cuckoo

One Flew Over the Cuckoo's Nest

Hack In The Box 2012 Amsterdam

May 24th 2012

Chapter 0x01

INTRODUCTION

Who we are

Here

- Claudio “*nex*” Guarnieri @botherder
  - Security Researcher at iSIGHT Partners
  - Core Member at The Shadowserver Foundation
  - Full Member at The Honeynet Project
  - Pizza, pasta, Ferrari
  - Cuckoo Creator and Lead Developer

Not here

- Alessandro "*jekil*" Tanasi @jekil
  - Dude from Hostmap, SecDocs
  - Cuckoo Core Developer and Fussy Bitch Engineer

- Dario "*bagode*" Fernandes
  - Google Summer of Code 2011 student
  - Cuckoo Windows components developer

Chapter 0x02

AUTOMATED MALWARE ANALYSIS

Problems

- Malwares in the wild are way too many
- Manual analysis takes a lot of time
- Static analysis requires strong skillsets
- Need to deal with packed, polymorphic, self-modifying code
- Performing dynamic analysis manually is a tedious work

SANDBOX!

Pros

- Can automate the whole analysis process
- Process high volumes of malwares
- Usable by virtually anyone
- Get actual executed code
- Can tweak to do cool sh1t
- Automating is cool
- Automating is cool
- Automating is cool

Lets you focus on more important duties

and still get paid

Cons

- Commercial solutions are very expensive
- Some portions of the malware code could be not triggered
- Environment could be detected
- Difficult to successfully automate exploit analysis
- Without proper consumption of the results, it gets useless

Preparation

- Need to define your requirements and expectations
- Need to design the analysis environment carefully
- Need to design and implement a proper use of the data and integration with other systems and storages

Ask yourself #1

- Why do I need a Sandbox?
- What do I expect to achieve?
- What information is most relevant to me or to my organization?
- Who is gonna consume the results and what for?
- How can I make it easily consumable

Ask yourself #2

- Do I want to analyze PDF exploits?
- Do I want to analyze Office exploits?
- Do I want to analyze PHP and Perl scripts?
- Do I want to analyze browsers' exploits?
- What else do I want to analyze?
- Do I want it to communicate with the outside?

Chapter 0x03

CUCKOO SANDBOX

What is it

- Automated malware analysis system
- Uses virtualization
- Easy to use
- Easy to customize
- Every single piece of it it’s Open Source!

History

- Google Summer of Code 2010
- DRG Security Innovation Grant 2011 finalist
- Google Summer of Code 2011
- Malwr.com
- Google Summer of Code 2012
- Rapid7 Magnificent7 winner of 1st round http://community.rapid7.com

powered by RAPID7

It can

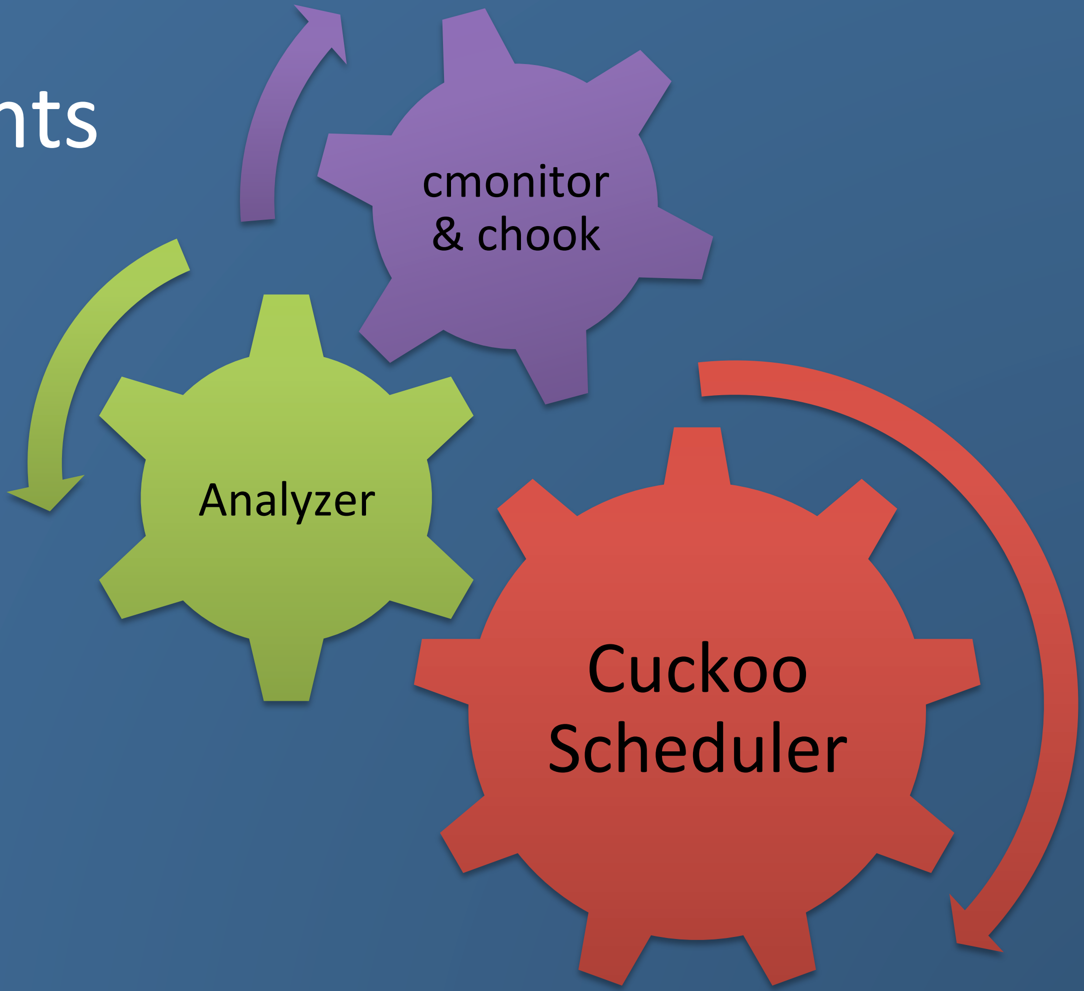
- Analyze *PE32, PDF, DOC, URLs, PHP, Perl, Python* scripts… you name it
- Be fully customized to do whatever you want
- Be integrated in larger threat intelligence frameworks

It generates

- Win32 functions calls trace
- Dropped files
- Screenshots
- Network traffic dump
- Comprehensive reports

http://www.flickr.com/photos/consonus/

Components



Scheduler

- Main component
- Dispatches the pending tasks to the pool of machines available
- Runs all the juicy modules we’re gonna see in a bit
- 100% Python

Analyzer

- Component that instruments the guest machine
- Chosen depending on the platform of the selected machine
- Only Windows now, but can support more
- Runs the malware and do stuff with it
- 100% Python

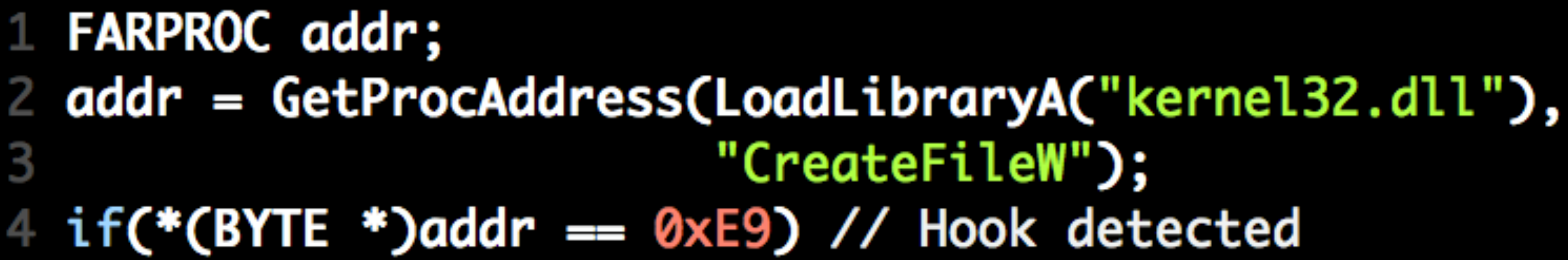
Cmonitor

- DLL using chook to install hooks on predefined win32 functions inside process memory
- Gets injected into the target process (QueueUserAPC or CreateRemoteThread)
- Logs the functions calls to files

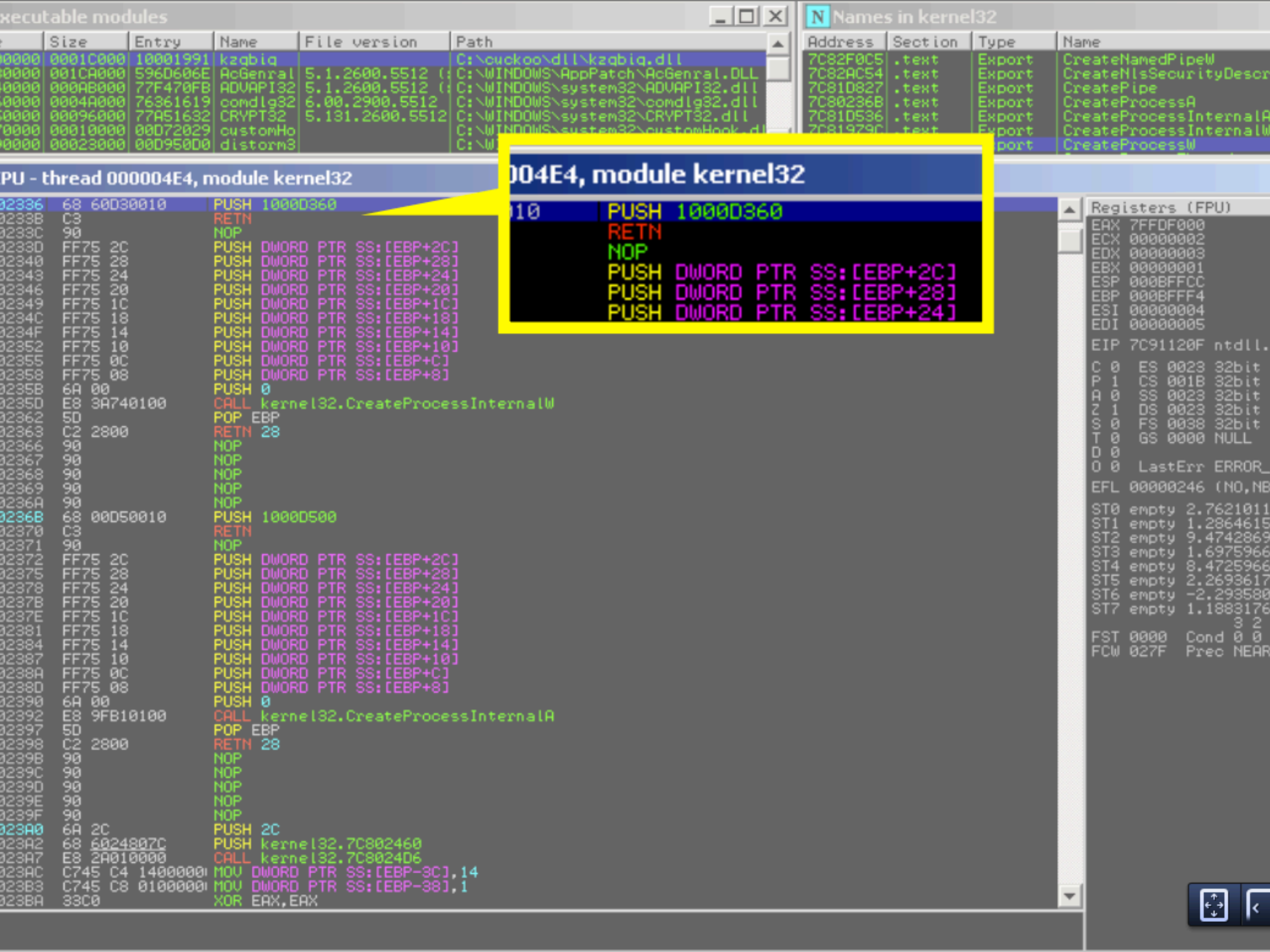
Chook

- Custom inline hooking library
- Allows definition of custom hook trampolines
- Replaced Microsoft Detours

Reason #1



```
FARPROC addr;
addr = GetProcAddress(LoadLibraryA("kernel32.dll"),
                      "CreateFileW");
if(*(BYTE *)addr == 0xE9) // Hook detected
```



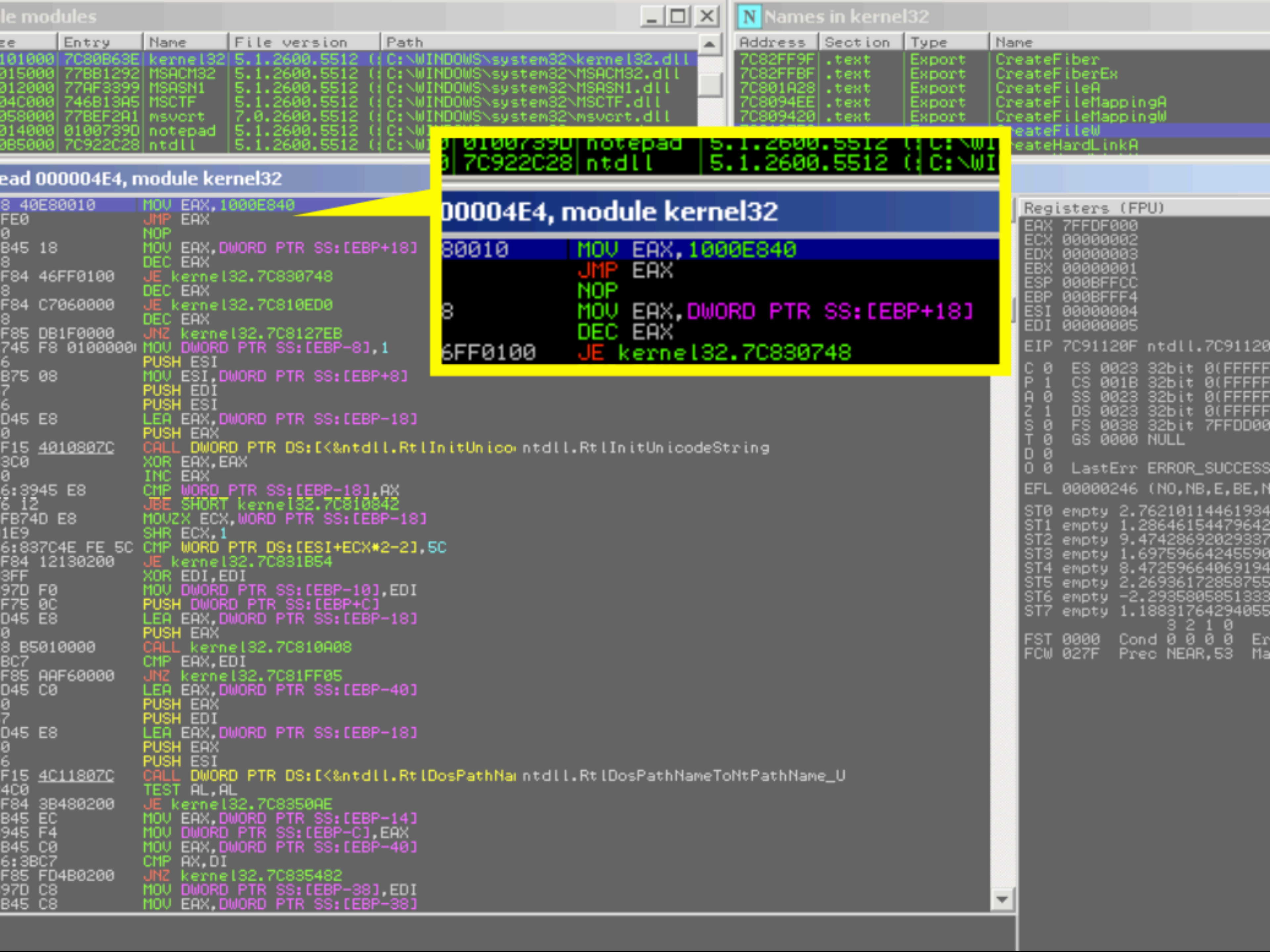
```
xecutable modules
  Size      Entry     Name      File version       Path
00000 0001C000 10001991 kzqbiq                      C:\cuckoo\dll\kzqbiq.dll
30000 001CA000 596D606E AcGenral 5.1.2600.5512 (:  C:\WINDOWS\AppPatch\AcGenral.DLL
40000 000AB000 77F470FB ADVAPI32 5.1.2600.5512 (:  C:\WINDOWS\system32\ADVAPI32.dll
60000 0004A000 76361619 comdlg32 6.00.2900.5512    C:\WINDOWS\system32\comdlg32.dll
50000 00096000 77A51632 CRYPT32  5.131.2600.5512   C:\WINDOWS\system32\CRYPT32.dll
70000 00010000 00D72029 customHo                   C:\WINDOWS\system32\customHook.dl
90000 00023000 00D950D0 distorm3                   C:\WI
```

```
Names in kernel32
Address   Section  Type    Name
7C82F0C5  .text    Export  CreateNamedPipeW
7C82AC54  .text    Export  CreateNlsSecurityDescr
7C81D827  .text    Export  CreatePipe
7C80236B  .text    Export  CreateProcessA
7C81D536  .text    Export  CreateProcessInternalA
7C81979C  .text    Export  CreateProcessInternalW
                     port  CreateProcessW
```

```
PU - thread 000004E4, module kernel32
02336  68 60D30010      PUSH 1000D360
0233B  C3               RETN
0233C  90               NOP
0233D  FF75 2C          PUSH DWORD PTR SS:[EBP+2C]
02340  FF75 28          PUSH DWORD PTR SS:[EBP+28]
02343  FF75 24          PUSH DWORD PTR SS:[EBP+24]
02346  FF75 20          PUSH DWORD PTR SS:[EBP+20]
02349  FF75 1C          PUSH DWORD PTR SS:[EBP+1C]
0234C  FF75 18          PUSH DWORD PTR SS:[EBP+18]
0234F  FF75 14          PUSH DWORD PTR SS:[EBP+14]
02352  FF75 10          PUSH DWORD PTR SS:[EBP+10]
02355  FF75 0C          PUSH DWORD PTR SS:[EBP+C]
02358  FF75 08          PUSH DWORD PTR SS:[EBP+8]
0235B  6A 00            PUSH 0
0235D  E8 3A740100      CALL kernel32.CreateProcessInternalW
02362  5D               POP EBP
02363  C2 2800          RETN 28
02366  90               NOP
02367  90               NOP
02368  90               NOP
02369  90               NOP
0236A  90               NOP
0236B  68 00D50010      PUSH 1000D500
02370  C3               RETN
02371  90               NOP
02372  FF75 2C          PUSH DWORD PTR SS:[EBP+2C]
02375  FF75 28          PUSH DWORD PTR SS:[EBP+28]
02378  FF75 24          PUSH DWORD PTR SS:[EBP+24]
0237B  FF75 20          PUSH DWORD PTR SS:[EBP+20]
0237E  FF75 1C          PUSH DWORD PTR SS:[EBP+1C]
02381  FF75 18          PUSH DWORD PTR SS:[EBP+18]
02384  FF75 14          PUSH DWORD PTR SS:[EBP+14]
02387  FF75 10          PUSH DWORD PTR SS:[EBP+10]
0238A  FF75 0C          PUSH DWORD PTR SS:[EBP+C]
0238D  FF75 08          PUSH DWORD PTR SS:[EBP+8]
02390  6A 00            PUSH 0
02392  E8 9FB10100      CALL kernel32.CreateProcessInternalA
02397  5D               POP EBP
02398  C2 2800          RETN 28
0239B  90               NOP
0239C  90               NOP
0239D  90               NOP
0239E  90               NOP
0239F  90               NOP
023A0  6A 2C            PUSH 2C
023A2  68 6024807C      PUSH kernel32.7C802460
023A7  E8 2A010000      CALL kernel32.7C8024D6
023AC  C745 C4 1400000  MOV DWORD PTR SS:[EBP-3C],14
023B3  C745 C8 0100000  MOV DWORD PTR SS:[EBP-38],1
023BA  33C0             XOR EAX,EAX
```

```
004E4, module kernel32
10   PUSH 1000D360
     RETN
     NOP
     PUSH DWORD PTR SS:[EBP+2C]
     PUSH DWORD PTR SS:[EBP+28]
     PUSH DWORD PTR SS:[EBP+24]
```

```
Registers (FPU)
EAX 7FFDF000
ECX 00000002
EDX 00000003
EBX 00000001
ESP 000BFFCC
EBP 000BFFF4
ESI 00000004
EDI 00000005

EIP 7C91120F ntdll.

C 0  ES 0023 32bit
P 1  CS 001B 32bit
A 0  SS 0023 32bit
Z 1  DS 0023 32bit
S 0  FS 0038 32bit
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_

EFL 00000246 (NO,NB

ST0 empty 2.7621011
ST1 empty 1.2864615
ST2 empty 9.4742869
ST3 empty 1.6975966
ST4 empty 8.4725966
ST5 empty 2.2693617
ST6 empty -2.293580
ST7 empty 1.1883176
               3 2
FST 0000  Cond 0 0
FCW 027F  Prec NEAR
```



```
ze      Entry    Name     File version        Path
101000  7C80B63E kernel32 5.1.2600.5512 (:  C:\WINDOWS\system32\kernel32.dll
015000  77BB1292 MSACM32  5.1.2600.5512 (:  C:\WINDOWS\system32\MSACM32.dll
012000  77AF3399 MSASN1   5.1.2600.5512 (:  C:\WINDOWS\system32\MSASN1.dll
04C000  746B13A5 MSCTF    5.1.2600.5512 (:  C:\WINDOWS\system32\MSCTF.dll
058000  77BEF2A1 msvcrt   7.0.2600.5512 (:  C:\WINDOWS\system32\msvcrt.dll
014000  0100739D notepad  5.1.2600.5512 (:  C:\WI
0B5000  7C922C28 ntdll    5.1.2600.5512 (:  C:\WI
```

```
Address   Section Type   Name
7C82FF9F  .text   Export CreateFiber
7C82FFBF  .text   Export CreateFiberEx
7C801A28  .text   Export CreateFileA
7C8094EE  .text   Export CreateFileMappingA
7C809420  .text   Export CreateFileMappingW
                         reateFileW
                         reateHardLinkA
```

```
8 40E80010         MOV EAX,1000E840
FE0                JMP EAX
0                  NOP
B45 18             MOV EAX,DWORD PTR SS:[EBP+18]
8                  DEC EAX
F84 46FF0100       JE kernel32.7C830748
8                  DEC EAX
F84 C7060000       JE kernel32.7C810ED0
8                  DEC EAX
F85 DB1F0000       JNZ kernel32.7C8127EB
745 F8 0100000     MOV DWORD PTR SS:[EBP-8],1
6                  PUSH ESI
B75 08             MOV ESI,DWORD PTR SS:[EBP+8]
7                  PUSH EDI
6                  PUSH ESI
D45 E8             LEA EAX,DWORD PTR SS:[EBP-18]
0                  PUSH EAX
F15 4010807C       CALL DWORD PTR DS:[<&ntdll.RtlInitUnico ntdll.RtlInitUnicodeString
3C0                XOR EAX,EAX
0                  INC EAX
6:3945 E8          CMP WORD PTR SS:[EBP-18],AX
6 12               JBE SHORT kernel32.7C810842
FB74D E8           MOVZX ECX,WORD PTR SS:[EBP-18]
1E9                SHR ECX,1
6:837C4E FE 5C     CMP WORD PTR DS:[ESI+ECX*2-2],5C
F84 12130200       JE kernel32.7C831B54
3FF                XOR EDI,EDI
97D F0             MOV DWORD PTR SS:[EBP-10],EDI
F75 0C             PUSH DWORD PTR SS:[EBP+C]
D45 E8             LEA EAX,DWORD PTR SS:[EBP-18]
0                  PUSH EAX
8 B5010000         CALL kernel32.7C810A08
BC7                CMP EAX,EDI
F85 AAF60000       JNZ kernel32.7C81FF05
D45 C0             LEA EAX,DWORD PTR SS:[EBP-40]
0                  PUSH EAX
7                  PUSH EDI
D45 E8             LEA EAX,DWORD PTR SS:[EBP-18]
0                  PUSH EAX
6                  PUSH ESI
F15 4C11807C       CALL DWORD PTR DS:[<&ntdll.RtlDosPathNa ntdll.RtlDosPathNameToNtPathName_U
4C0                TEST AL,AL
F84 3B480200       JE kernel32.7C8350AE
B45 EC             MOV EAX,DWORD PTR SS:[EBP-14]
945 F4             MOV DWORD PTR SS:[EBP-C],EAX
B45 C0             MOV EAX,DWORD PTR SS:[EBP-40]
6:3BC7             CMP AX,DI
F85 FD4B0200       JNZ kernel32.7C835482
97D C8             MOV DWORD PTR SS:[EBP-38],EDI
B45 C8             MOV EAX,DWORD PTR SS:[EBP-38]
```

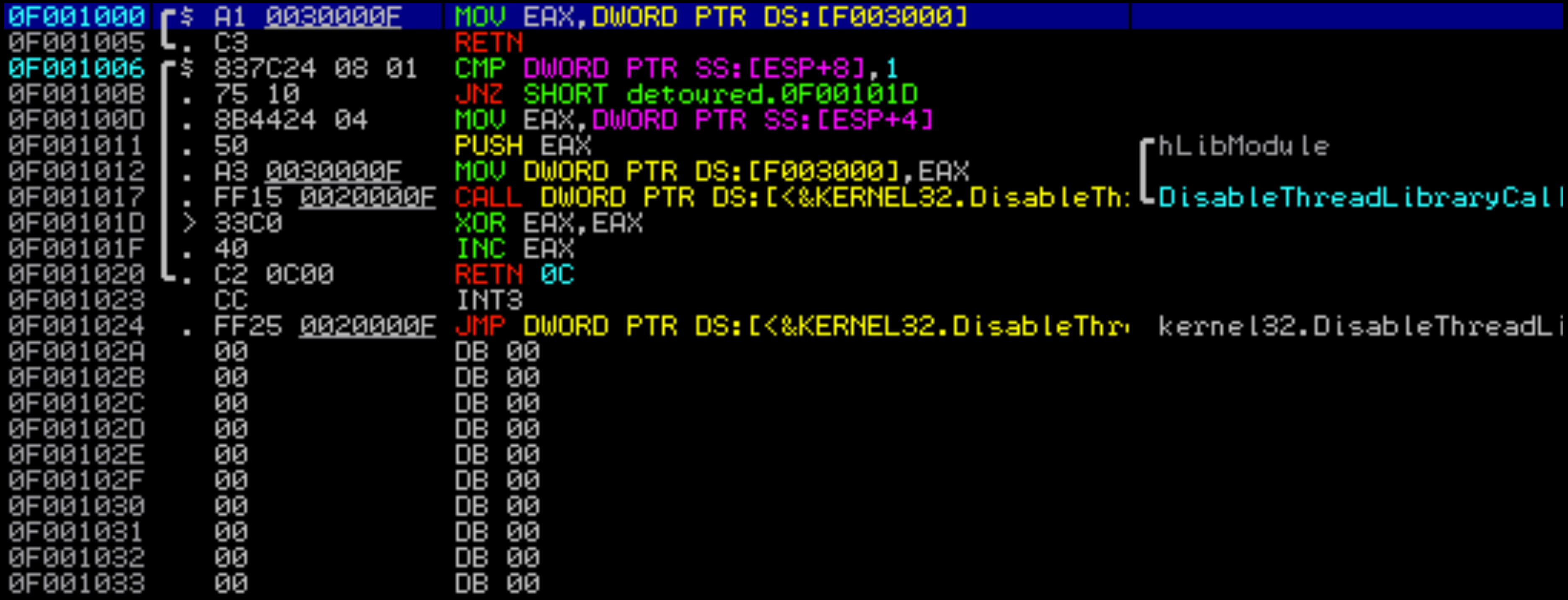
```
0 0100739D notepad  5.1.2600.5512 (: C:\WI
0 7C922C28 ntdll    5.1.2600.5512 (: C:\WI
```

00004E4, module kernel32

```
80010     MOV EAX,1000E840
          JMP EAX
          NOP
8         MOV EAX,DWORD PTR SS:[EBP+18]
          DEC EAX
6FF0100   JE kernel32.7C830748
```

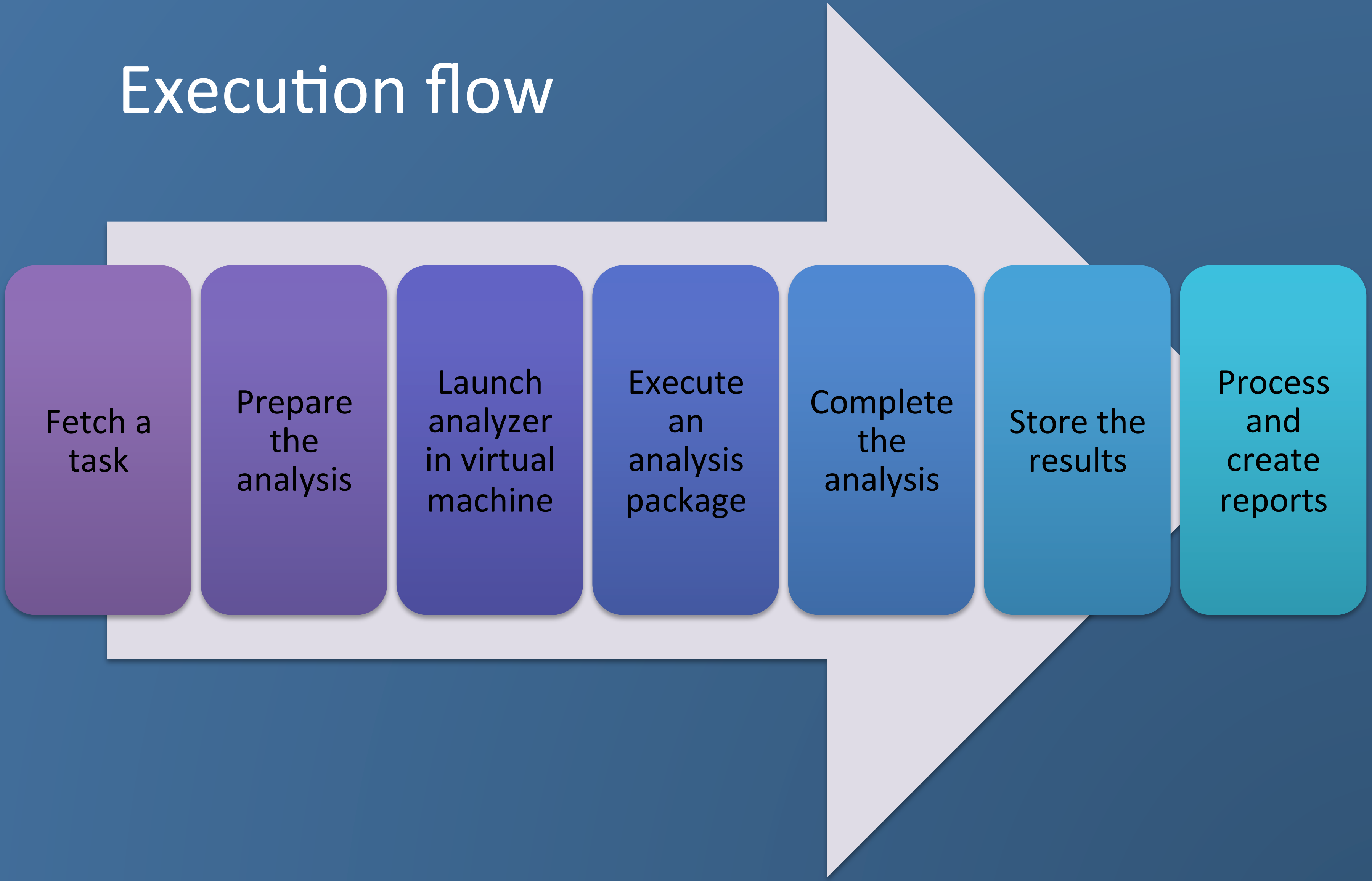
```
Registers (FPU)
EAX 7FFDF000
ECX 00000002
EDX 00000003
EBX 00000001
ESP 000BFFCC
EBP 000BFFF4
ESI 00000004
EDI 00000005

EIP 7C91120F ntdll.7C91120

C 0  ES 0023 32bit 0(FFFFF
P 1  CS 001B 32bit 0(FFFFF
A 0  SS 0023 32bit 0(FFFFF
Z 1  DS 0023 32bit 0(FFFFF
S 0  FS 0038 32bit 7FFDD00
T 0  GS 0000 NULL
D 0
O 0  LastErr ERROR_SUCCESS

EFL 00000246 (NO,NB,E,BE,N

ST0 empty 2.76210114461934
ST1 empty 1.28646154479642
ST2 empty 9.47428692029337
ST3 empty 1.69759664245590
ST4 empty 8.47259664069194
ST5 empty 2.26936172858755
ST6 empty -2.29358058513333
ST7 empty 1.18831764294055
                3 2 1 0
FST 0000  Cond 0 0 0 0  Er
FCW 027F  Prec NEAR,53  Ma
```

Reason #2



```
0F001000  $ A1 0030000F     MOV EAX,DWORD PTR DS:[F003000]
0F001005  . C3              RETN
0F001006  $ 837C24 08 01    CMP DWORD PTR SS:[ESP+8],1
0F00100B  . 75 10           JNZ SHORT detoured.0F00101D
0F00100D  . 8B4424 04       MOV EAX,DWORD PTR SS:[ESP+4]
0F001011  . 50              PUSH EAX                                       hLibModule
0F001012  . A3 0030000F     MOV DWORD PTR DS:[F003000],EAX
0F001017  . FF15 0020000F   CALL DWORD PTR DS:[<&KERNEL32.DisableTh:      DisableThreadLibraryCal
0F00101D  > 33C0            XOR EAX,EAX
0F00101F  . 40              INC EAX
0F001020  . C2 0C00         RETN 0C
0F001023    CC              INT3
0F001024  . FF25 0020000F   JMP DWORD PTR DS:[<&KERNEL32.DisableThr      kernel32.DisableThreadLi
0F00102A    00              DB 00
0F00102B    00              DB 00
0F00102C    00              DB 00
0F00102D    00              DB 00
0F00102E    00              DB 00
0F00102F    00              DB 00
0F001030    00              DB 00
0F001031    00              DB 00
0F001032    00              DB 00
0F001033    00              DB 00
```

Execution flow



Fetch a task

Prepare the analysis

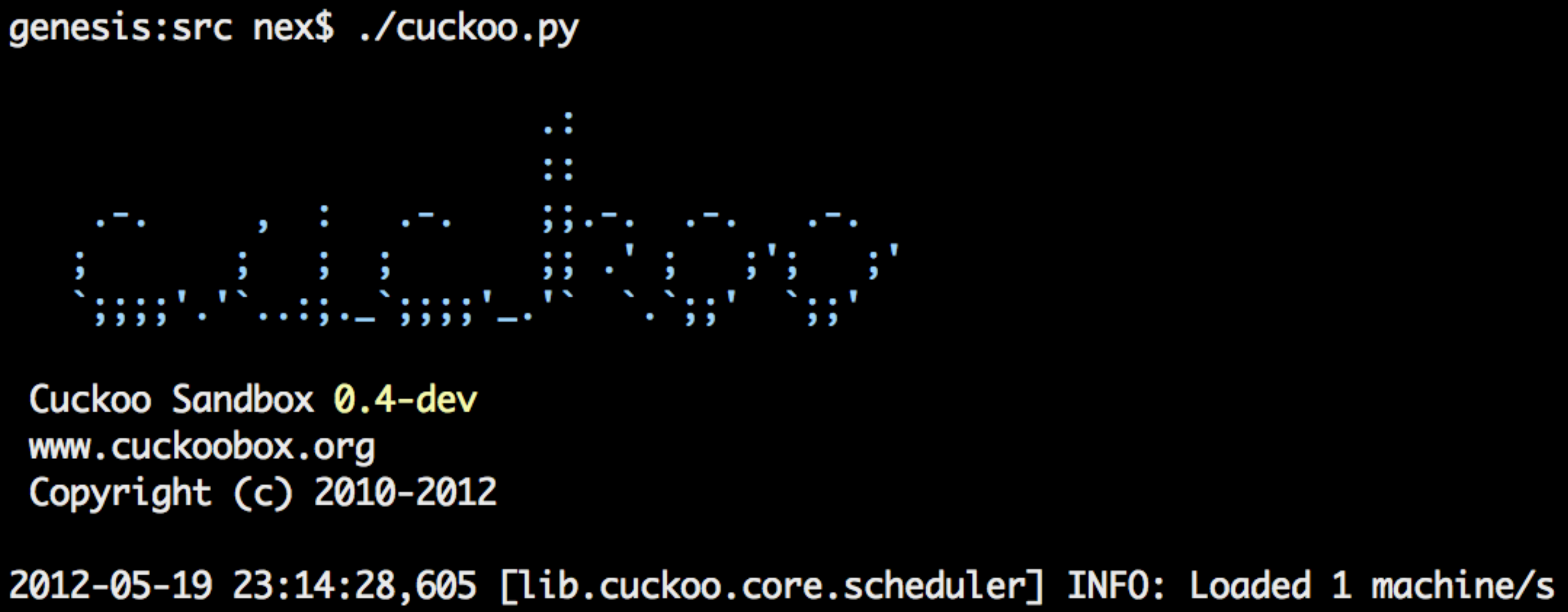
Launch analyzer in virtual machine

Execute an analysis package

Complete the analysis

Store the results

Process and create reports



```
genesis:src nex$ ./cuckoo.py

                          .:
                          ::
    .-.     ,  :   .-.    ;;.-.  .-.   .-.
   ;       ;   ;  ;       ;; .' ;   ;';   ;'
   `;;;;'.'`..:;._`;;;;'_.'`  `.`;;'  `;;'

 Cuckoo Sandbox 0.4-dev
 www.cuckoobox.org
 Copyright (c) 2010-2012

2012-05-19 23:14:28,605 [lib.cuckoo.core.scheduler] INFO: Loaded 1 machine/s
```

Submission

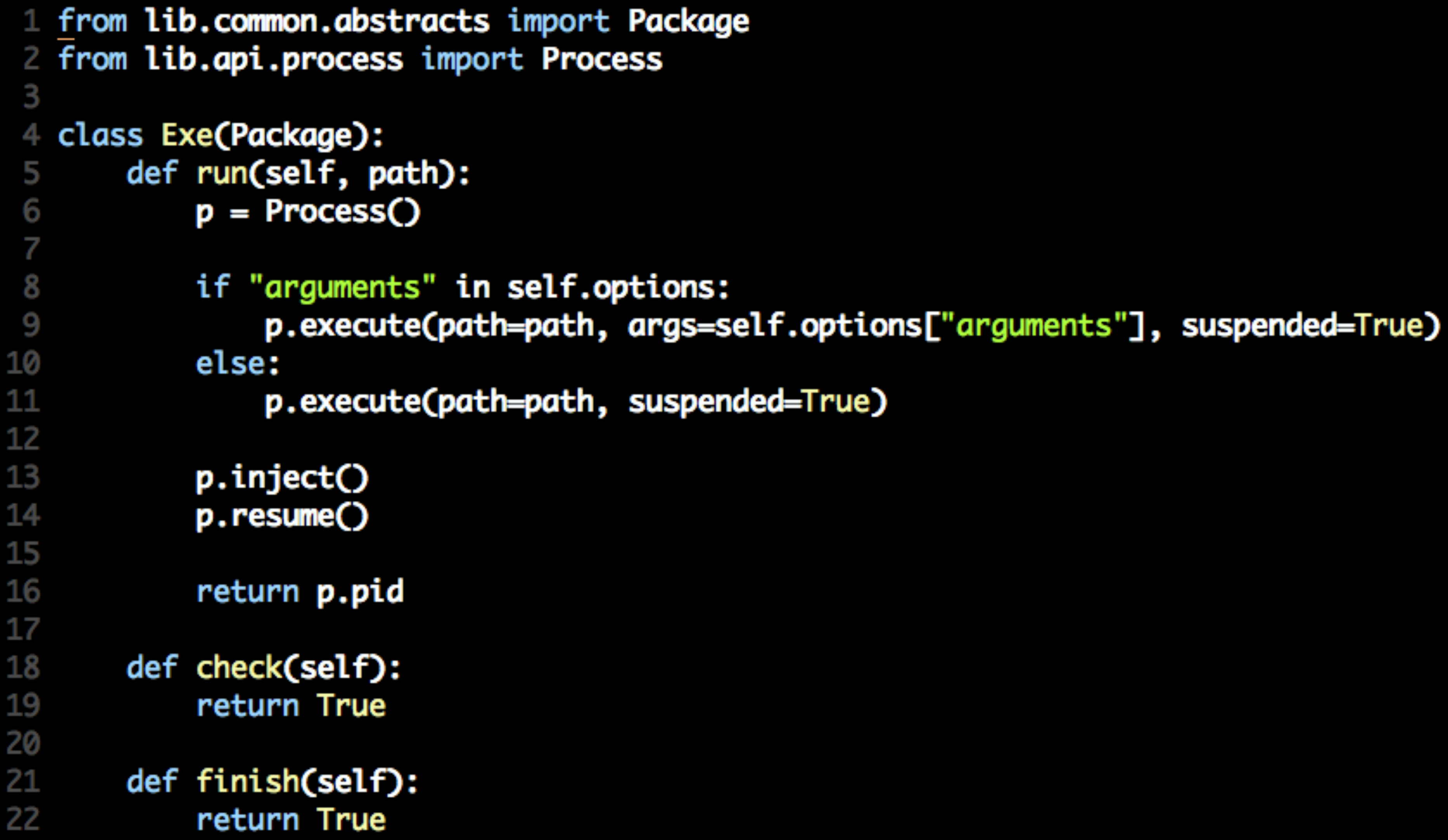
- From command-line, Python API or SQLite DB
- Specify file path
- Specify analysis package and its options
- Specify machine to be used or operating system
- Specify timeout, priority

Modules & Customization

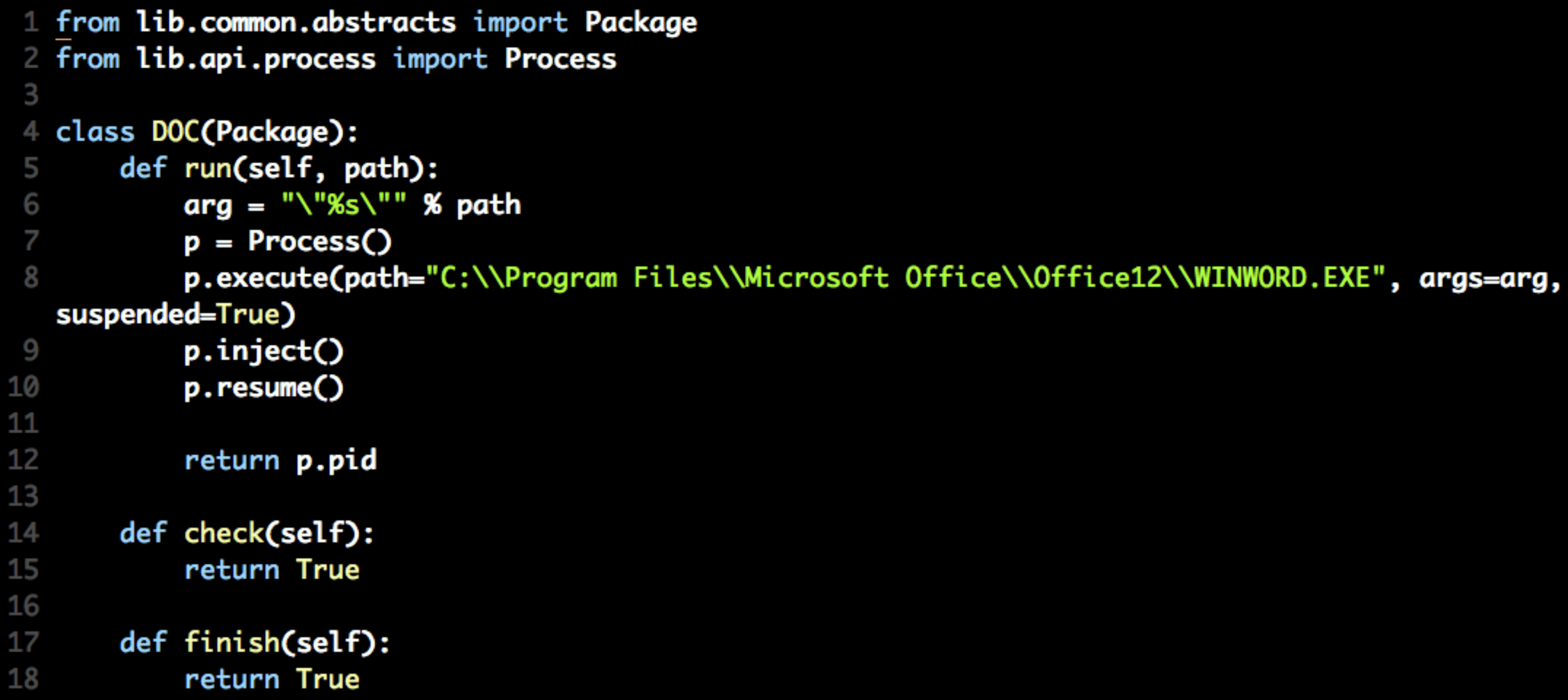
- Analysis Packages
- Machine Managers
- Processing
- Reporting
- Signatures

Analysis Packages

- Python classes ☺
- Defines how the analyzer should start and interact with the malware
- Specified at submission or selected upon file type
- Can create as many as you want and do whatever you want



```python
from lib.common.abstracts import Package
from lib.api.process import Process

class Exe(Package):
    def run(self, path):
        p = Process()

        if "arguments" in self.options:
            p.execute(path=path, args=self.options["arguments"], suspended=True)
        else:
            p.execute(path=path, suspended=True)

        p.inject()
        p.resume()

        return p.pid

    def check(self):
        return True

    def finish(self):
        return True
```



```python
from lib.common.abstracts import Package
from lib.api.process import Process

class DOC(Package):
    def run(self, path):
        arg = "\"%s\"" % path
        p = Process()
        p.execute(path="C:\\Program Files\\Microsoft Office\\Office12\\WINWORD.EXE", args=arg, suspended=True)
        p.inject()
        p.resume()

        return p.pid

    def check(self):
        return True

    def finish(self):
        return True
```

DEMO!

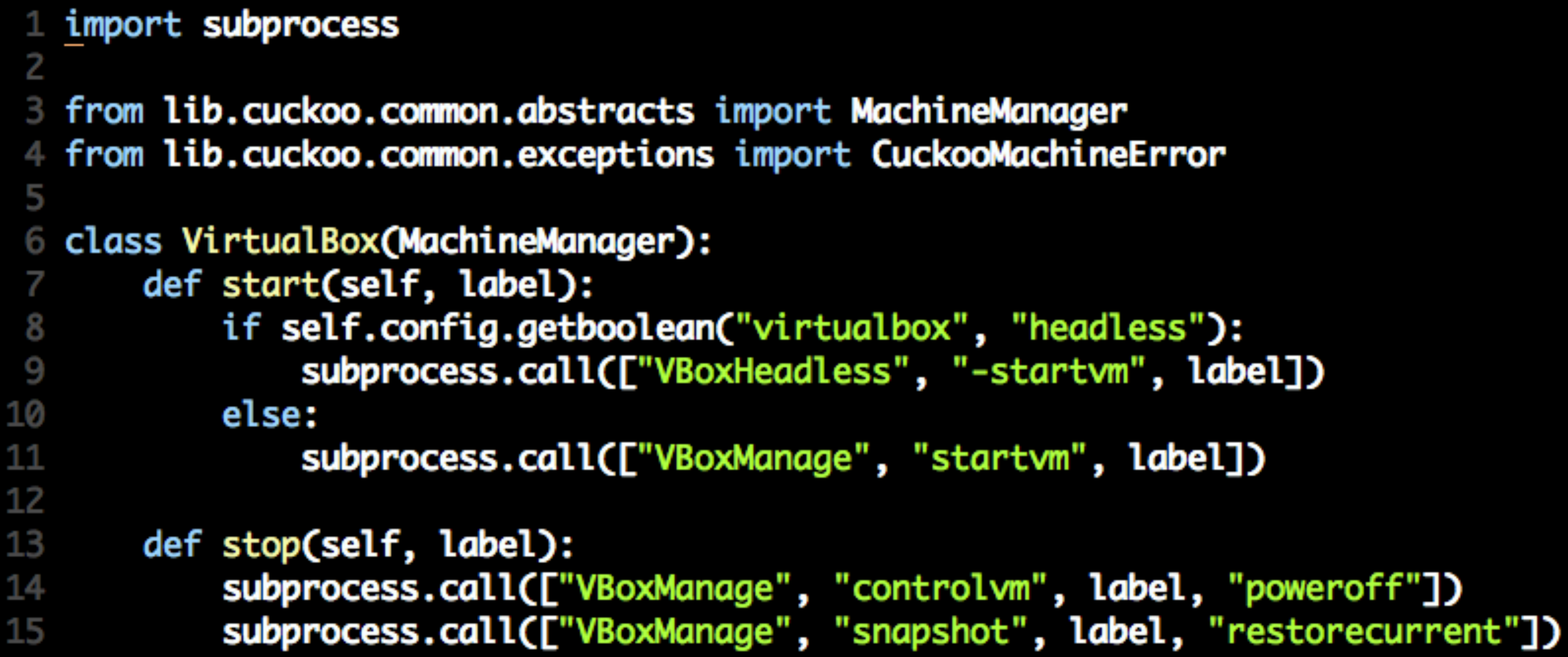
Other examples

- Honeyclient?
- Banking trojan analyzer
- USB Honeypot
- Up to you…

```
genesis:src nex$ tree -d modules/
modules/
├── machinemanagers
├── processing
├── reporting
└── signatures

4 directories
```

Machine Managers

- Yes, Python classes ☺
- Define interaction with virtualization software

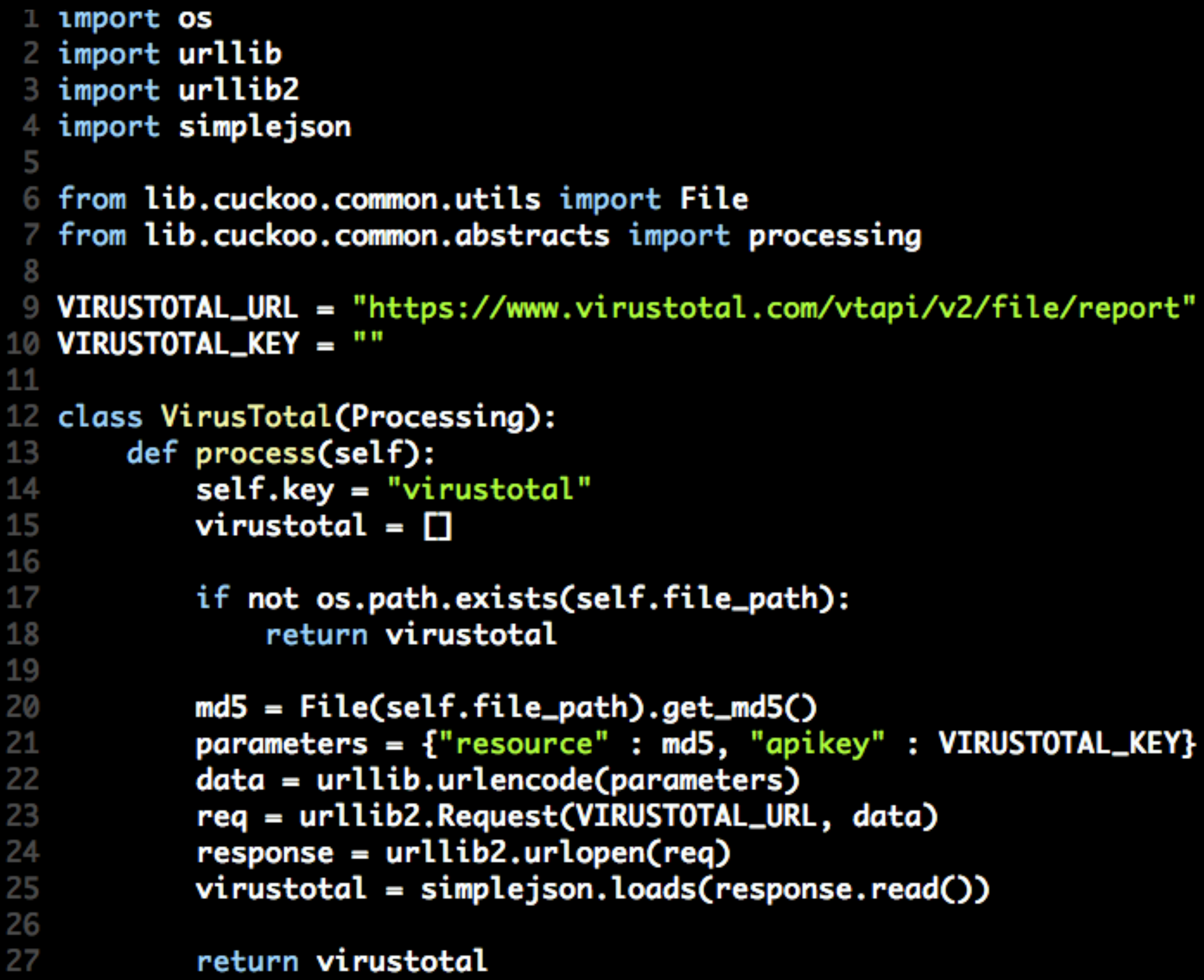


```python
import subprocess

from lib.cuckoo.common.abstracts import MachineManager
from lib.cuckoo.common.exceptions import CuckooMachineError

class VirtualBox(MachineManager):
    def start(self, label):
        if self.config.getboolean("virtualbox", "headless"):
            subprocess.call(["VBoxHeadless", "-startvm", label])
        else:
            subprocess.call(["VBoxManage", "startvm", label])

    def stop(self, label):
        subprocess.call(["VBoxManage", "controlvm", label, "poweroff"])
        subprocess.call(["VBoxManage", "snapshot", label, "restorecurrent"])
```

Processing

- Python classes, again 😐
- Modules used to generate a container of normalized information on the analysis
- Can create as many as you want

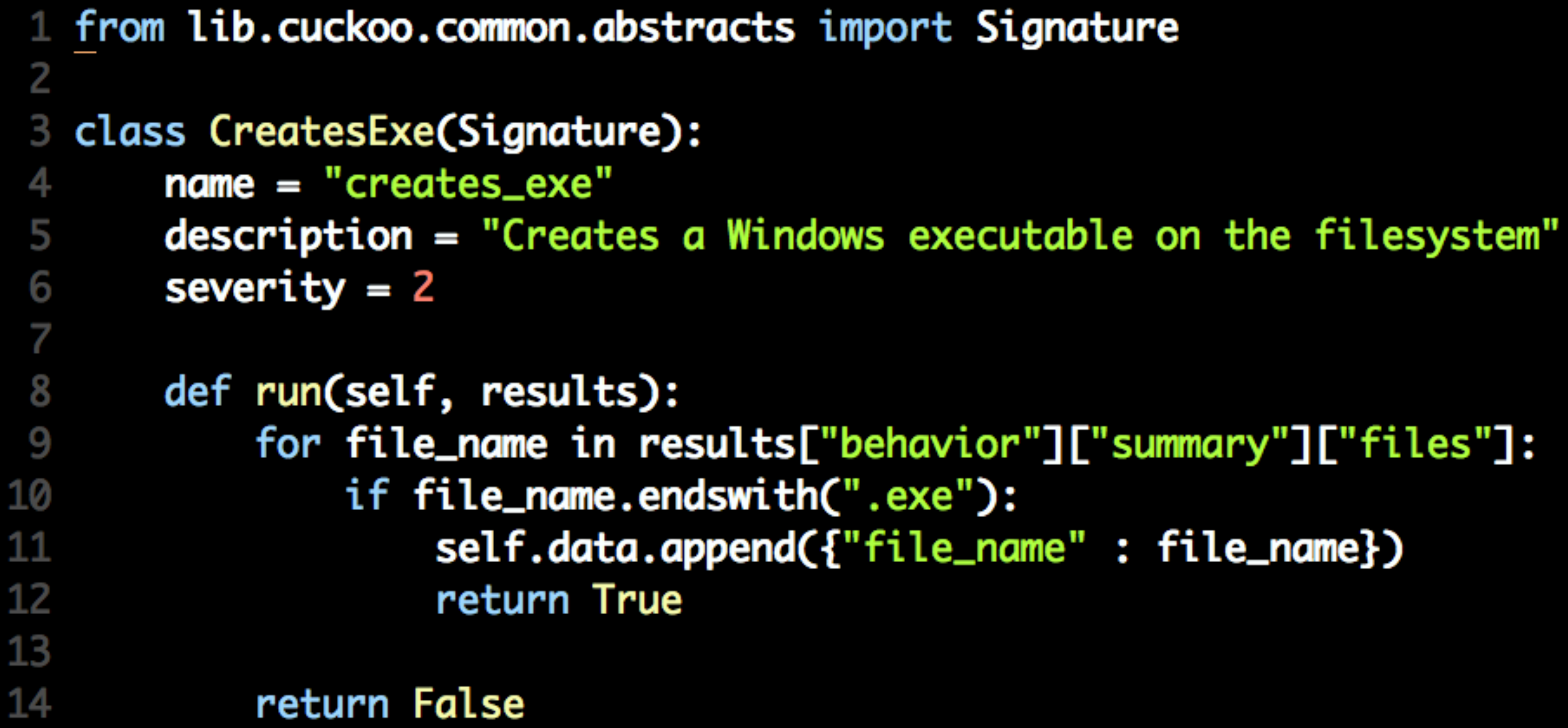
```python
from lib.cuckoo.common.utils import File
from lib.cuckoo.common.abstracts import Processing

class FileAnalysis(Processing):
    def run(self):
        self.key = "file"
        file_info = File(self.file_path).get_all()
        return file_info
```



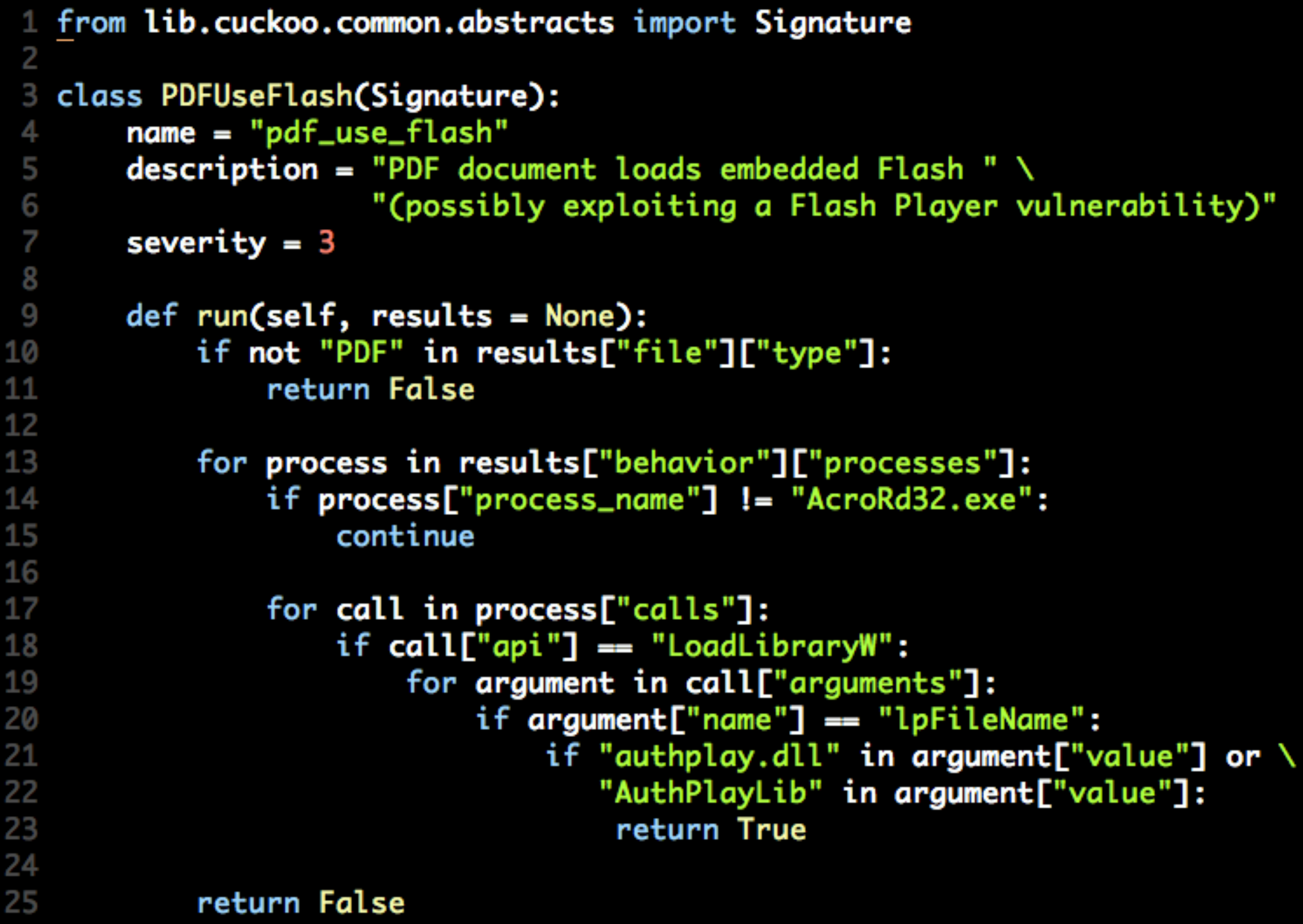
```python
import os
import urllib
import urllib2
import simplejson

from lib.cuckoo.common.utils import File
from lib.cuckoo.common.abstracts import processing

VIRUSTOTAL_URL = "https://www.virustotal.com/vtapi/v2/file/report"
VIRUSTOTAL_KEY = ""

class VirusTotal(Processing):
    def process(self):
        self.key = "virustotal"
        virustotal = []

        if not os.path.exists(self.file_path):
            return virustotal

        md5 = File(self.file_path).get_md5()
        parameters = {"resource" : md5, "apikey" : VIRUSTOTAL_KEY}
        data = urllib.urlencode(parameters)
        req = urllib2.Request(VIRUSTOTAL_URL, data)
        response = urllib2.urlopen(req)
        virustotal = simplejson.loads(response.read())

        return virustotal
```

Signatures

- Python classes!
- Look for patterns or specific events
- Assign them a description and severity level
- Give context to the reports
- Help non-malware experts understand
- Can be used to receive email alerts



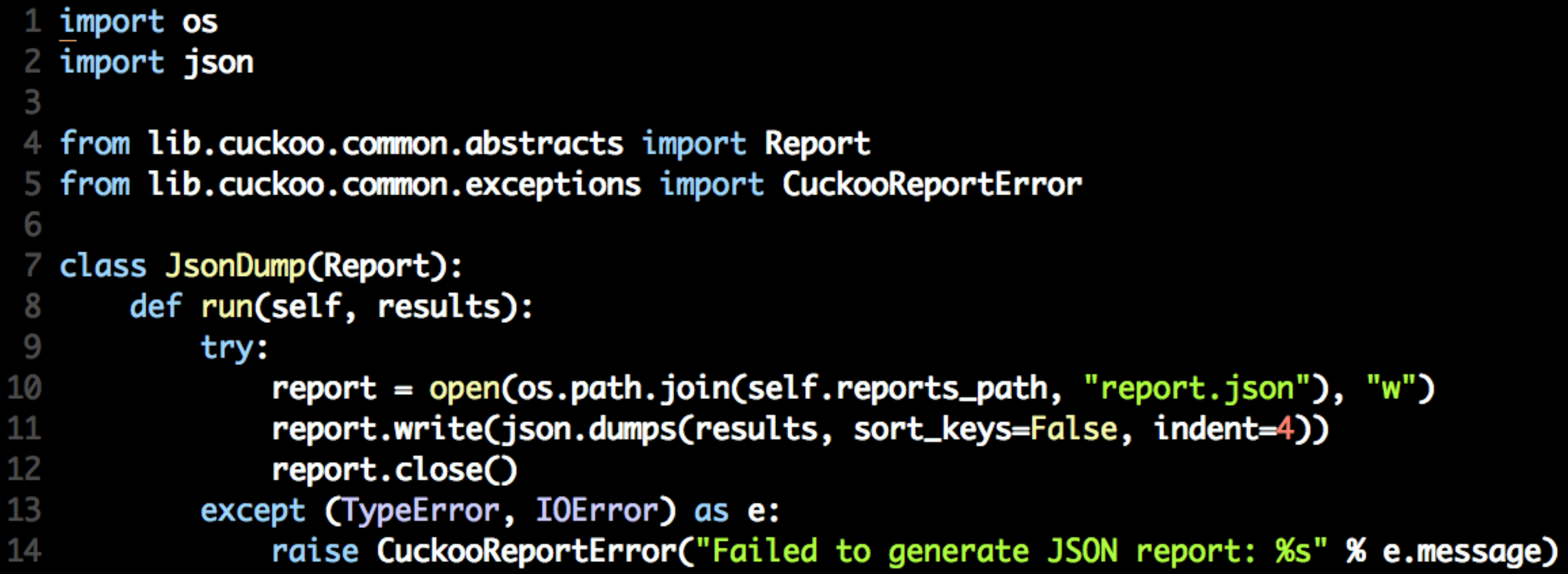
```python
from lib.cuckoo.common.abstracts import Signature

class CreatesExe(Signature):
    name = "creates_exe"
    description = "Creates a Windows executable on the filesystem"
    severity = 2

    def run(self, results):
        for file_name in results["behavior"]["summary"]["files"]:
            if file_name.endswith(".exe"):
                self.data.append({"file_name" : file_name})
                return True

        return False
```



```python
from lib.cuckoo.common.abstracts import Signature

class PDFUseFlash(Signature):
    name = "pdf_use_flash"
    description = "PDF document loads embedded Flash " \
                  "(possibly exploiting a Flash Player vulnerability)"
    severity = 3

    def run(self, results = None):
        if not "PDF" in results["file"]["type"]:
            return False

        for process in results["behavior"]["processes"]:
            if process["process_name"] != "AcroRd32.exe":
                continue

            for call in process["calls"]:
                if call["api"] == "LoadLibraryW":
                    for argument in call["arguments"]:
                        if argument["name"] == "lpFileName":
                            if "authplay.dll" in argument["value"] or \
                               "AuthPlayLib" in argument["value"]:
                                return True

        return False
```

Reporting

- OMG Python classes ☹
- Use the normalized results and do something with them
- Can create as many as you want



```python
import os
import json

from lib.cuckoo.common.abstracts import Report
from lib.cuckoo.common.exceptions import CuckooReportError

class JsonDump(Report):
    def run(self, results):
        try:
            report = open(os.path.join(self.reports_path, "report.json"), "w")
            report.write(json.dumps(results, sort_keys=False, indent=4))
            report.close()
        except (TypeError, IOError) as e:
            raise CuckooReportError("Failed to generate JSON report: %s" % e.message)
```

or mongo!

Community Effort

- Create a community repository for sharing modules & signatures
- Expand our line-up of developers and contributors
- Make Malwr.com a major community resource for malware research

Future Work

- A full-fledged web interface
- Improve Windows analysis components
- Support for other operating systems, Mac OS X?
- Support native machines

Websites

- http://cuckoosandbox.org
- http://github.com/cuckoobox/cuckoo
- http://blog.cuckoobox.org
- http://malwr.com
- http://www.honeynet.org

?

claudio@shadowserver.org

THANK YOU!

NOW LET’S GET SOME LUNCH!