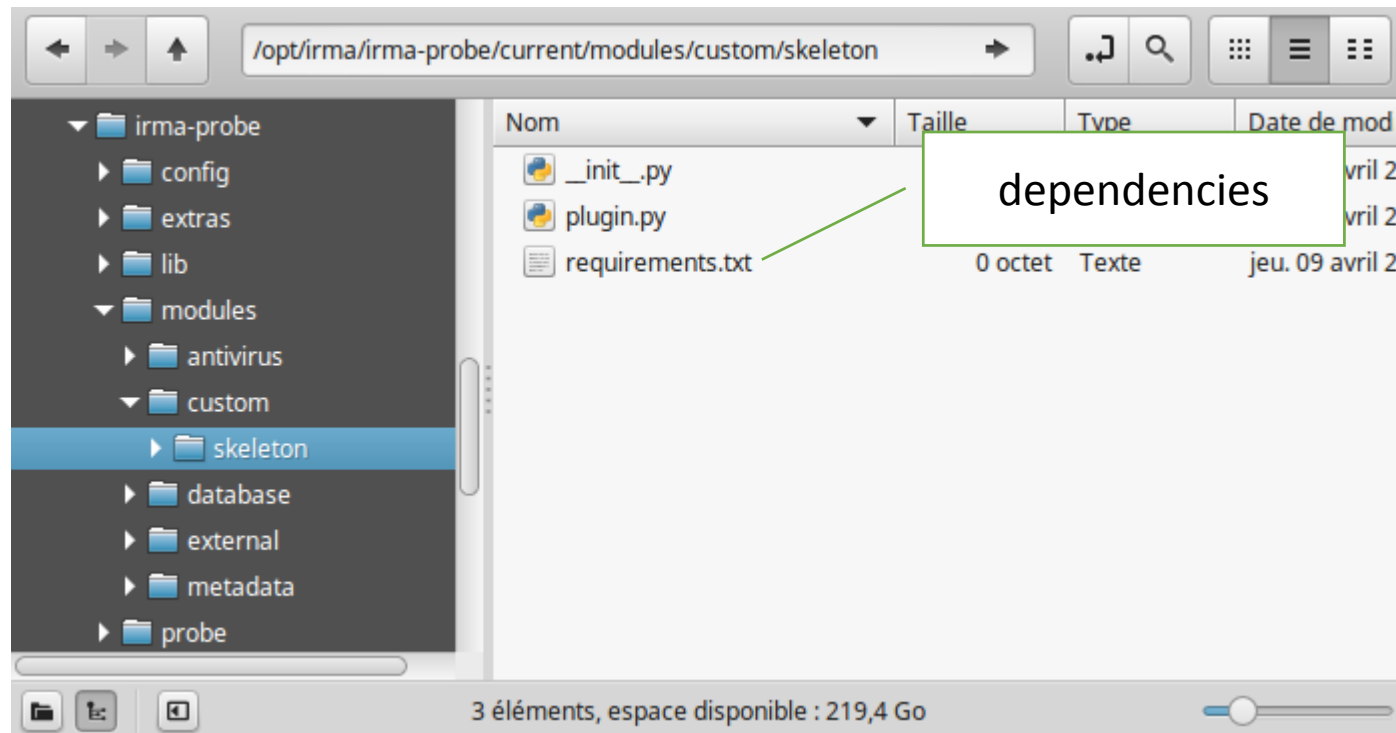
# Probe skeleton

50



# Probe skeleton

51



# Probe Creation – Balbuzard probe

52



**Balbuzard - malware analysis tools to extract patterns of interest and crack obfuscation such as XOR**

Author: Philippe Lagadec

Homepage: <http://www.decalage.info/python/balbuzard>

# Balbuzard 101

53

```
>> from balbuzard.balbuzard import patterns, Balbuzard
>> Bal = Balbuzard(patterns=patterns)
>> data = open("./attachment1.exe").read()
>> list(Bal.scan(data))
[(<balbuzard.balbuzard.Pattern at 0x7fd37cda23d0>, [(0, 'MZ'), (15320, 'MZ')]),
 (<balbuzard.balbuzard.Pattern at 0x7fd37cda2410>,
 [(232, 'PE'), (9541, 'PE'), (50172, 'PE'), (78332, 'PE')]),
 [...],
 (<balbuzard.balbuzard.Pattern at 0x7fd37cda2710>, [(27129, 'Pop')])]
```

# Balbuzard probe – connect to VM

54

## VM ADDRESS ?

Credentials: vagrant/vagrant

## SSH TIME

```
$ ssh vagrant@vm_address -i vagrant_insecure_private_key  
vagrant@brain:~$
```

# Balbuzard probe – level 0

55

## Create directory

- Copy Skeleton directory

```
$ sudo su deploy
$ cd /opt/irma/irma-probe/current/modules/metadata
$ git clone https://github.com/quarkslab/irma-probe-tutorial balbuzard_analyzer
$ cd balbuzard_analyzer
$ git checkout origin/balbuzard-level0
```

# Balbuzard probe – level 1

56

## Update metadata

- Rename all Skeleton in Balbuzard
- Update Metadata



# Balbuzard probe – level 1

57

## Update metadata

- Rename all Skeleton in Balbuzard
- Update Metadata

```
$ git diff origin/balbuzard-level1  
$ git checkout -f origin/balbuzard-level1
```

# Balbuzard probe – level 2

58



## Handle dependencies

- declare module dependencies

# Balbuzard probe - dependencies - level 2

59

```
>> from balbuzard.balbuzard import patterns, Balbuzard
```



 plugin.py  
 requirements.txt

```
_plugin_dependencies_ = [  
    ModuleDependency(  
        'balbuzard',  
        help='See requirements.txt for needed dependencies'  
    ),
```

# Balbuzard probe - dependencies - level2

60

```
>> from balbuzard.balbuzard import patterns, Balbuzard
```

 plugin.py  
 requirements.txt



```
_plugin_dependencies_ = [  
    ModuleDependency(  
        'balbuzard',  
        help='See requirements.txt for needed dependencies'  
    ),
```

```
balbuzard>=0.19
```

# Balbuzard probe - dependencies - level2

61

```
>> from balbuzard.balbuzard import patterns, Balbuzard
```

 plugin.py  
 requirements.txt

```
_plugin_dependencies_ = [  
    ModuleDependency(  
        'balbuzard',  
        help='See requirements.txt for needed dependencies'  
    ),
```

```
balbuzard>=0.19
```

```
$ git diff origin/balbuzard-level2  
$ git checkout -f origin/balbuzard-level2
```

# Balbuzard probe – level 3

62


## Output results

- use analysis module to output interesting results

# Balbuzard probe - processing - level3

63

```
>> Bal = Balbuzard(patterns=patterns)
>> data = open("./attachment1.exe").read()
>> list(Bal.scan(data))
```

 plugin.py


```
def __init__(self):
    module = sys.modules['balbuzard.balbuzard']
    patterns = module.patterns
    self.Analyzer = module.Balbuzard(patterns=patterns)

def run(self, paths):
    [...]
    try:
        started = timestamp(datetime.utcnow())
        with open(paths, "rb") as f:
            data = f.read()
            res = list(self.Analyzer.scan(data))
        response.results = res
```

# Balbuzard probe - processing - level3

64

```
>> Bal = Balbuzard(patterns=patterns)
>> data = open("./attachment1.exe").read()
>> list(Bal.scan(data))
```

 plugin.py

```
def __init__(self):
    module = sys.modules['balbuzard.balbuzard']
    patterns = module.patterns
    self.Analyzer = module.Balbuzard(patterns=patterns)
```

```
def run(self, paths):
    [...]
    try:
```

```
$ git diff origin/balbuzard-level3
$ git checkout -f origin/balbuzard-level3
```

```
res = list(self.Analyzer.scan(data))
response.results = res
```



# Test it

65

```
vagrant@brain:~$ sudo su irma

irma@brain:~$ cd /opt/irma/irma-probe/current

irma@brain:~$ venv/bin/python -m tools.run_module

irma@brain:~$ venv/bin/python -m tools.run_module Balbuzard /bin/ls
[...]
{'duration': 0.03014206886291504,
 'error': None,
 'name': 'Balbuzard',
 [...]
 'type': 'metadata',
 'version': None}
```

# Workshop agenda

66

- PROBE - Create your own probe
- **PROBE - Integrate it in IRMA**
- FRONTEND - Add a formatter to customize its output
- FRONTEND - API 101

```
vagrant@brain:~$ sudo supervisorctl restart probe_app
probe_app: stopped
probe_app: started

vagrant@brain:~$ sudo supervisorctl tail probe_app
[...]
WARNING:probe.tasks: *** [metadata] Plugin Balbuzard successfully
loaded
```

# Job done!

68

## Metadata

Balbuzard

Responded in 0.07 s

```
{"EXE: section name":[[480,".text"],[560,".data"],[520,".rdata"],[600,".rsrc"]],"Executable filename":[[631
```

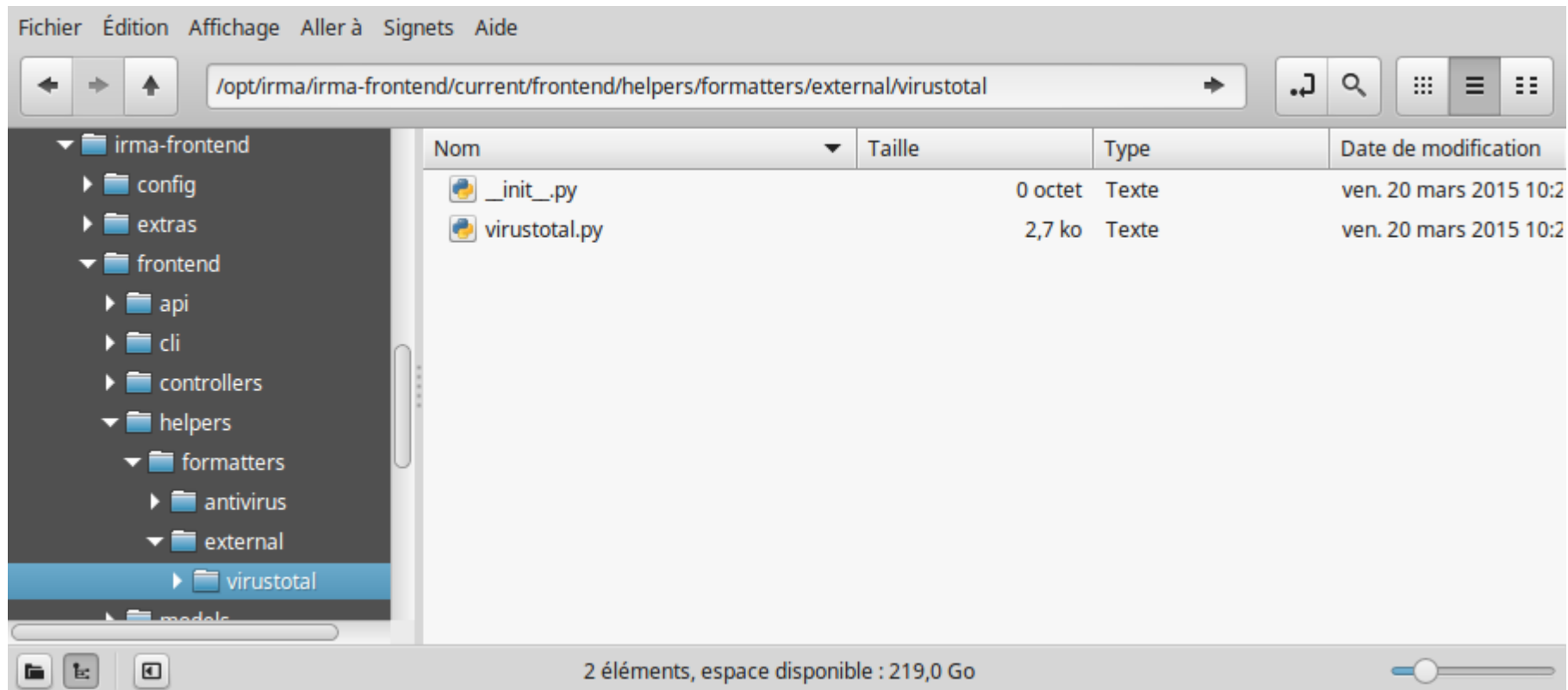
# Workshop agenda

69

- PROBE - Create your own probe
- PROBE - Integrate it in IRMA
- FRONTEND - Add a formatter to customize its output
- FRONTEND - API 101

# Formatter files

70



# Balbuzard probe – level 0

71

## Empty formatter

- Create empty formatter directory
- Apply only current formatter to balbuzard probe

```
$ sudo su deploy
$ cd /opt/irma/irma-frontend/current/frontend/helpers/formatters
$ git clone https://github.com/quarkslab/irma-formatter-tutorial balbuzard
$ cd balbuzard
$ git checkout origin/balbuzard-level0
```

# Test it

72

```
vagrant@brain:~$ sudo supervisorctl restart frontend_api  
frontend_api: stopped  
frontend_api: started
```



# Balbuzard probe – level 1

73

## First shot

- return something

```
$ git diff origin/balbuzard-level1  
$ git checkout -f origin/balbuzard-level1
```

# Balbuzard probe – level 2

74

## Exception handling

- catch exceptions in format

```
$ git diff origin/balbuzard-level2  
$ git checkout -f origin/balbuzard-level2
```

# Balbuzard probe – level 3

75

## Pretty output

- iterate through results items to pretty print it

```
$ git diff origin/balbuzard-level3  
$ git checkout -f origin/balbuzard-level3
```

```
vagrant@brain:~$ sudo supervisorctl restart frontend_api
frontend_api: stopped
frontend_api: started
```

## Metadata

### Balbuzard

Responded in 0.13 s

```
EXE: section name:
  .rdata (704)
  .rsrc (784)
  .reloc (744)
Executable filename:
  Explorer.exe (17184)
  winzip32.exe (18188)
  WinRAR.exe (18264)
  rar.bat (18287)
  zip.bat (18307)
  sIRC4.exe (52512)
  kernel32.dll (56400)
  user32.dll (56914)
  advapi32.dll (56970)
  oleaut32.dll (57034)
```

- PROBE - Create your own probe
- PROBE - Integrate it in IRMA
- FRONTEND - Add a formatter to customize its output
- **FRONTEND - API 101**

# Swagger documentation

78

visit [http://<vm\\_address>/swagger](http://<vm_address>/swagger)

# Swagger documentation

79

## IRMA API

[Apache 2.0](#)

### Scans

Show/Hide | List Operations | Expand Operations

GET	/scans	List all scans
POST	/scans	Create a scan
GET	/scans/{scanId}	Retrieve a scan
POST	/scans/{scanId}/launch	Launch a scan
POST	/scans/{scanId}/cancel	Cancel a scan
POST	/scans/{scanId}/files	Create a file upload
GET	/scans/{scanId}/results	List all results from a scan

#### Implementation Notes

When retrieving a scan, you'll get a results property containing the total count of scan results items. With this url you can retrieve the full paginated list of items.

#### Response Class (Status 200)

Model | Model Schema

```
{
```

# Agenda

1. Problematic
2. Internals and results
3. A community project
4. Workshop
5. Conclusion



# Modular solution to face malware infections<sup>81</sup>

- File analysis framework.
- Private, customisable.
- Central brick for incident response.
- Various usages.

# Contact

<http://irma.quarkslab.com> - [contact@quarkslab.com](mailto:contact@quarkslab.com)



<https://github.com/quarkslab/irma>

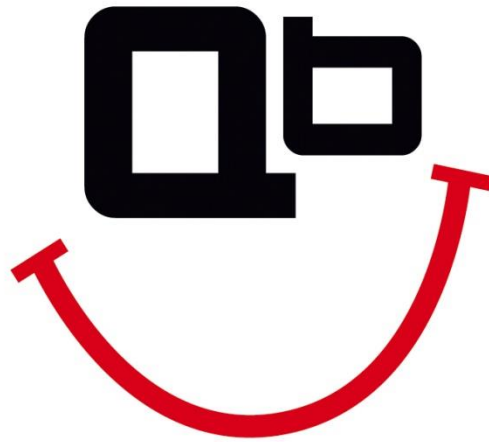


@qb\_irma

#irc

#qb\_irma@freenode





[www.quarkslab.com](http://www.quarkslab.com)

[contact@quarkslab.com](mailto:contact@quarkslab.com) | [@quarkslab](https://twitter.com/quarkslab)