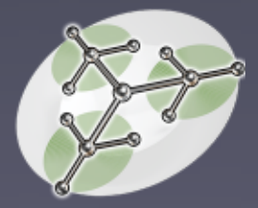
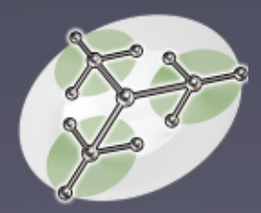
4 x 5: Reverse Engineering Automation with Python

Ero Carrera
ero.carrera@sabre-security.com
Sabre Security GmbH



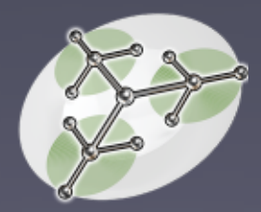
The 4 x 5...

- pefile
- pydasm
- ida2sql
- vxclass



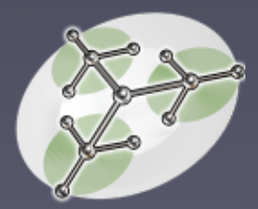
pefile - 1.2.6

http://dkbza.org/pefile.html



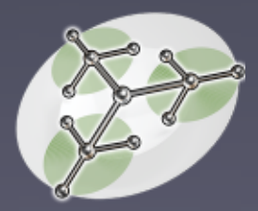
pefile

- *pefile* is a multi-platform full Python module intended for handling PE Files.
- *pefile* requires some basic understanding of the layout of a PE file. Armed with it it's possible to explore nearly every single feature of the file.



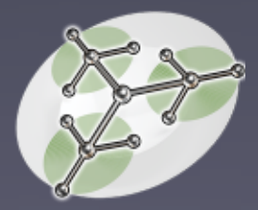
Features

- Loading a PE file
- Modifying and writing back to the PE image
- Header Inspection
- Sections analysis
- Retrieving data
- Warnings for suspicious values
- Packer detection with PEiD's signatures
- PEiD signature generation



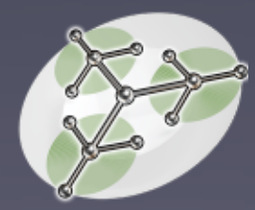
Sections Inspection

```
>>> import pefile

>>> pe =  pefile.PE('/path/to/pefile.exe')

>>> for section in pe.sections:
...  print (section.Name,
        hex(section.VirtualAddress),
        hex(section.Misc_VirtualSize),
        section.SizeOfRawData )
...
('.text', '0x1000L', '0x6D72L', 28160L)
('.data', '0x8000L', '0x1BA8L', 1536L)
('.rsrc', '0xA000L', '0x8948L', 35328L)
```



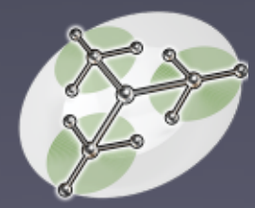
Imports

```
>>> for entry in pe.DIRECTORY_ENTRY_IMPORT:
...   print entry.dll
...   for imp in entry.imports:
...     print '\t', hex(imp.address), imp.name
...
comdlg32.dll
        0x10012A0L PageSetupDlgW
        0x10012A4L FindTextW
        0x10012A8L PrintDlgExW
        [snip]
SHELL32.dll
        0x1001154L DragFinish
        0x1001158L DragQueryFileW
```



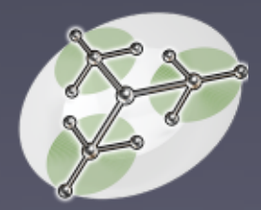
Exports

```
>>> for exp in pe.DIRECTORY_ENTRY_EXPORT.symbols:
...   print hex(pe.OPTIONAL_HEADER.ImageBase +
       exp.address), exp.name
0x77e72ada ActivateActCtx
0x77e682c2 AddAtomA
0x77e6d39f AddAtomW
0x77ec5b2d AddConsoleAliasA
0x77ec5af6 AddConsoleAliasW
0x77eb2a10 AddLocalAlternateComputerNameA
0x77eb28fb AddLocalAlternateComputerNameW
0x77ec0ffa AddRefActCtx
```



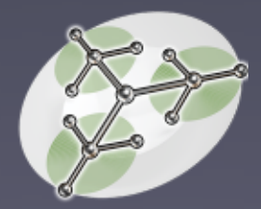
Full Dump

```
>>> print pe.dump_info()
```



pefile 1.2.6: What's new

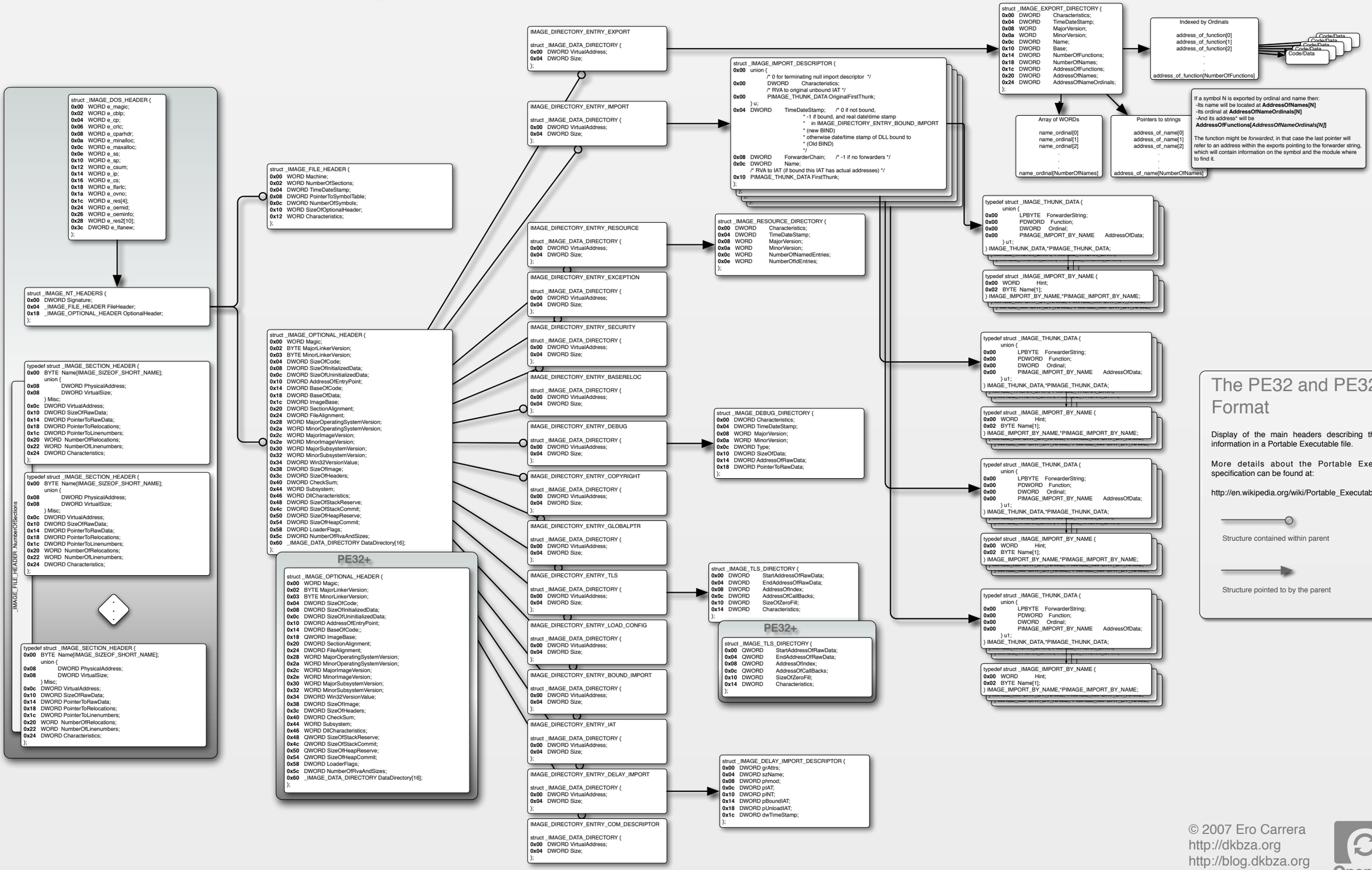
- Multiple bugfixes. Thanks to Gera (from Core), Danny Quist, Valsmith, Tim Ebringer and everybody else contributing
- Added warning messages for suspicious values
- Added calculation of section entropy
- pefile can parse and generate PEiD signatures



Parsing Warnings

- Headers of packed and manually modified PE files sometimes contain invalid values
- IDA, ProcDump, OllyDdg, ImpRec, PEiD and a bunch of other tools barf on some PEs
- The warnings give hints as to which values might be causing the indigestion

Portable Executable Format Layout



The PE32 and PE32+ Format

Display of the main headers describing the basic information in a Portable Executable file.

More details about the Portable Executable specification can be found at:

http://en.wikipedia.org/wiki/Portable_Executable

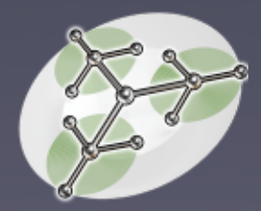
Structure contained within parent

Structure pointed to by the parent

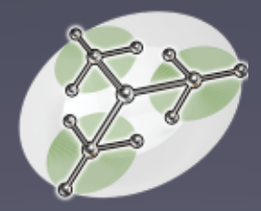
OpenRCE.org

Evilness...

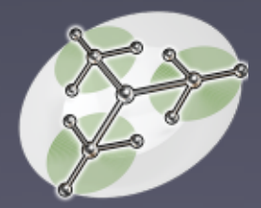
- NumberOfRvaAndSizes
- SizeOfOptionalHeader
- FileAlignment
- SizeOfRawData
- Delayed Imports
- Invalid Directories



- Error parsing section 1. SizeOfRawData is larger than file.
- Suspicious value found parsing section 1. VirtualAddress is beyond 0x10000000.
- Error parsing section 1. Suspicious value for FileAlignment in the Optional Header. Normally the PointerToRawData entry of the sections' structures is a multiple of FileAlignment, this might imply the file is trying to confuse tools which parse this incorrectly
- Error parsing section 2. SizeOfRawData is larger than file.
- Suspicious value found parsing section 2. VirtualAddress is beyond 0x10000000.
- Error parsing the Import directory at RVA: 0xd000
- Suspicious value found parsing section 1. VirtualSize is extremely large > 256MiB.



PEiD Signature Parsing



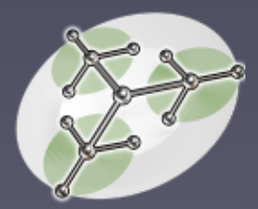
Loading PEiD signatures

```
import peutils
signatures = peutils.SignatureDatabase(
   '/path/to/signature.txt')
```

```
signatures = peutils.SignatureDatabase(
   'http://url.to/signature/file.txt')
```

```
signatures = peutils.SignatureDatabase(
   data='/path/to/signature/file.txt')
```

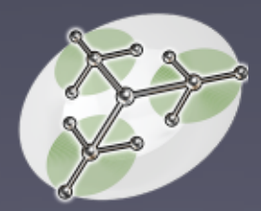
```
sig.load('/Users/ero/Devel/pefile/userdb-
extra.txt')
```



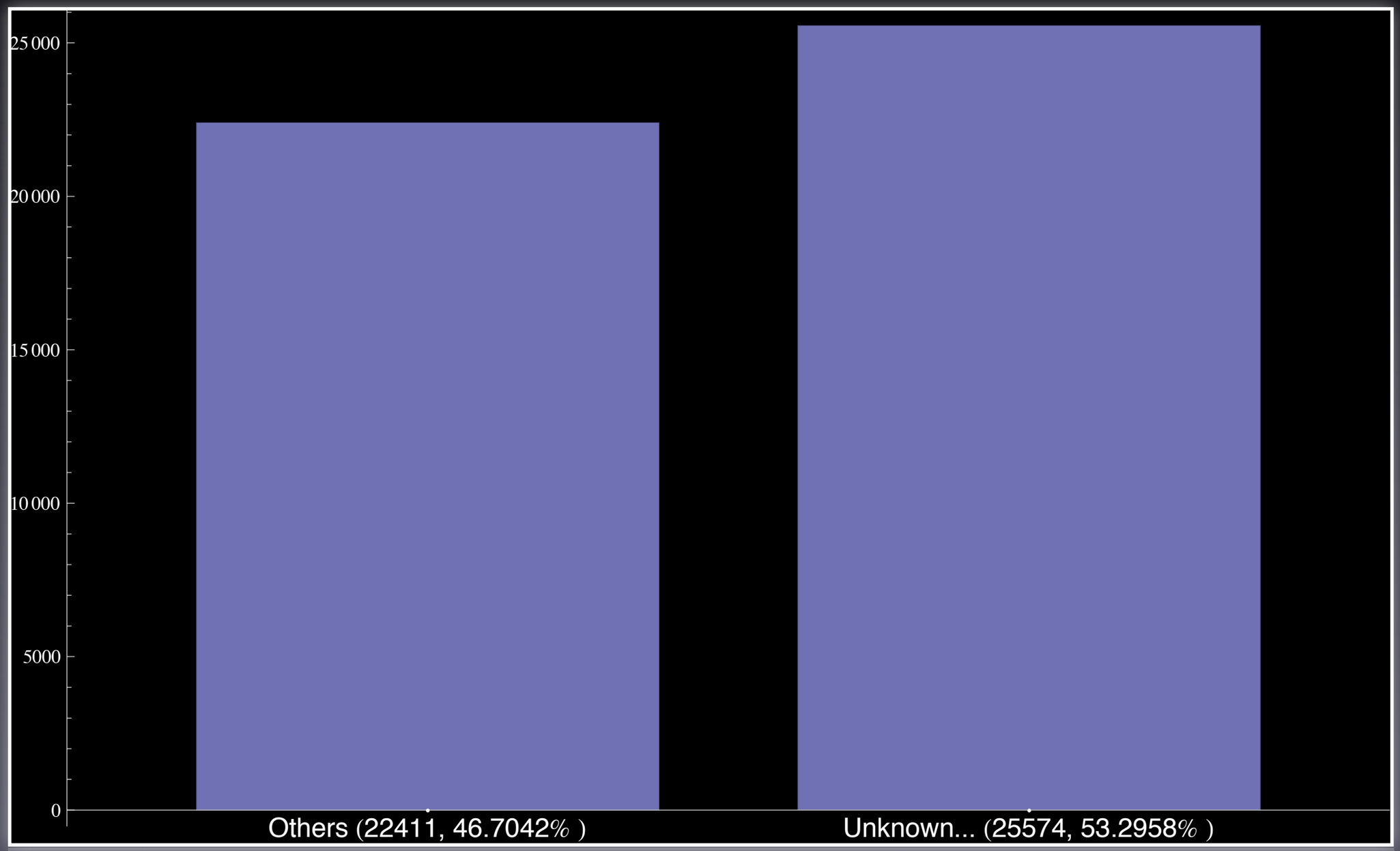
Matching

```
matches = sig.match(pe, ep_only = True)

   ['Upack 0.24 - 0.27 beta / 0.28 alpha -> Dwing']
```

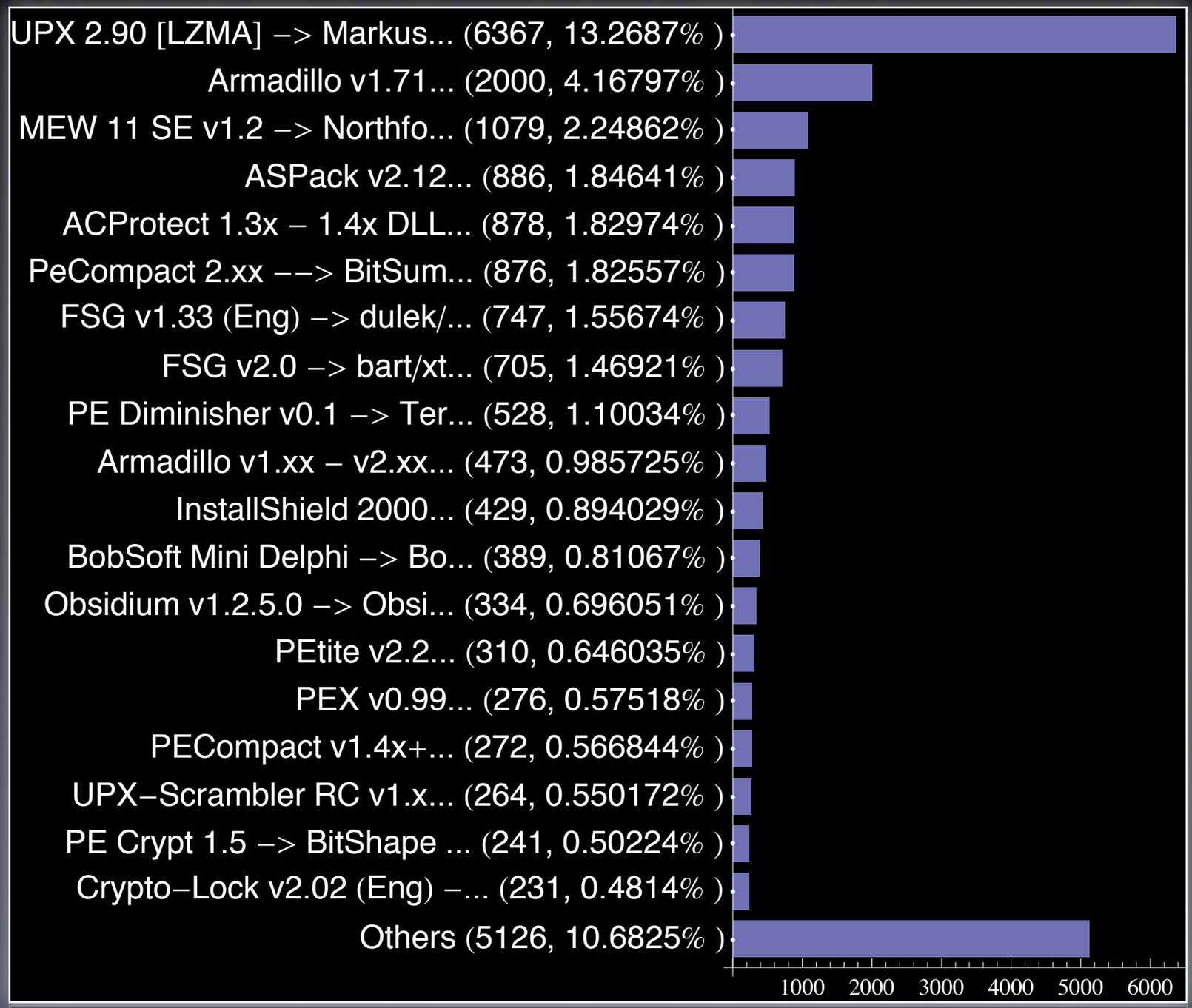
```
matches = sig.match_all(pe, ep_only = True)

   [['Upack v0.1x - v0.2x -> Dwing'],

   ['Upack v0.24 ~ v0.28 Alpha -> Dwing'],

   ['Upack 0.24 - 0.27 beta / 0.28 alpha -> Dwing']]
```

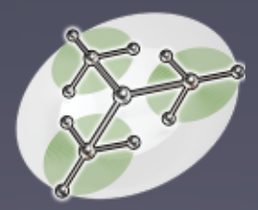


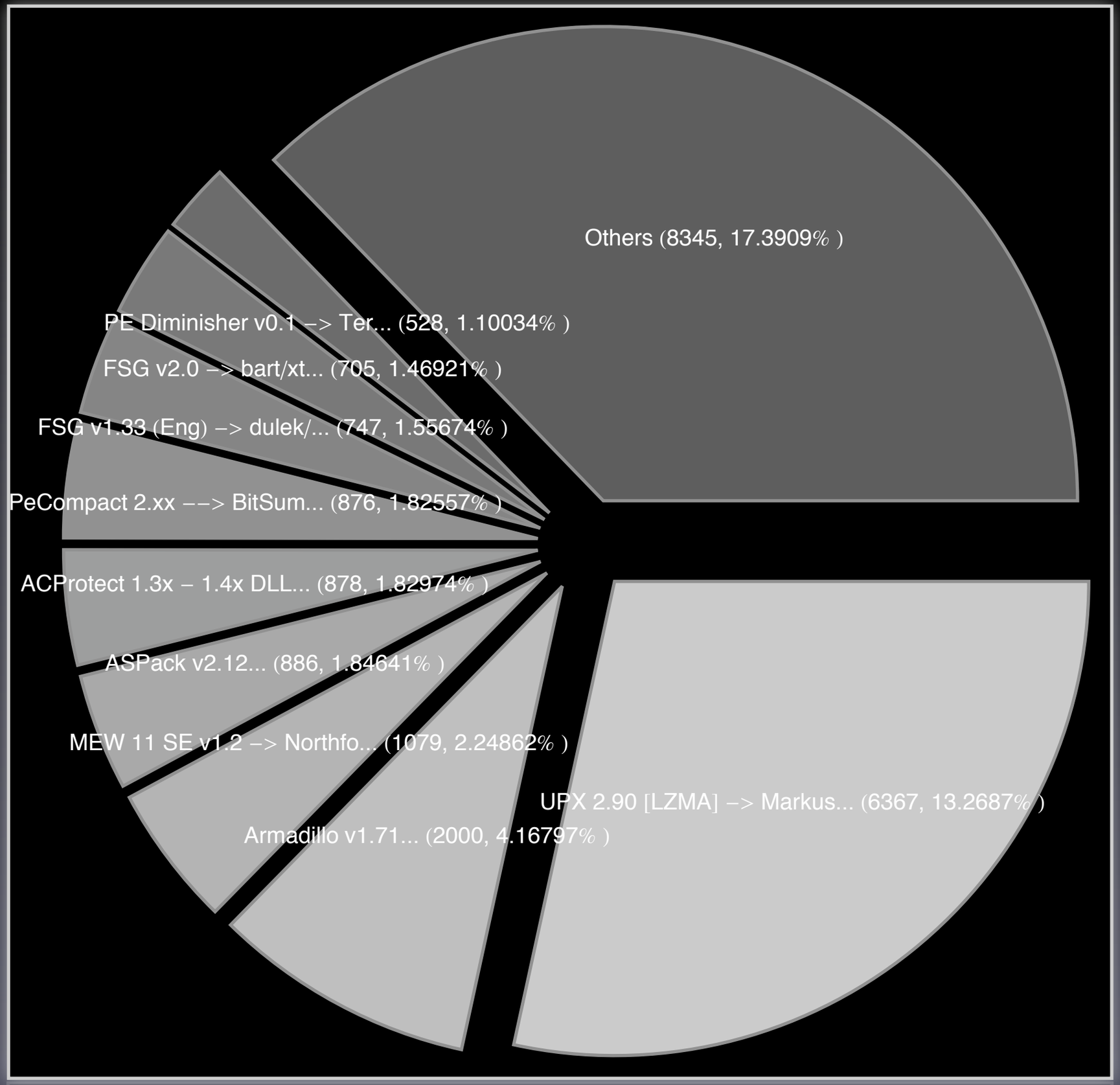
Detection rates



Packer break-down

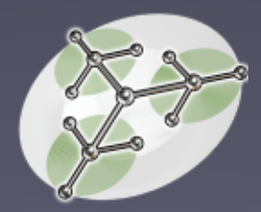





Others (8345, 17.3909%)
PE Diminisher v0.1 -> Ter... (528, 1.10034%)
FSG v2.0 -> bart/xt... (705, 1.46921%)
FSG v1.33 (Eng) -> dulek/... (747, 1.55674%)
PeCompact 2.xx -–> BitSum... (876, 1.82557%)
ACProtect 1.3x - 1.4x DLL... (878, 1.82974%)
ASPack v2.12... (886, 1.84641%)
MEW 11 SE v1.2 -> Northfo... (1079, 2.24862%)
UPX 2.90 [LZMA] -> Markus... (6367, 13.2687%)
Armadillo v1.71... (2000, 4.16797%)

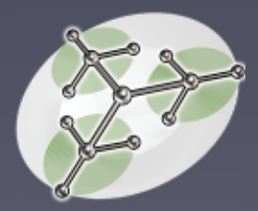
pydasm

http://dkbza.org/pydasm.html



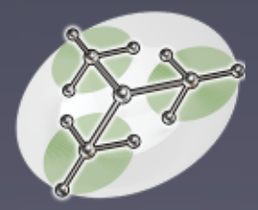
pydasm

- *pydasm* is a multi-platform Python module wrapping jt's *libdasm*
- *pydasm* together with *pefile* provide with a good tool set to develop mini-IDA wannabes
- Has been recently used, in conjunction with *pefile*, by Cody Pierce from TippingPoint in pyemu, his x86 emulator



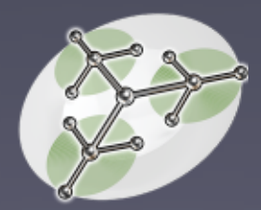
Disassembling

```
>>> import pydasm

>>> i = pydasm.get_instruction('\x90', pydasm.MODE_32)
>>> pydasm.get_instruction_string(
      i, pydasm.FORMAT_INTEL, 0)

['nop ']

>>> i = pydasm.get_instruction(
       '\x8B\x04\xBD\xE8\x90\x00\x01', pydasm.MODE_32)
>>> pydasm.get_instruction_string(
       i, pydasm.FORMAT_INTEL, 0)

['mov eax,[edi*4+0x10090e8]']
```



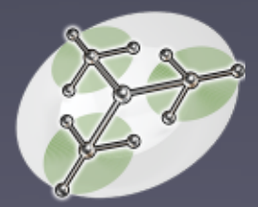
pefile+pydasm

```
>>> ep = pe.OPTIONAL_HEADER.AddressOfEntryPoint
>>> ep_ava = ep+pe.OPTIONAL_HEADER.ImageBase
>>> data = pe.get_memory_mapped_image()[ep:ep+100]
>>> offset = 0
>>> while offset < len(data):
...     i = pydasm.get_instruction(
              data[offset:], pydasm.MODE_32)
...     print pydasm.get_instruction_string(
              i, pydasm.FORMAT_INTEL, ep_ava+offset)
...     offset += i.length
```



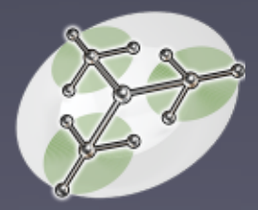
pefile+pydasm

```
push byte 0x70
push dword 0x1001888
call 0x1006ca8
xor ebx,ebx
push ebx
mov edi,[0x100114c]
call edi
cmp word [eax],0x5a4d
jnz 0x1006b1d
mov ecx,[eax+0x3c]
add ecx,eax
cmp dword [ecx],0x4550
jnz 0x1006b1d
movzx eax,[ecx+0x18
```



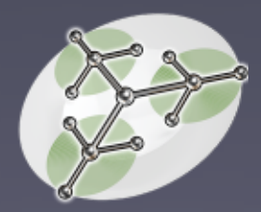
ida2sql

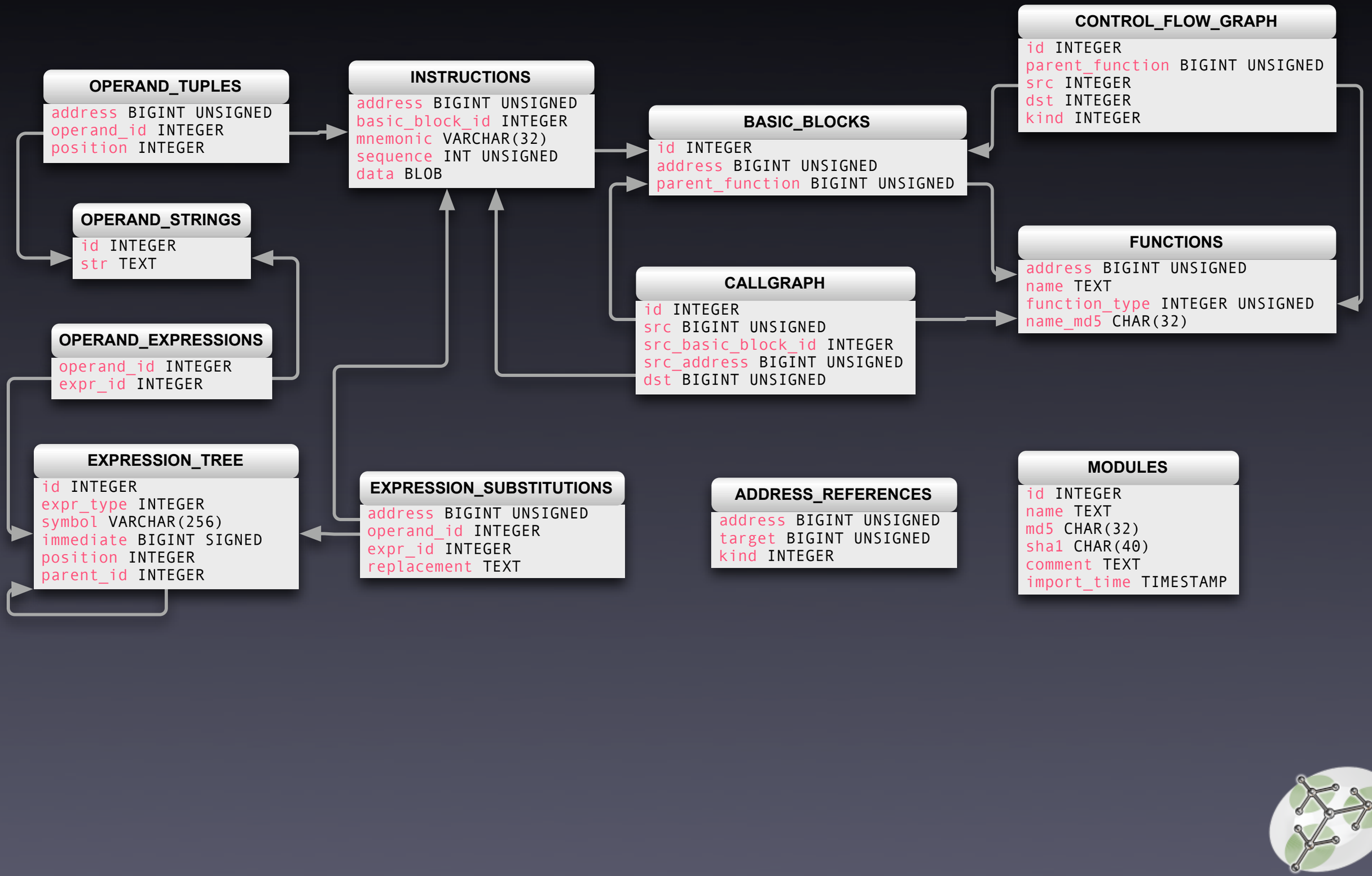
http://dkbza.org/ida2sql.html



ida2sql

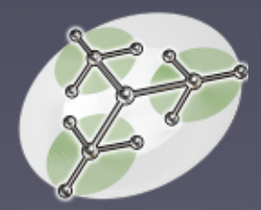
- Part of BinNavi
- Open source
- Export IDA's IDB database to SQL
- Store all your disassemblies in a single repository
- Perform advanced data-mining
- Paimei will use a nearly identical schema

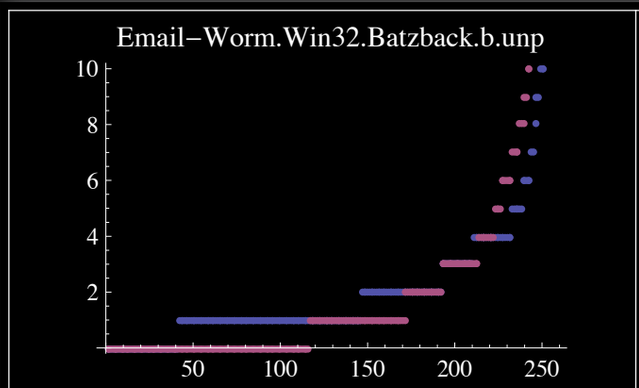


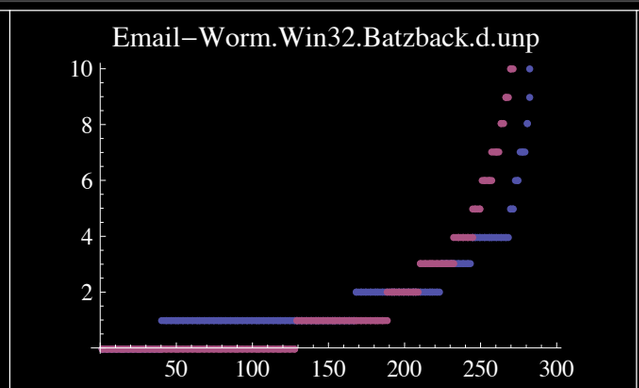

OPERAND_TUPLES
address BIGINT UNSIGNED
operand_id INTEGER
position INTEGER
INSTRUCTIONS
address BIGINT UNSIGNED
basic_block_id INTEGER
mnemonic VARCHAR(32)
sequence INT UNSIGNED
data BLOB
CONTROL_FLOW_GRAPH
id INTEGER
parent_function BIGINT UNSIGNED
src INTEGER
dst INTEGER
kind INTEGER
BASIC_BLOCKS
id INTEGER
address BIGINT UNSIGNED
parent_function BIGINT UNSIGNED
OPERAND_STRINGS
id INTEGER
str TEXT
FUNCTIONS
address BIGINT UNSIGNED
name TEXT
function_type INTEGER UNSIGNED
name_md5 CHAR(32)
CALLGRAPH
id INTEGER
src BIGINT UNSIGNED
src_basic_block_id INTEGER
src_address BIGINT UNSIGNED
dst BIGINT UNSIGNED
OPERAND_EXPRESSIONS
operand_id INTEGER
expr_id INTEGER
EXPRESSION_TREE
id INTEGER
expr_type INTEGER
symbol VARCHAR(256)
immediate BIGINT SIGNED
position INTEGER
parent_id INTEGER
EXPRESSION_SUBSTITUTIONS
address BIGINT UNSIGNED
operand_id INTEGER
expr_id INTEGER
replacement TEXT
ADDRESS_REFERENCES
address BIGINT UNSIGNED
target BIGINT UNSIGNED
kind INTEGER
MODULES
id INTEGER
name TEXT
md5 CHAR(32)
sha1 CHAR(40)
comment TEXT
import_time TIMESTAMP

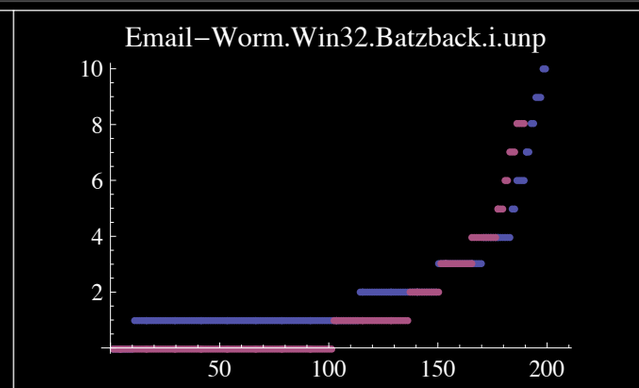
Indegree / Outdegree Plots

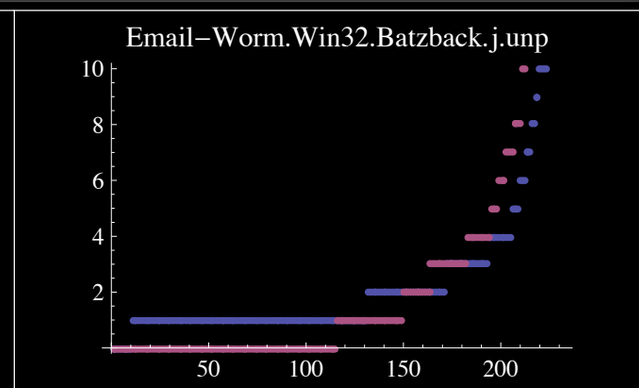
```
SELECT
( SELECT name
  FROM functions_N
  WHERE address=src),
(SELECT name
  FROM functions_N
  WHERE address=dst)
FROM callgraph_N
```

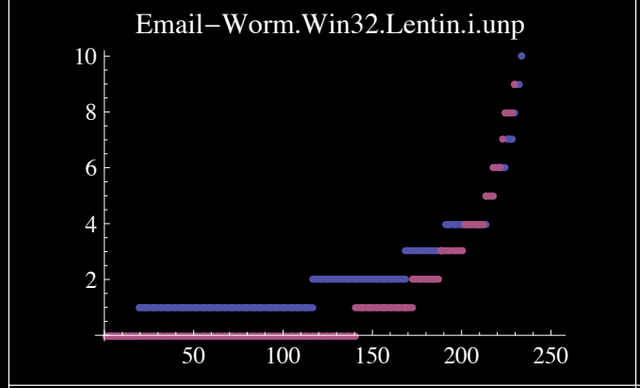


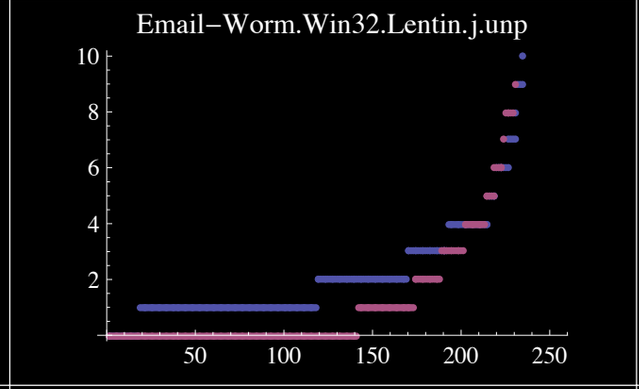

Email−Worm.Win32.Batzback.b.unp
10
8
6
4
2
50
100
150
200
250

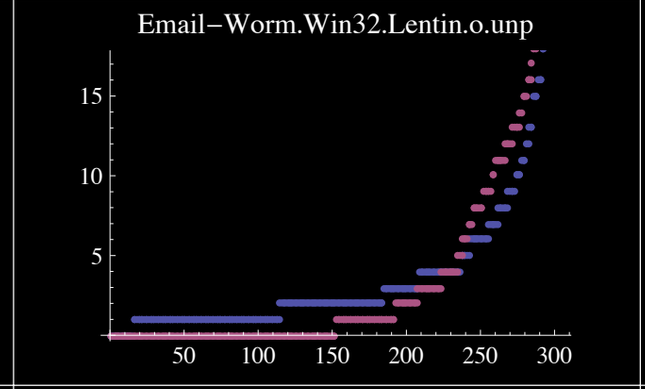

Email−Worm.Win32.Batzback.d.unp
10
8
6
4
2
50
100
150
200
250
300

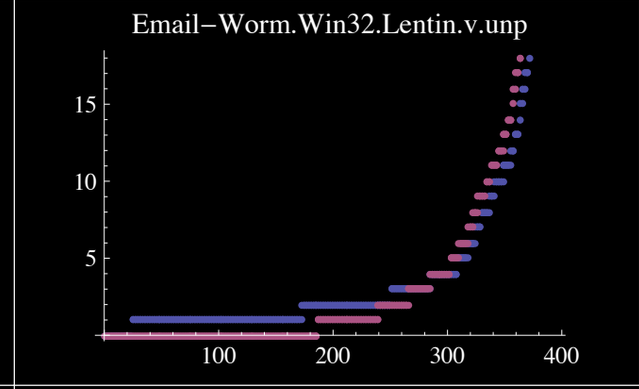

Email−Worm.Win32.Batzback.i.unp
10
8
6
4
2
50
100
150
200

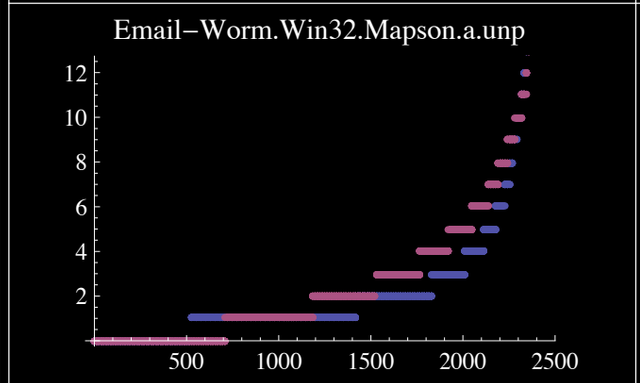

Email−Worm.Win32.Batzback.j.unp
10
8
6
4
2
50
100
150
200

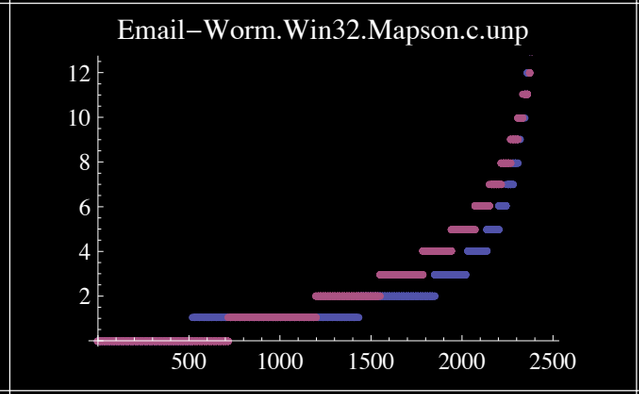

Email−Worm.Win32.Lentin.i.unp
10
8
6
4
2
50
100
150
200
250

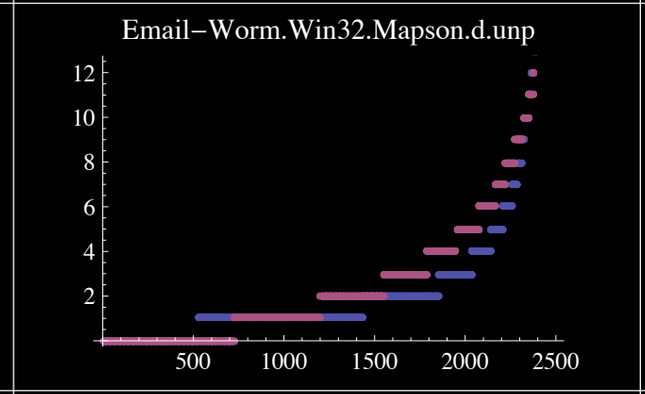

Email−Worm.Win32.Lentin.j.unp
10
8
6
4
2
50
100
150
200
250

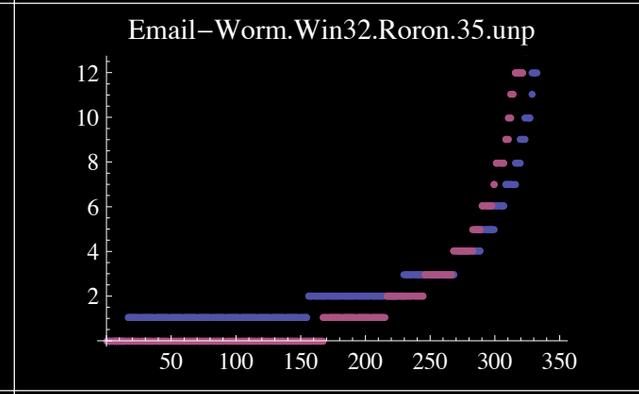

Email−Worm.Win32.Lentin.o.unp
15
10
5
50
100
150
200
250
300

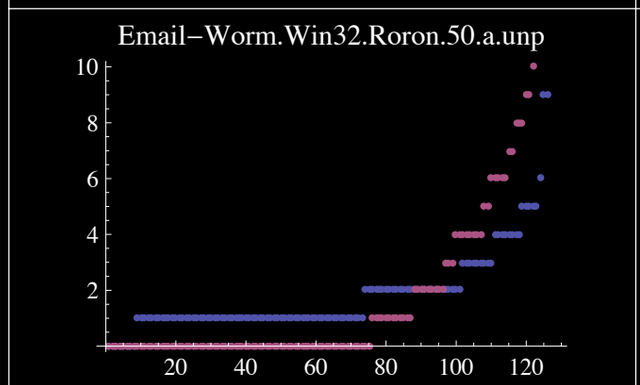

Email−Worm.Win32.Lentin.v.unp
15
10
5
100
200
300
400

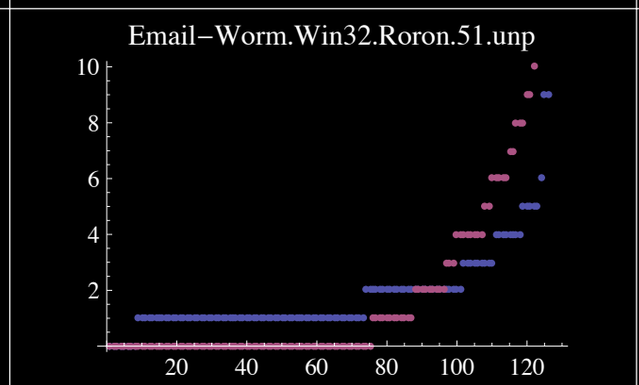

Email−Worm.Win32.Mapson.a.unp
12
10
8
6
4
2
500
1000
1500
2000
2500

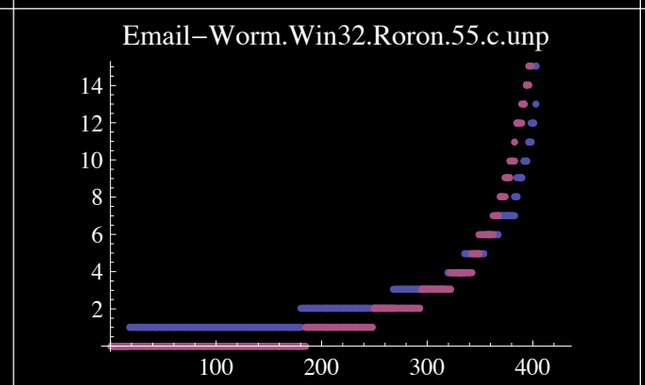

Email−Worm.Win32.Mapson.c.unp
12
10
8
6
4
2
500
1000
1500
2000
2500

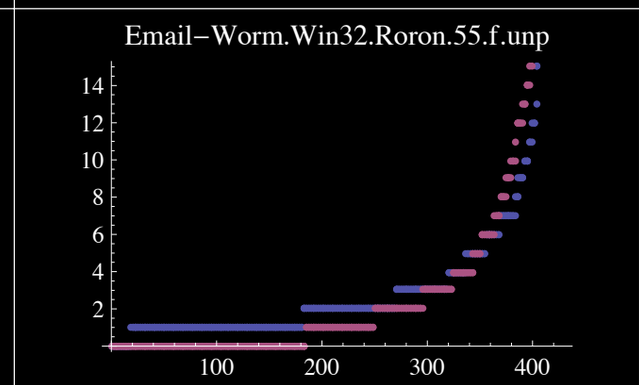

Email−Worm.Win32.Mapson.d.unp
12
10
8
6
4
2
500
1000
1500
2000
2500


Email−Worm.Win32.Roron.35.unp
12
10
8
6
4
2
50
100
150
200
250
300
350


Email−Worm.Win32.Roron.50.a.unp
10
8
6
4
2
20
40
60
80
100
120

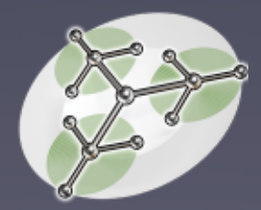

Email−Worm.Win32.Roron.51.unp
10
8
6
4
2
20
40
60
80
100
120

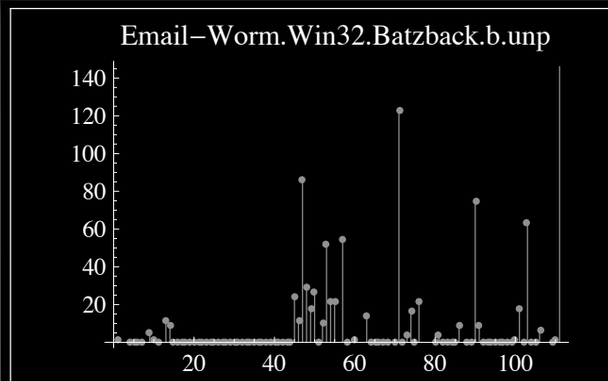

Email−Worm.Win32.Roron.55.c.unp
14
12
10
8
6
4
2
100
200
300
400

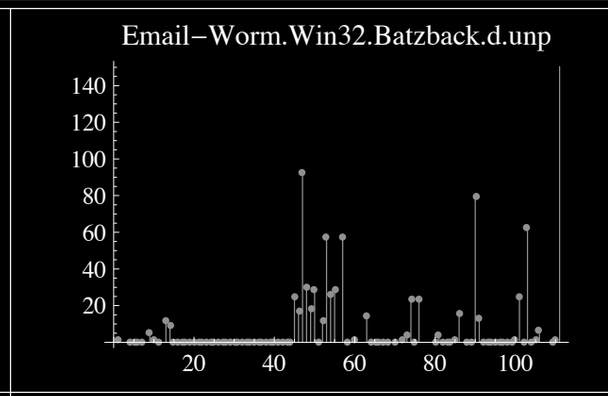

Email−Worm.Win32.Roron.55.f.unp
14
12
10
8
6
4
2
100
200
300
400

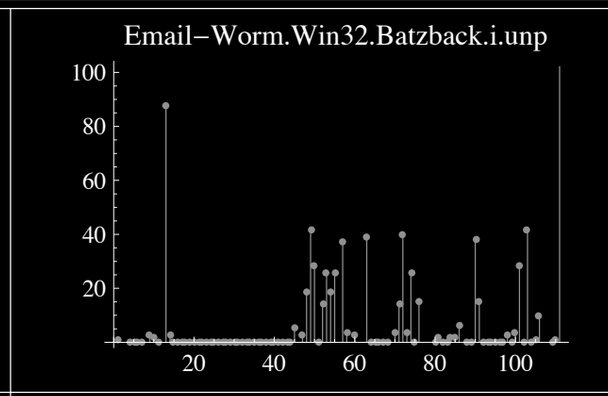
Mnemonic Histograms

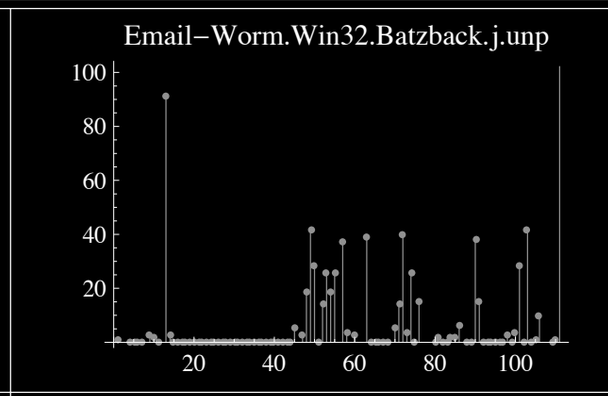
```
SELECT
   mnemonic, COUNT(mnemonic)
FROM
   instructions_N
WHERE
   mnemonic IS NOT NULL GROUP BY
   mnemonic
```

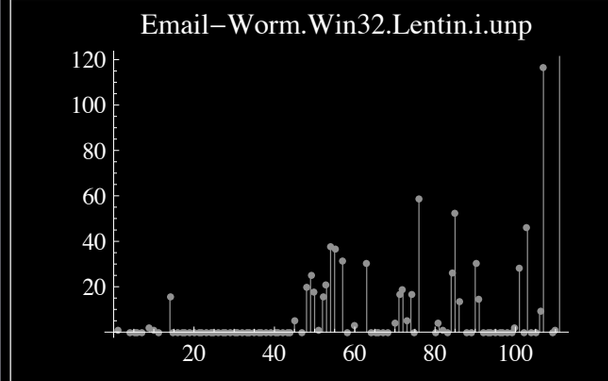


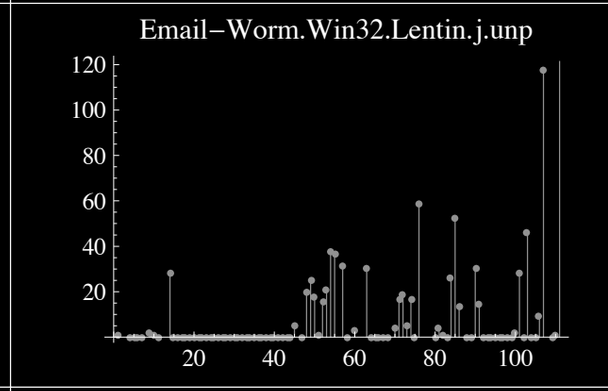

Email−Worm.Win32.Batzback.b.unp

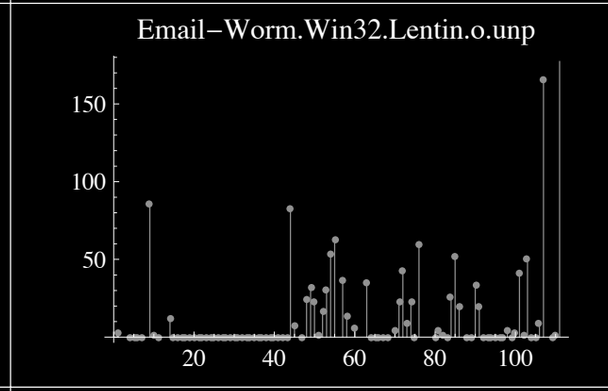

Email−Worm.Win32.Batzback.d.unp

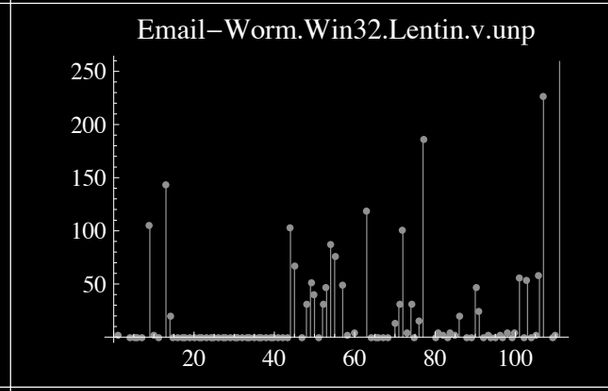

Email−Worm.Win32.Batzback.i.unp

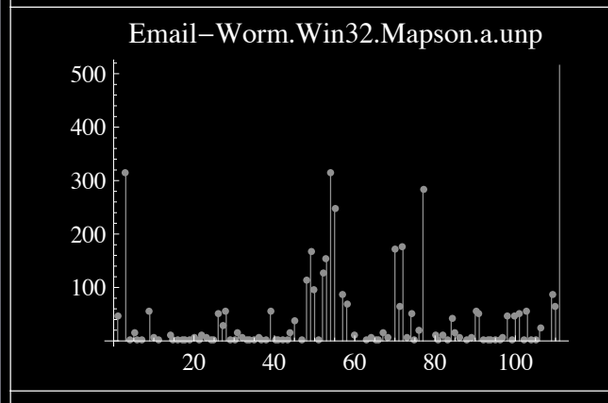

Email−Worm.Win32.Batzback.j.unp

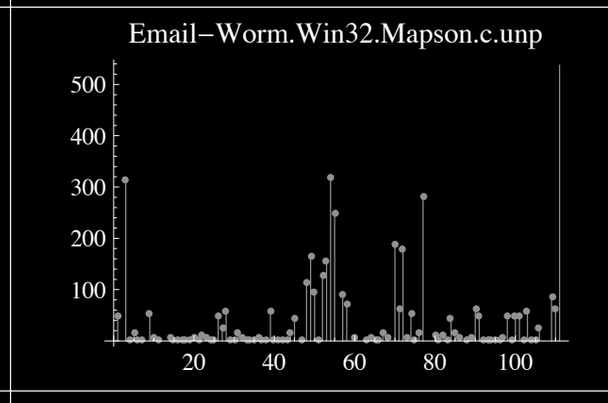

Email−Worm.Win32.Lentin.i.unp

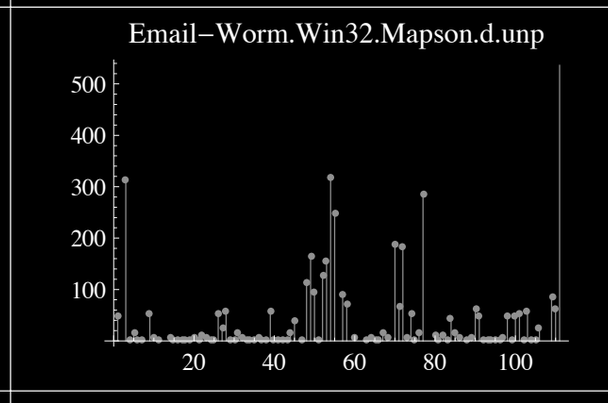

Email−Worm.Win32.Lentin.j.unp

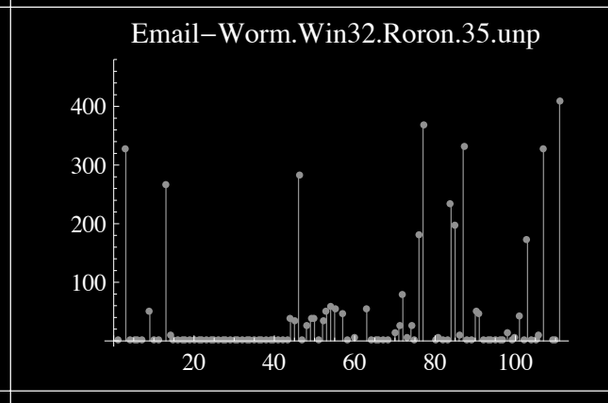

Email−Worm.Win32.Lentin.o.unp

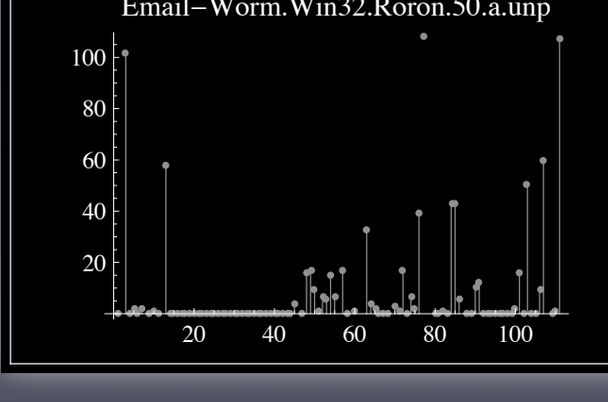

Email−Worm.Win32.Lentin.v.unp

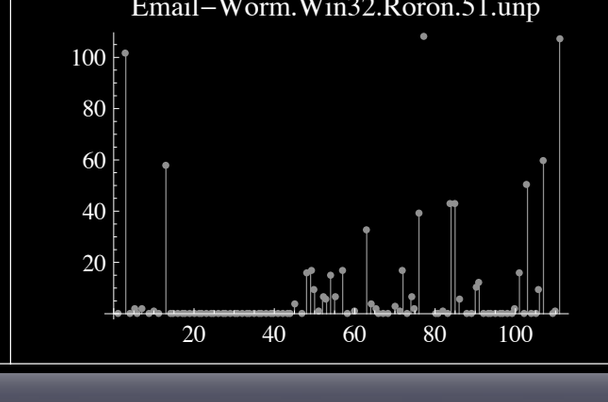

Email−Worm.Win32.Mapson.a.unp

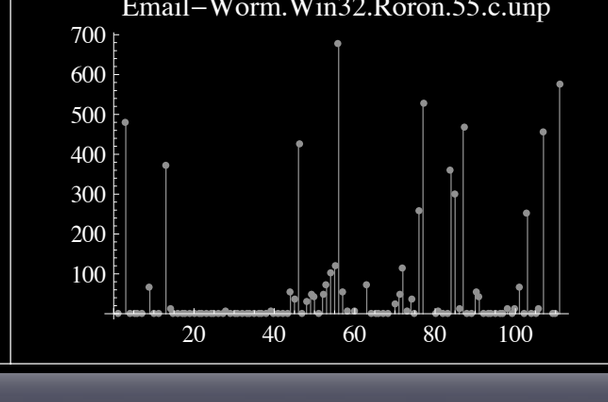

Email−Worm.Win32.Mapson.c.unp

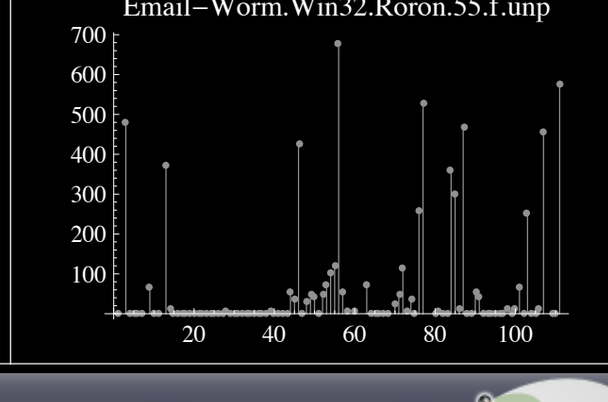

Email−Worm.Win32.Mapson.d.unp


Email−Worm.Win32.Roron.35.unp

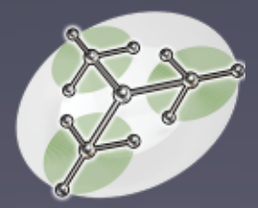

Email−Worm.Win32.Roron.50.a.unp


Email−Worm.Win32.Roron.51.unp

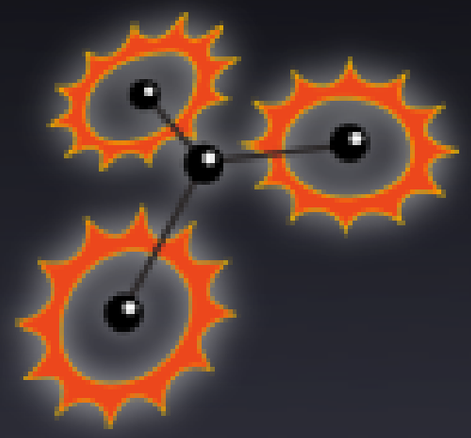

Email−Worm.Win32.Roron.55.c.unp


Email−Worm.Win32.Roron.55.f.unp

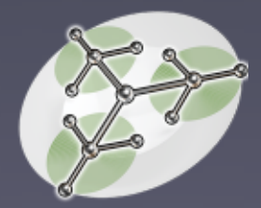
Structure-based Clustering



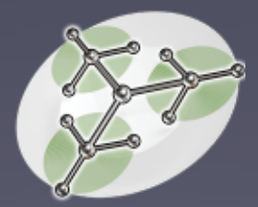

Email-Worm .Win32 Roron.35.unp
Email-Worm .Win32 Roron.41.a.unp
Email-Worm .Win32 Roron.40.b.unp
Email-Worm .Win32 Roron.41.c.unp
Email-Worm .Win32 Roron.gen.unp
Email-Worm .Win32 Roron.55.f.unp
Email-Worm .Win32 Roron.55.c.unp
Email-Worm .Win32 Roron.4999.c.unp
Email-Worm .Win32 Roron.497.unp
Email-Worm .Win32 Lentin.v.unp
Email-Worm .Win32 Mydoom.q.unp
Email-Worm .Win32 Mydoom.u.unp
Email-Worm .Win32 Mydoom.m.unp
Email-Worm .Win32 Mydoom.d.unp
Email-Worm .Win32 Mydoom.g.unp
Email-Worm .Win32 Mydoom.e.unp
Email-Worm .Win32 Roron.4997.unp
Email-Worm .Win32 Roron.50.a.unp
Email-Worm .Win32 Roron.51.unp
Email-Worm .Win32 Lentin.j.unp
Email-Worm .Win32 Lentin.i.unp
Email-Worm .Win32 Mydoom.l.unp
Email-Worm .Win32 Lentin.o.unp
Email-Worm .Win32 Batzback.e.unp
Email-Worm .Win32 Batzback.k.unp
Email-Worm .Win32 Batzback.j.unp
Email-Worm .Win32 Mydoom.a.unp
Email-Worm .Win32 Batzback.i.unp
Email-Worm .Win32 Batzback.d.unp
Email-Worm .Win32 Mapson.d.unp
Email-Worm .Win32 Mapson.c.unp
Email-Worm .Win32 Mapson.a.unp
Email-Worm .Win32 Batzback.b.unp

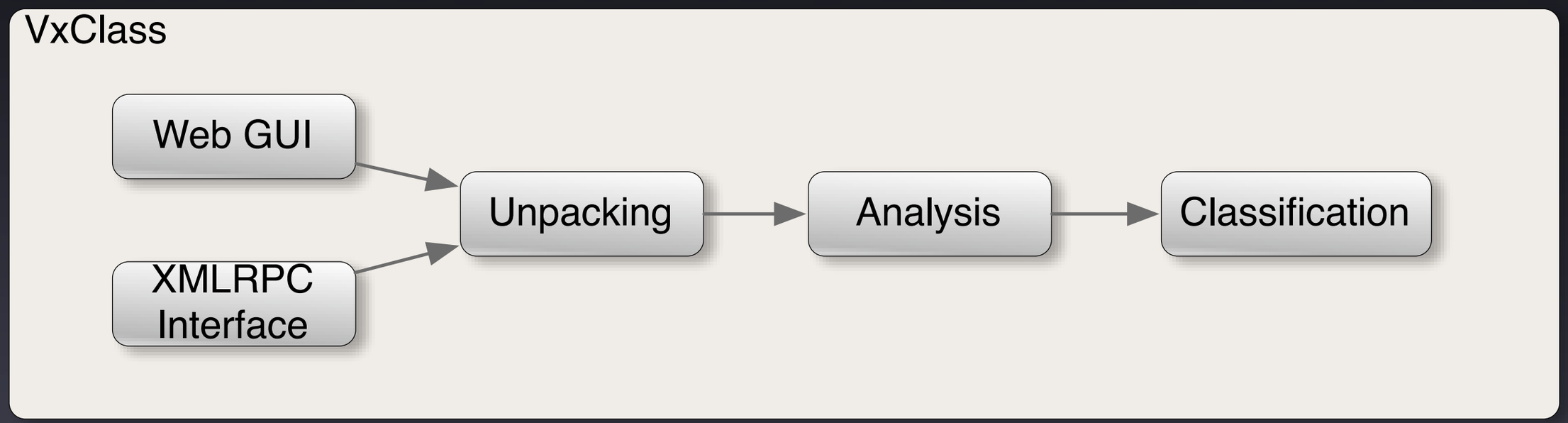


VxClass

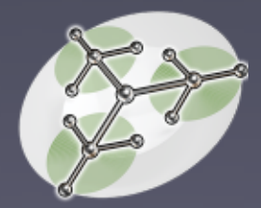


- Automatic classification of malware
- Structural properties are taken into account
- Malware is automatically clustered into families



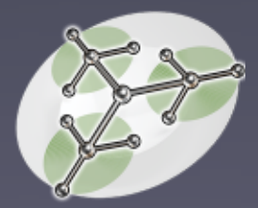

VxClass
Web GUI
XMLRPC Interface
Unpacking
Analysis
Classification

VxClass' Unpacker



Unpacking

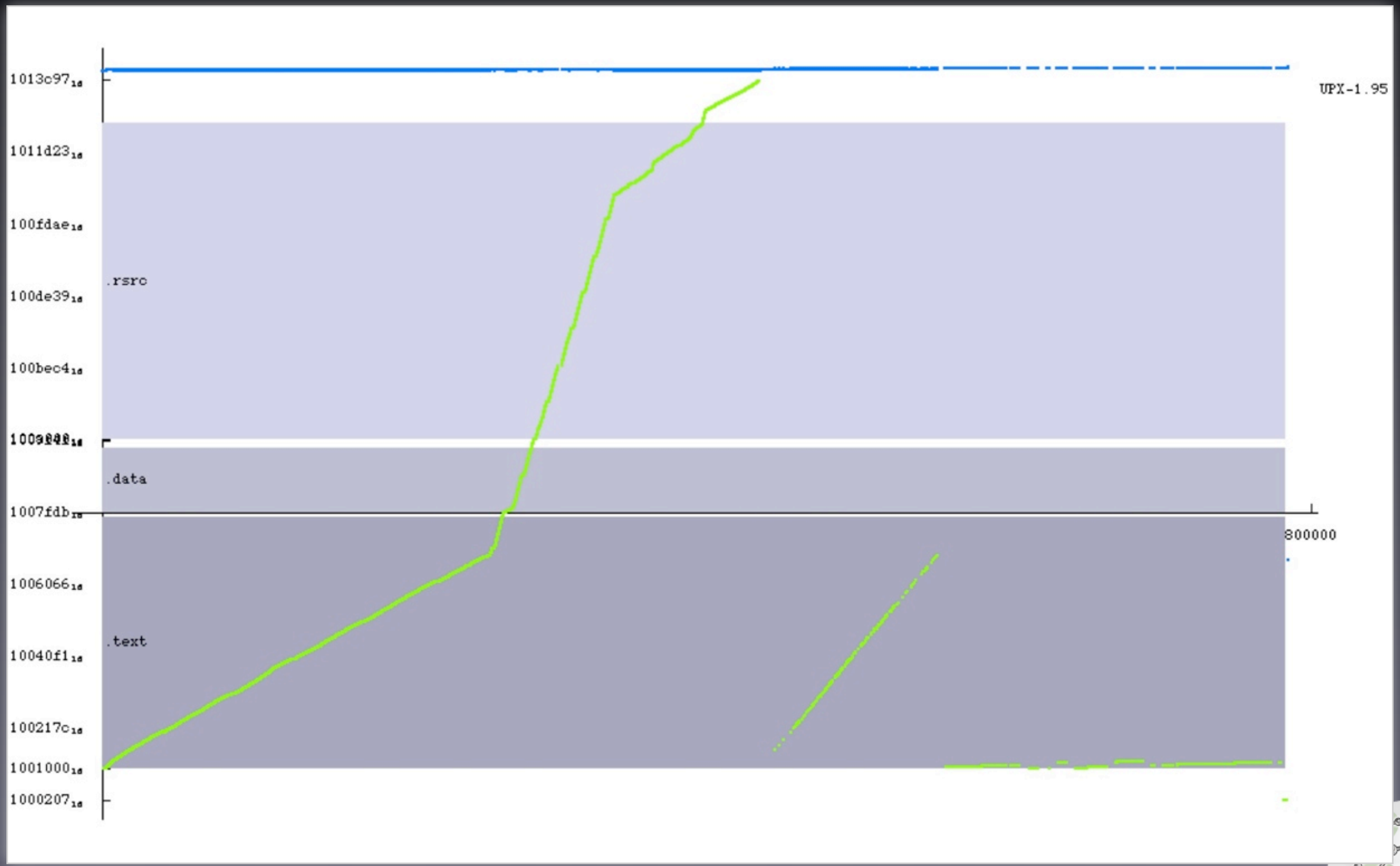
- Bochs (Open Source Intel x86 Emulator)
- Fully Python Instrumented
- Built Heuristics for Unpacking



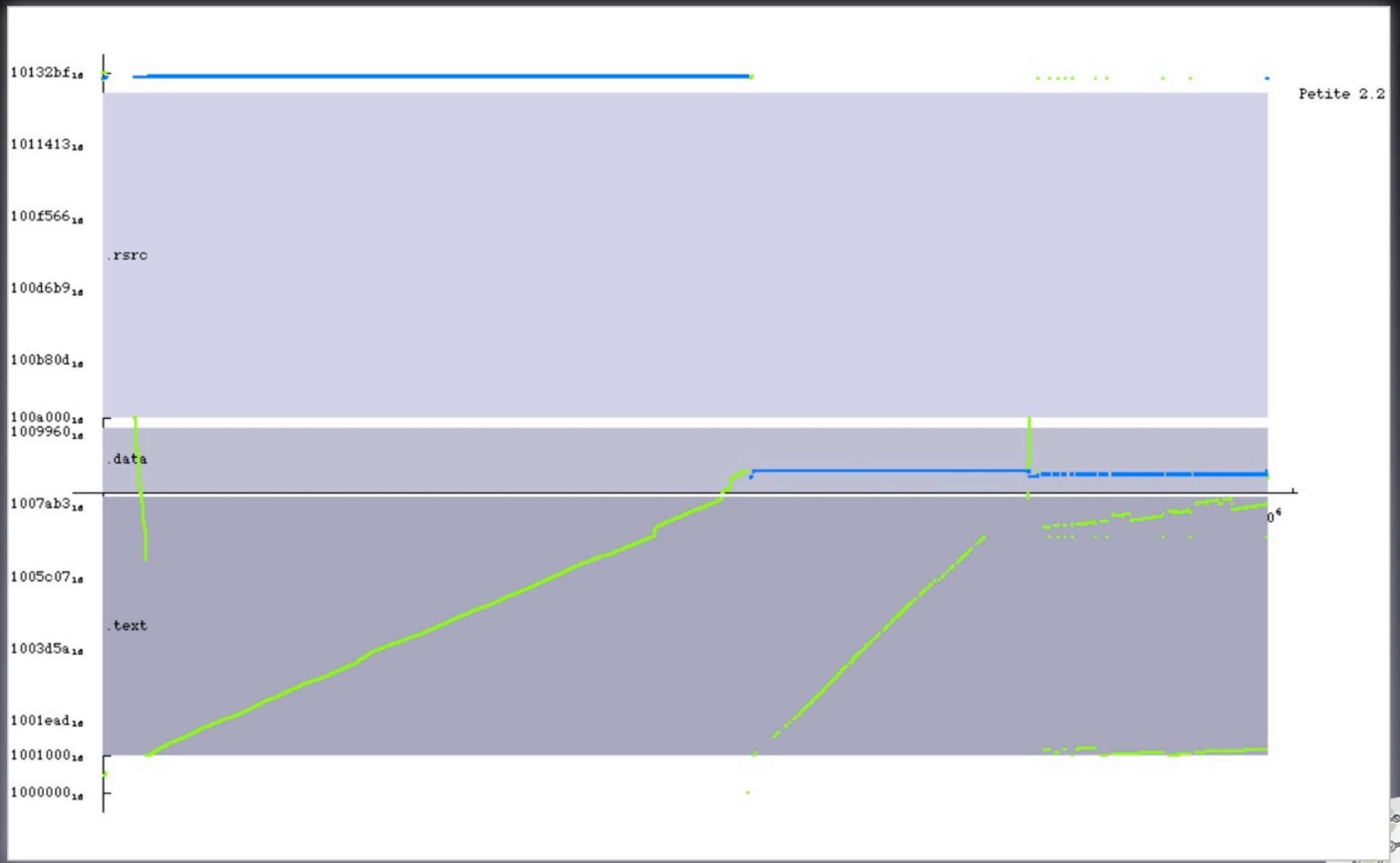
FSG 2.0


FSG v2.0
101f263
101c226
10191e9
10161ac
101316f
1010132
.rsrc
100d0f5
100a0b8
.data
100707b
.text
100403e
1001000
× 10^6

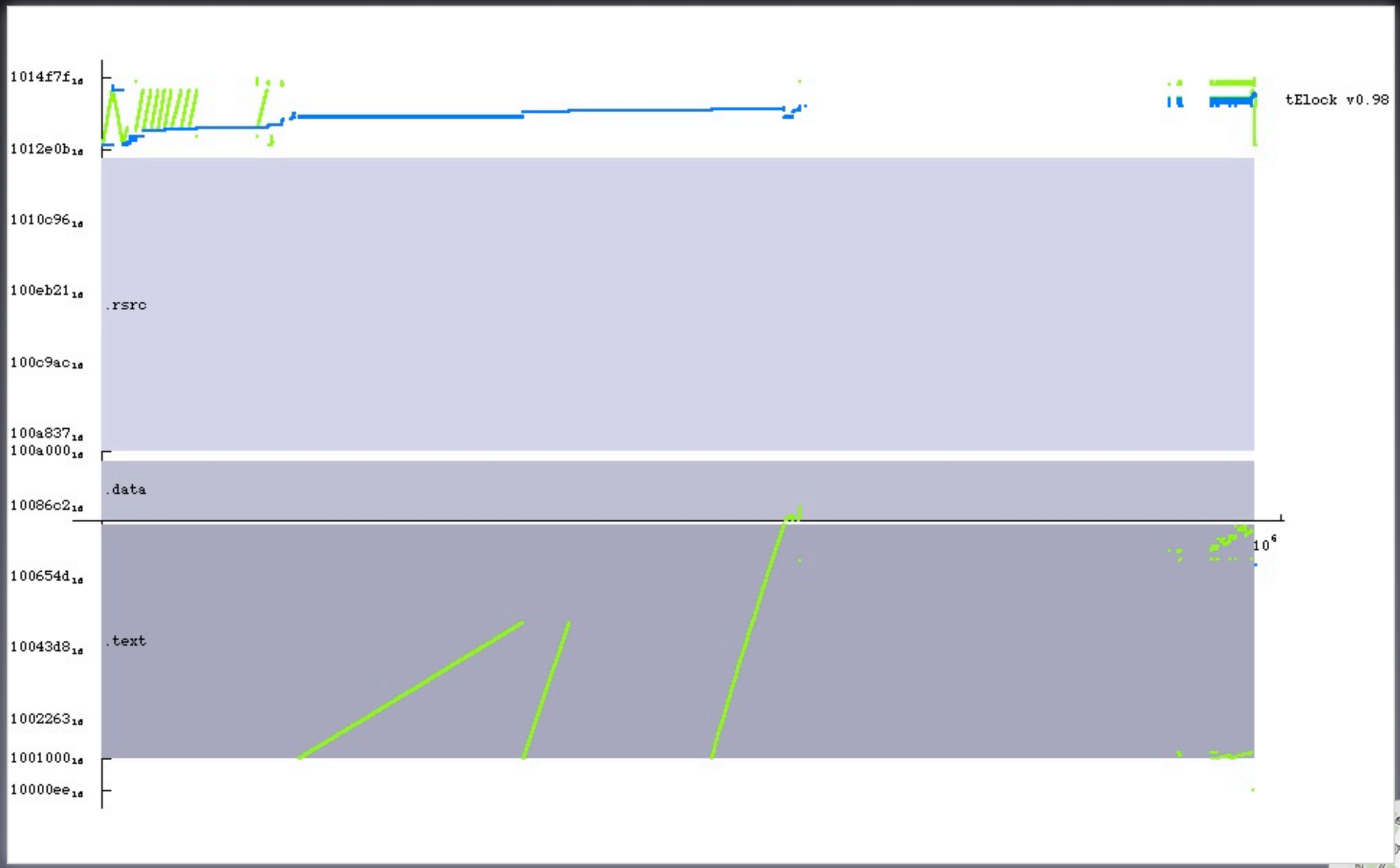
UPX 1.95


1013c97
1011d23
100fdae
100de39
100bec4
1007fdb
1006066
10040f1
100217c
1001000
1000207
.rsrc
.data
.text
UPX-1.95
800000

Petite 2.2

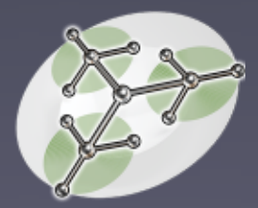

10132bf16
1011413₁₆
100f566₁₆
.rsrc
100d6b9₁₆
100b80d₁₆
100a000₁₆
1009960₁₆
.data
1007ab3₁₆
1005c07₁₆
.text
1003d5a₁₆
1001ead₁₆
1001000₁₆
1000000₁₆
Petite 2.2
0⁶

tElo3k 0.94


1014f7f16
1012e0b16
1010c9616
100eb2116
.rsrc
100c9ac16
100a83716
100a00016
.data
10086c216
100654d16
10043d816
.text
100226316
100100016
10000ee16
tElock v0.98
10^6

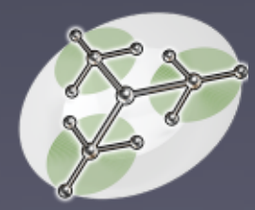
Analysis and Classification

- IDA
- Loads of Python, IDAPython (more than it's healthy)
- The BinDiff Engine



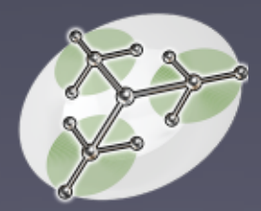
Interfacing

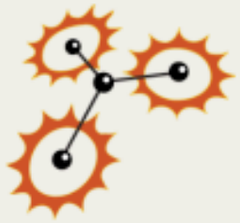
- XMLRPC interface
  - Send file for unpacking
  - Send file for classification
  - Download IDBs
  - Retrieve classification information
- Web interface



XMLRPC

- `classify_executable()`
- `classify_idb()`
- `unpack_executable()`
- `get_classifier_items()`
- `get_classifier_families()`
- `get_classifier_items_by_family()`





Home | Unpacking | Classification

Choose local file to upload:

Browse...

File description:

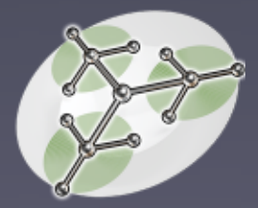
is known malware needs unpacking

Send for classification | Send for unpacking

Manage VxClass

Shutdown VxClass | Restart VxClass

VxClass storage usage: **57% 6%**



Home | Unpacking | Classification

Families | Files | Tree

Classification results

Page: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29

Item Id	Unpacking Time	Classification Time	Status	Filename	File Description	Hash	
145	00:03:20	00:01:02	classifier done	UPack->Net-Worm.Win32.Mytob.bi	Original File: test_54.exe	e72cb162...	a93
146	00:02:59	00:01:02	classifier done	PE_Patch.Morphine->Morphine->UPX->Net-Worm.Win32.Mytob.bi	Original File: test_55.exe	d07462d6...	a93
147	00:03:22	00:01:02	classifier done	UPack->Net-Worm.Win32.Mytob.bi	Original File: test_56.exe	407854dc...	a93
148	00:03:23	00:01:03	classifier done	Net-Worm.Win32.Mytob.bi	Original File: test_57.exe	c009ac33...	a93
149	00:03:18	00:01:02	classifier done	UPack->Net-Worm.Win32.Mytob.bi	Original File: test_58.exe	faed8e81...	a93
150	00:03:16	00:01:02	classifier done	UPack->Net-Worm.Win32.Mytob.bi	Original File: test_59.exe	2fbc010e...	a93
151	00:03:29	00:00:01	classifier done	FSG->Trojan-Dropper.Win32.VB.iv	Original File: test_6.exe	482fd2b9...	482
152	00:03:15	00:01:02	classifier done	UPack->Net-Worm.Win32.Mytob.bi	Original File: test_60.exe	002260a8...	a93
153	00:03:12	00:01:04	classifier done	PE_Patch->MewBundle->MEW->Net-Worm.Win32.Mytob.bi	Original File: test_61.exe	75f72e29...	a93
154	00:03:08	00:01:03	classifier done	PE_Patch.Morphine->Morphine->FSG->Net-Worm.Win32.Mytob.bi	Original File: test_62.exe	839166d1...	a93
155	00:03:42	00:01:02	classifier done	PESpin->Net-Worm.Win32.Mytob.bi	Original File: test_63.exe	d6602c86...	a93
156	00:03:22	00:01:02	classifier done	UPack->Net-Worm.Win32.Mytob.bi	Original File: test_64.exe	8a102438...	a93
157	00:03:33	00:01:03	classifier done	PE_Patch.Morphine->Morphine->UPack->Net-Worm.Win32.Mytob.bi	Original File: test_65.exe	3fc00334...	a93
158	00:02:41	00:01:03	classifier done	PE_Patch.Morphine->Morphine->UPX->Net-Worm.Win32.Mytob.bi	Original File: test_66.exe	cc35e4a8...	a93
159	00:03:37	00:01:03	classifier done	PE_Patch.Morphine->Morphine->UPack->Net-Worm.Win32.Mytob.bi	Original File: test_67.exe	67316180...	a93
160	00:02:37	00:01:03	classifier done	Petite->Net-Worm.Win32.Mytob.bi	Original File: test_68.exe	ba8691e3...	a93

New file name/comment:

Get IDBs | Rename | Comment

Home | Unpacking | Classification

Families | Files | Tree

classification results

Page: 1 2 3 4 5

	Id	Family Name	Family Description	Number of Files in the Family
☐	2	bbf8091b6ca1891667d842da0cd088bc6407077eb13a26248048fa2bc700962a		3
☐	3	a936f96fabf78062c55da994e9a0dcef64494333ddca42730df49b21ba679ab7		57
☐	4	3f9554d1411032007b771876ac9faaacbcfa22e936d69eed6941b66b9b9ef42b		6
☐	5	1ba387a7259a01b1e425d613c9716f54c3261f60cbf96c6b80fb30bec9228334		3
☐	6	ac3df1e77a00fdba8cadfc4a4e73a311c50f5afe4a264ee0b4ba4397933948cd		58
☐	7	dd0a2090d3bc458dff550e89d524188ef6dcb509ea5c62dd84341ebd249ddec3		2
☐	8	14754f53ea131020f945e160c77eb362e6aba5d116d85c84f67deaa0dfe1cfab		1
☐	9	2bc46146a8f1d9982b5c12e639435612bdc3685c53fc33f1317452afb601764c		4
☐	10	5a769f990936c7e0bf856b90ffa340a968f13463cffd5643408df62a49f5ea83		1
☐	11	1967e6a9013aa7a6d11fa4e9cabee49ec5c48b87ffa61cbba2d35c75e20d30cf		9
☐	12	b06840dad794177214c84540a83a32df9cdb76b2343e4533597cdba5cbfa892c		1
☐	13	e43b1dcf435897190f346e54130c03f3b855bbc32556a38c08a6821ecf6a7d15		5
☐	14	4b2a7c74a792b2a42f0838a36e5a6dde32e158e57a99437275d661c32134fd45		1
☐	15	029269655941e1a963916837cd3c6489422b8e75a41c33a87e28bf4af02da7dc		1
☐	16	cb61c18a7318ed6c429067ef34afd0a70d69b8d2703486a9749861e24e50f39b		1
☐	17	1a9b47a08bf5ee52ce24507f322222d73b46095ace1e74e4025a4f92b02ed5ab		2

New family name/comment:

Rename | Comment

Home | Unpacking | Classification

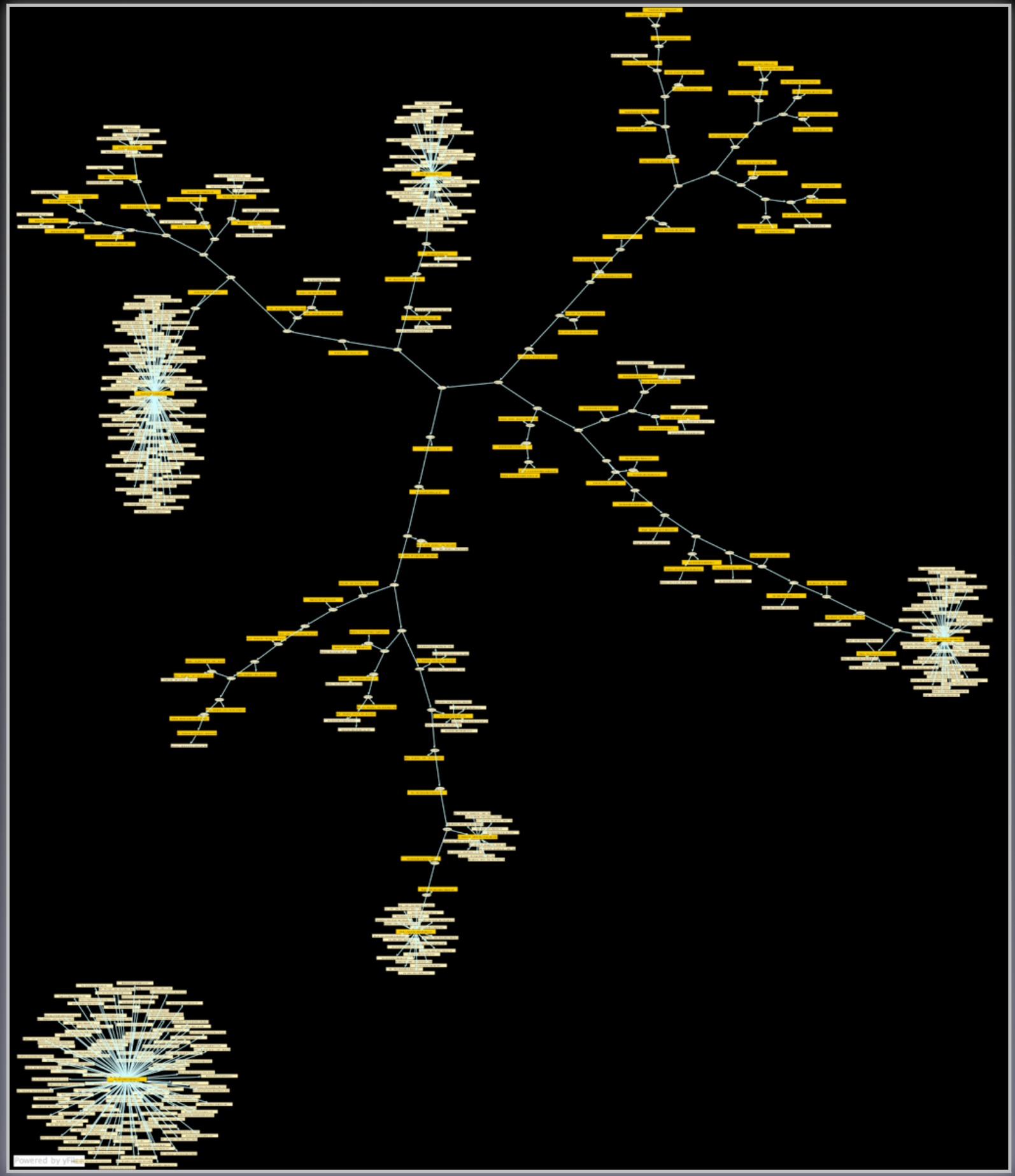
Families | Files | Tree

classification results

Search Regular expression

a902a09035664dbb9652
ac9e22e5f4367323f83f21
b9aa0e09bb2d30b00d03
b8d6fb4c6554601f80ee65
a11ba166b3ee85433f92e
abcda210bb1b00508d
a6c3551eac11da431cd9f0
a7b6685648ea163405
b995c31601ae98fa26d51
a432d727c647b47601f4a
b946bd8c208667a773bf8
a9fe9692a593afe45a3d99
b51809cdfb95b5af814f87
a0c560609aac19cea8bb2f
b4a2aa96f248a1fe5c642c
ae10ad93cfbc4821609ca2
a294e02be5da8db84a8da

0%


Powered by yFiles

Thanks,

Questions?

(*grab a t-shirt!*)