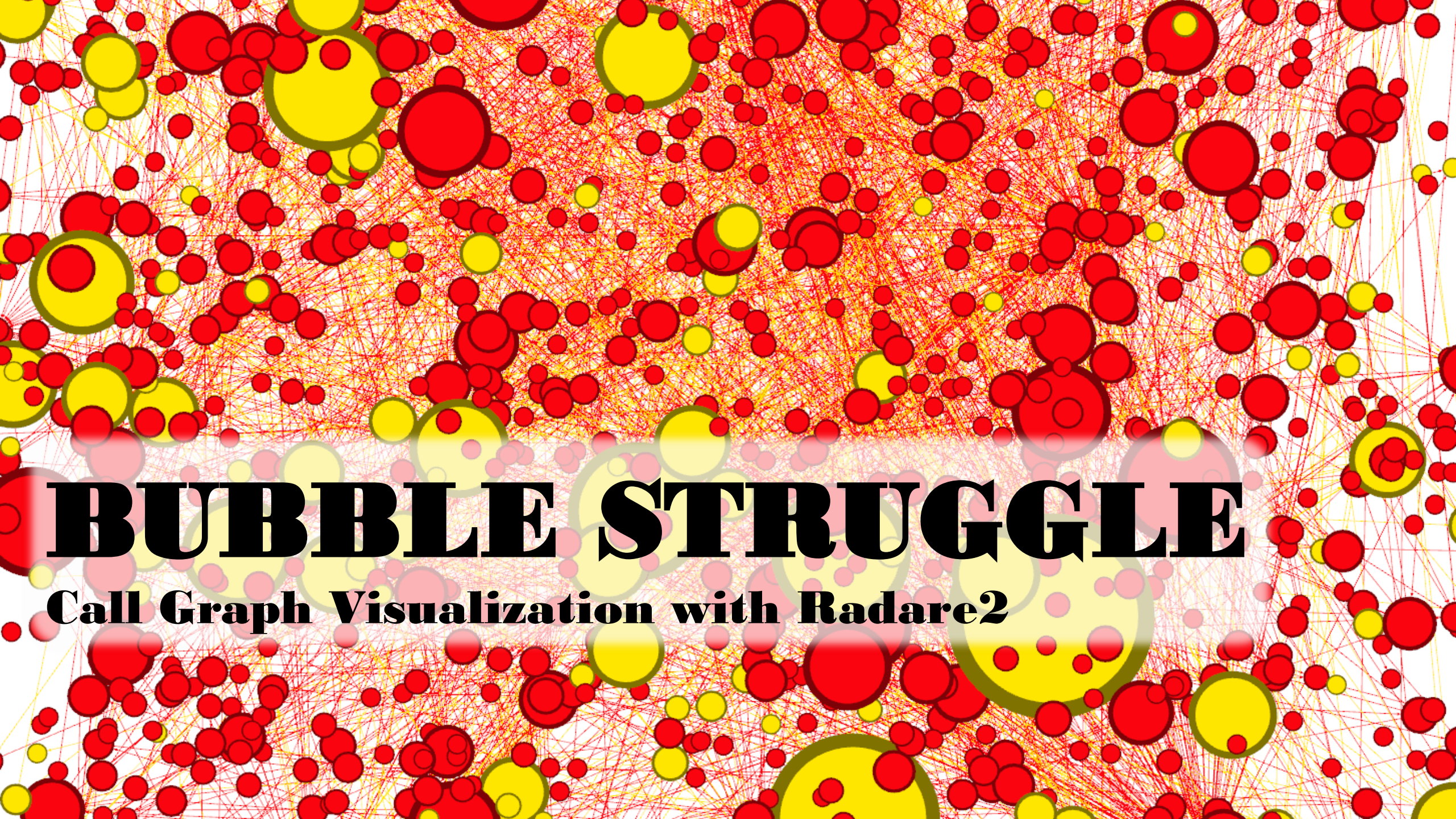
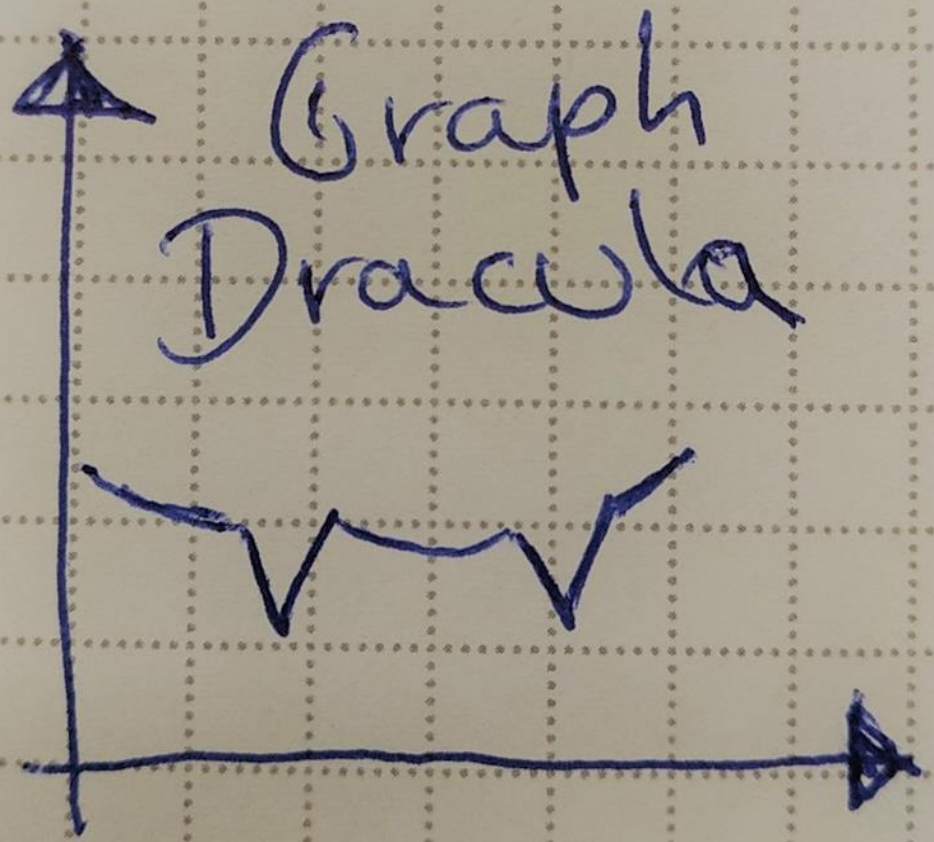

BUBBLE STRUGGLE
Call Graph Visualization with Radare2

Marion Marschalek

marion@0x1338.at

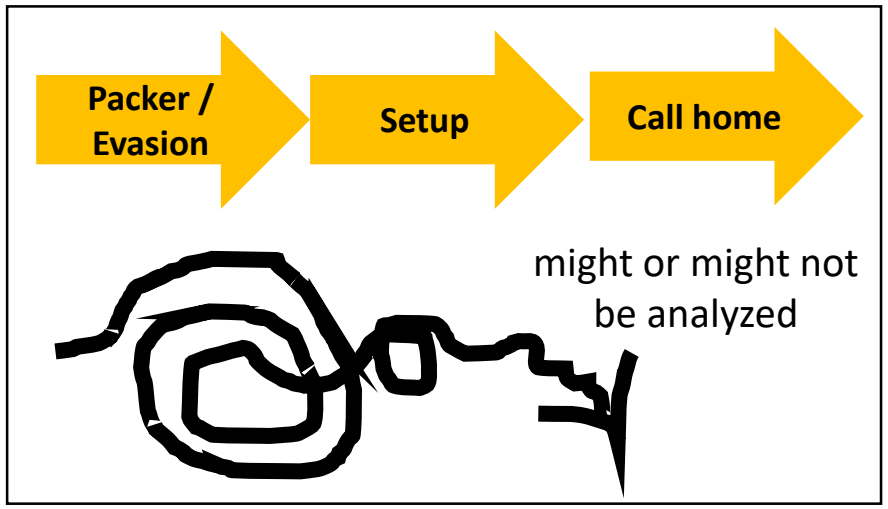
@pinkflawd



Static Analysis is ~~King~~ Princess

What my customer thought the malware does

What my sandbox thought the malware does



What I thought the malware does

What the malware REALLY does

Encrypting files
Keylogging
Screenshots
Screen captures
DDoS
Downloading more malware

r2graphity

https://github.com/pinkflawd/r2graphity

Python3
radare2 & r2pipe
NetworkX
pefile
pydeep
numpy
Neo4j/py2neo

Scalable

Scriptable

GUI-free

Great support

Quick bug fixes

Can analyze entire binaries

Provides

- functions and cross references
- symbols
- strings
- basic PE information

```
R2handle = r2pipe.open(<file>)
R2handle.cmd(<cmd>)
Watch magic
```

r2 command cheat sheet

aaa – analyze the target binary
afr @ [address] – recursively analyze function at [address]
iS – get information about file sections
iij – get import table in JSON format
axtj @@ sym.* - get cross references on found symbols in JSON
axtj @ [address] – get cross references for [address]
pd 300 @ [address] – disassemble 300 instructions at [address]
pd -30 @ [address] – disassemble backwards 30 instructions at [address]
pdf @ [address] – disassemble function at [address], after e.g. aaa command
izzj – get strings out of entire binary in JSON
iz – get strings out of code section
iEj – get exports of a library
?v $FB @ [address] – get function which contains [address]
aflj – get list of functions with supporting information in JSON

Function Detection is Key

Win8 32-bit benign

(Little agreed on method to verify whether TP/FP)

Sample SHA-1	R2 Function Count	OTHER Function Count
051bfe73d395973f5679dba2309f70906de67829	2260	1740
14acdb96c0cf537b20099962b2536bca48775dc4	48	42
18befbfc692df3d6b2205a90a70e64e1787bd11b	35	32
36a13e7f9bb93218695b391b387407b9c197c1ba	394	380
36e870c189f1a5006ac7d989cdfc160ec07f3b5a	1011	805
4d0c5033fadf53bdd0ff330f0ec146df5f7104cf	169	233
64428d1a4aad359c78155d1bcf96bad98162dbb0	42	36
911d81d9c7df4d63c33f51f758ba26489808c4e2	813	788
927592cfea4497a27fe95af9978ffb9e93cb85af	343	317
98a9ac93fe31f38f47f38db78bf12fa0c6214f9a	775	467
9c3e75f34fec80660a754aff4d213810a2753d66	34	28
9cff7f11e977200a9326c22d17463262de8f0a2b	392	245
a29930dd7dc2ba835bdf648ba20a273939c7815d	51	45
a44af16487babd1f625964ec53ce6bd5d9672a22	1916	1373
b1b9e83f5adf8bf22ce9f4943775a9d8f52a87e5	593	434
c0ae1f729dc0d7fa5132200a4f54cb26a2af70e1	1964	1256
c632ae4d41821da3f16d8678fb29a880c2035a4a	223	158
e6429de6fc6d117e203455be9a8d6f475428b658	232	222

Function Detection is Key

32-bit malicious

(Little agreed on method to verify whether TP/FP)

Sample SHA-1	R2 Function Count	Other Function Count
0e8ca304d7907f2d01a3cad2ac8334cde4e53dd8	10	8
151e04886df09fc5c85a0b92ab22cad8264ae9a1	143	137
161f950df5a75b557f2c200d5dc2498937990475	31	16
189c1f5a8a2efaf6477bc3208bb72971eca081d3	16	13
2453dab3b42af9f25e38e22fcd39ff68f35755c3	45	33
25a08e26773ebd5bcbf7d51586d5dc863acc0204	2	1
2d8550af89ad7a964566e090036c0cd75e7cdddc	239	217
2e731d396571254744dc3643c9c4970d49428c38	71	63
315fbb2fb4dcb103839d7a307a7c39a47b9bfe27	127	120
32b1b98177cfc94d515b76e24b09003e9a241c2b	30	26
430f578d2ec7e4d781067340ebf90a9ee3f1f4da	507	497
441d7b8362480e872ae9e0ee784fbd7dd41f18c9	211	195
49352a95766a39aa537a9c4dc8119cc02f9975d3	34	38
4f843e2c8270f594fa016af6bf12de36cfb83232	62	43
58d9b1c60a297d71ccd0c433e85e2cec80f0580a	63	27
5966f710ccf432427c5c333bae63431dd22127c5	2	1
5a9a634a2b6b8516b43da27a8a6003d161d33424	127	120
62bc57417a42d7199c909c8c81616d5767a0851d	325	291
63b65772bdabb67667d41dca8164117bd7c056e5	118	112

Function call graphs

Function cross references within code section

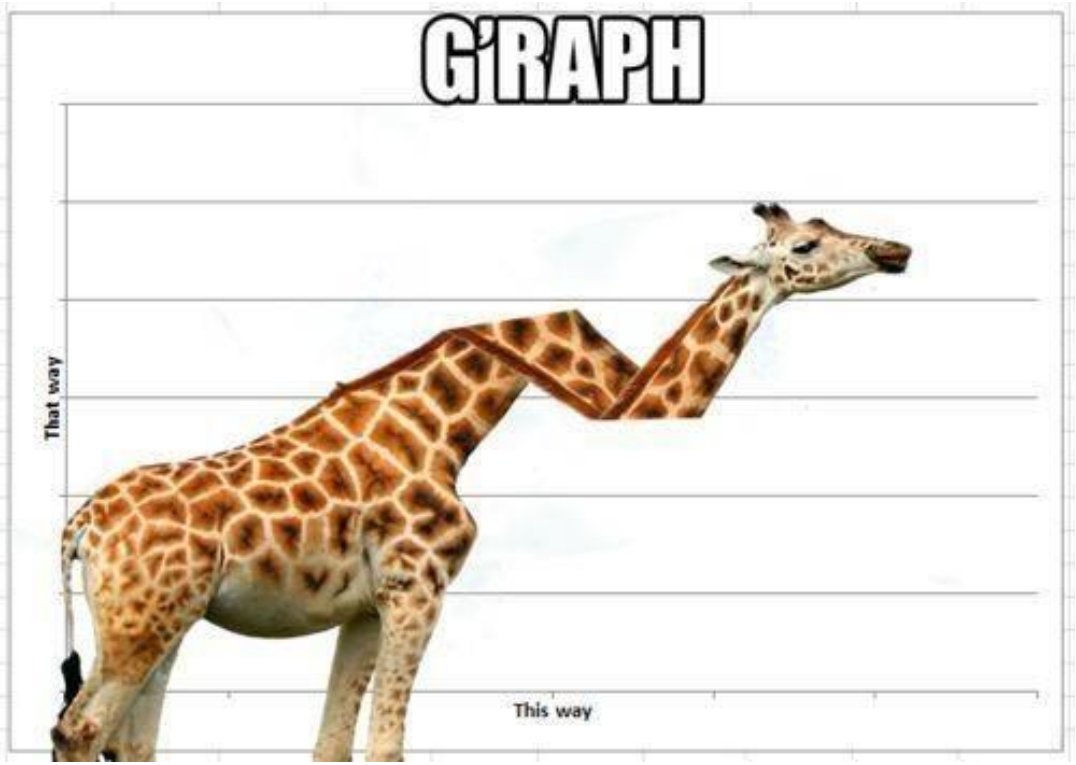
References to function offsets

Outside executable section(s)

Nodes: functions

=> Offset, size, calling convention

Edges: calls, indirect calls



r2graphity

Strings

String parsing

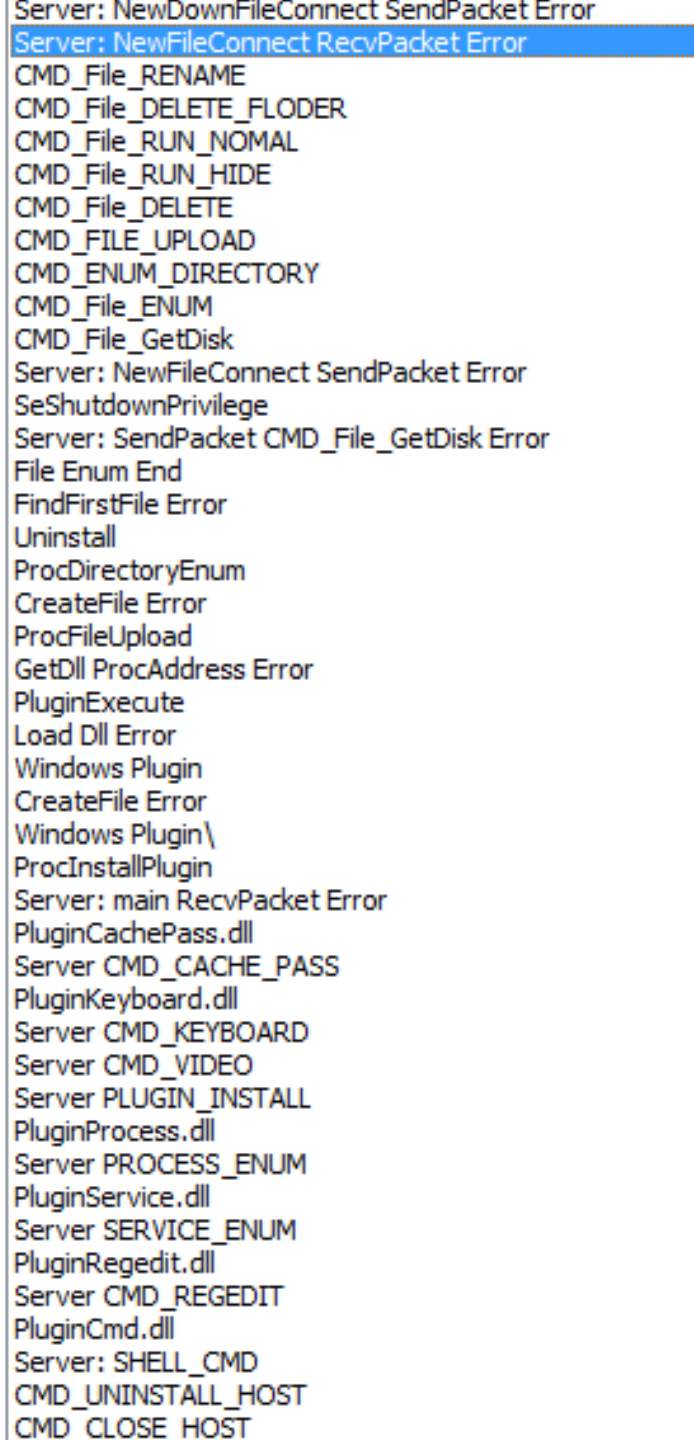
Evaluation: ASCII, cross references, character frequency count

String list detection

- string length + alingment
- string following w/o cross reference

Fitting strings into the graph

Whats the information one can gain from strings?



APIs

Cross references on symbols

Indirect calls

- parsing for mov/lea
- disassembling further
- call and jmp considered xref

Thunk pruning

Dynamic loading

```
[0x004344b6]> axt @@ sym.*
data 0x40e552 mov ebp, dword [sym.imp.KERNEL32.dll_LoadLibraryA] in fcn.00402db0
data 0x40e558 mov ebx, dword [sym.imp.KERNEL32.dll_GetProcAddress] in fcn.00402db0
call 0x4345de call dword [sym.imp.KERNEL32.dll_GetModuleHandleA] in entry0
data 0x4345de call dword [sym.imp.KERNEL32.dll_GetModuleHandleA] in entry0
call 0x4345ba call dword [sym.imp.KERNEL32.dll_GetStartupInfoA] in entry0
data 0x4345ba call dword [sym.imp.KERNEL32.dll_GetStartupInfoA] in entry0
call 0x401c3f call dword [sym.imp.GDI32.dll_RealizePalette] in fcn.00401040
data 0x401c3f call dword [sym.imp.GDI32.dll_RealizePalette] in fcn.00401040
call 0x401b5b call dword [sym.imp.GDI32.dll_CreateDIBSection] in fcn.00401040
call 0x401bd6 call dword [sym.imp.GDI32.dll_CreateDIBSection] in fcn.00401040
data 0x401b5b call dword [sym.imp.GDI32.dll_CreateDIBSection] in fcn.00401040
data 0x401bd6 call dword [sym.imp.GDI32.dll_CreateDIBSection] in fcn.00401040
call 0x401b6b call dword [sym.imp.GDI32.dll_IntersectClipRect] in fcn.00401040
data 0x401b6b call dword [sym.imp.GDI32.dll_IntersectClipRect] in fcn.00401040
call 0x401c5d call dword [sym.imp.GDI32.dll_CreateRectRgn] in fcn.00401040
data 0x401c5d call dword [sym.imp.GDI32.dll_CreateRectRgn] in fcn.00401040
call 0x401c4f call dword [sym.imp.GDI32.dll_GetBkMode] in fcn.00401040
data 0x401c4f call dword [sym.imp.GDI32.dll_GetBkMode] in fcn.00401040
call 0x401c47 call dword [sym.imp.GDI32.dll_CreateCompatibleDC] in fcn.00401040
data 0x401c47 call dword [sym.imp.GDI32.dll_CreateCompatibleDC] in fcn.00401040
data 0x401c2d mov esi, dword [sym.imp.GDI32.dll_SetPaletteEntries] in fcn.00401040
call 0x401c27 call dword [sym.imp.GDI32.dll_GetClipBox] in fcn.00401040
```

Indirect Calls

„Top-down“

Disassemble upwards
Check the arguments for function cross references
Add edge and tag
Currently only CreateThread and SetWindowsHookEx,
because context

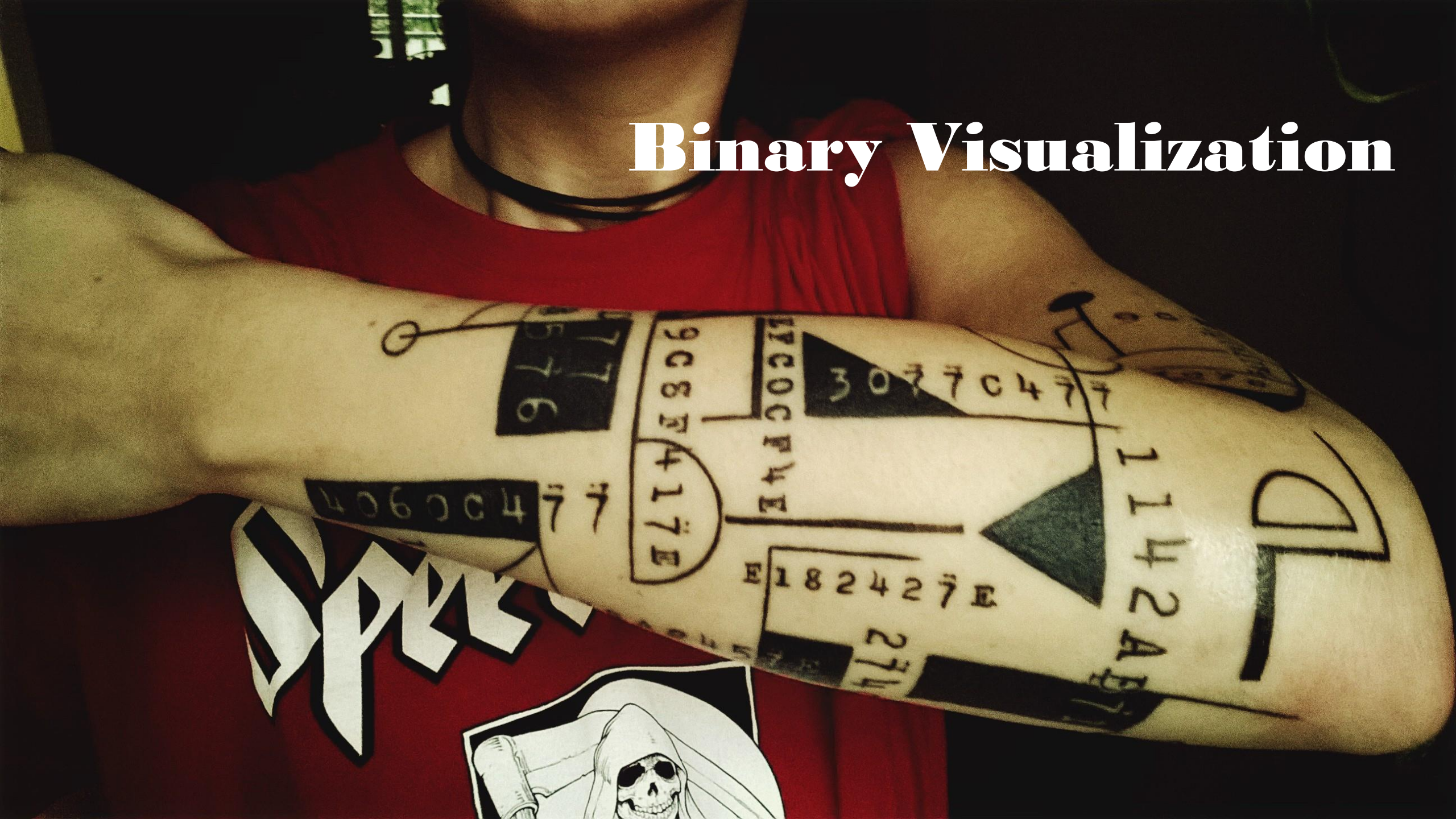
„Bottom-up“

Sweep for nodes without inbound edges
Check for cross references within functions
Add edge and tag

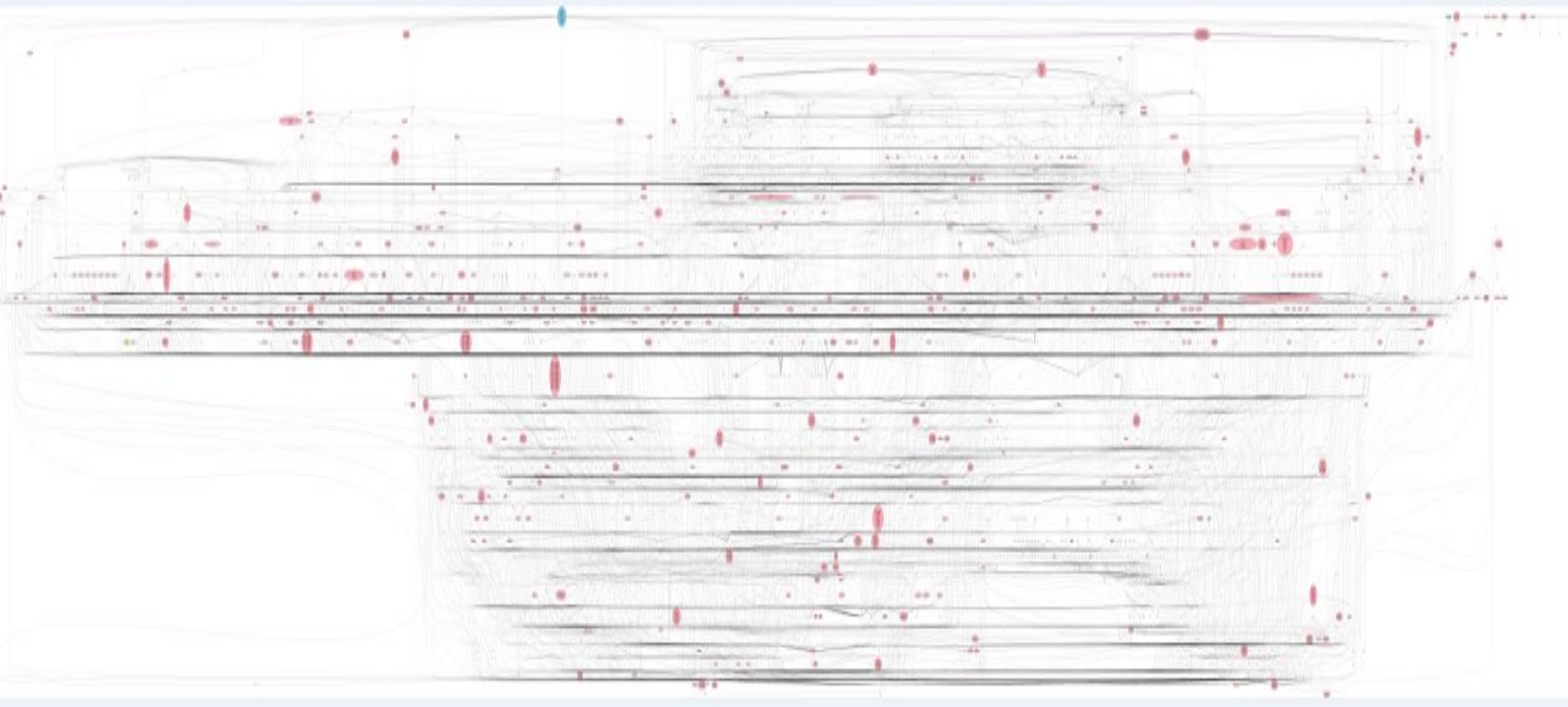
The r2graphity graph structure

```
### NetworkX Graph Structure ###

# FUNCTION as node, attributes: function address, size, calltype, list of calls, list of strings,
count of calls, functiontype[Callback, Export, Supernode], alias (e.g. export name), mnemonic
distribution

# FUNCTION REFERENCE as edge (function address -> target address), attributes: ref offset (at)

# INDIRECT REFERENCE as edge (currently for threads and Windows hooks, also indirect code and
indirect data references)

# API CALLS (list attribute of function node): address, API name

# STRINGS (list attribute of function node): address, string, eval

####
```

Binary Visualization



„Useful“ ain‘t easy



Large graphs, small graphs, dense graphs, lose graphs, dense subgraphs, disconnected subgraphs, ...

DLLs & GUI applications

Spaghetti code

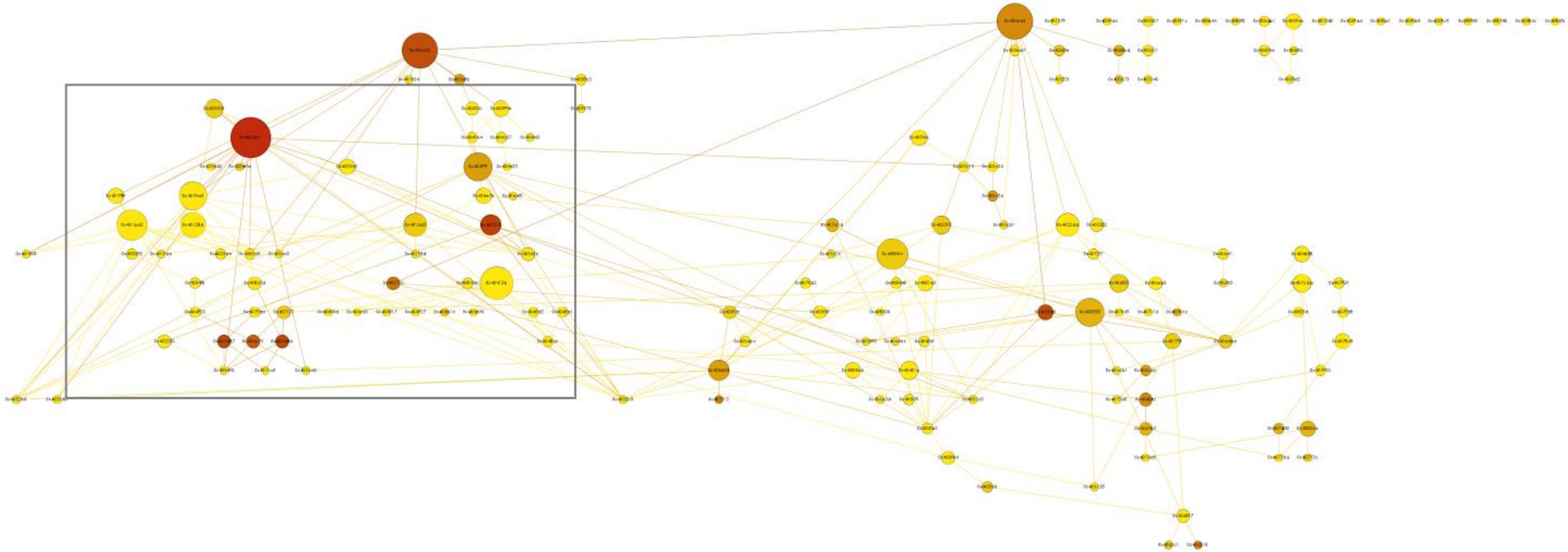
Copy/paste code

Packed code

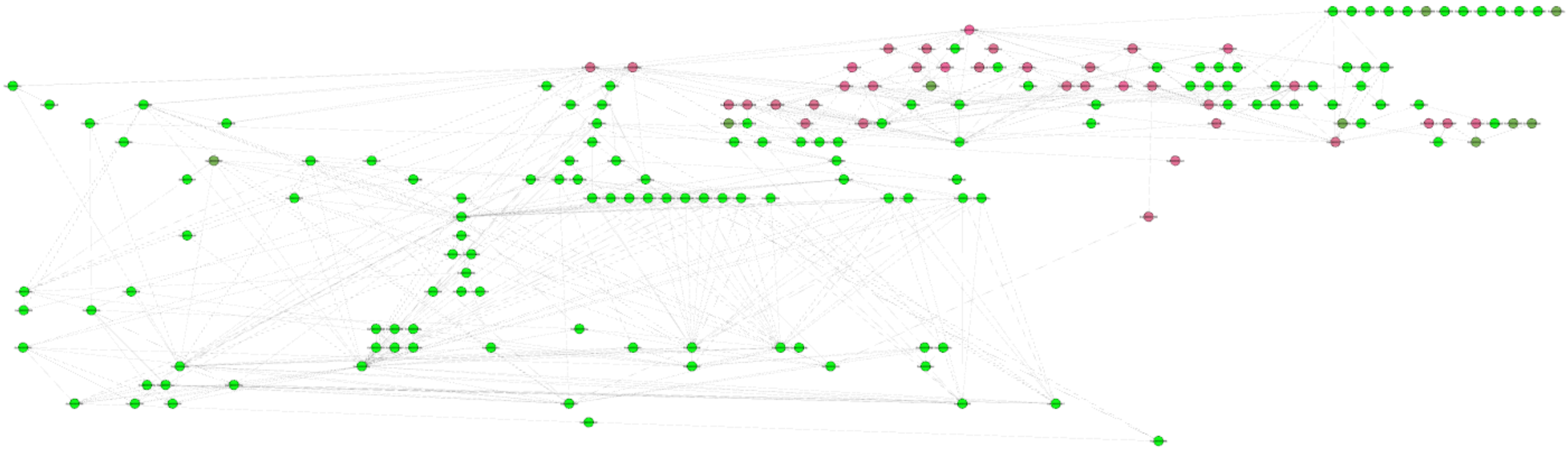
Repetitive patterns

Noise

Recovering code structure from call graphs

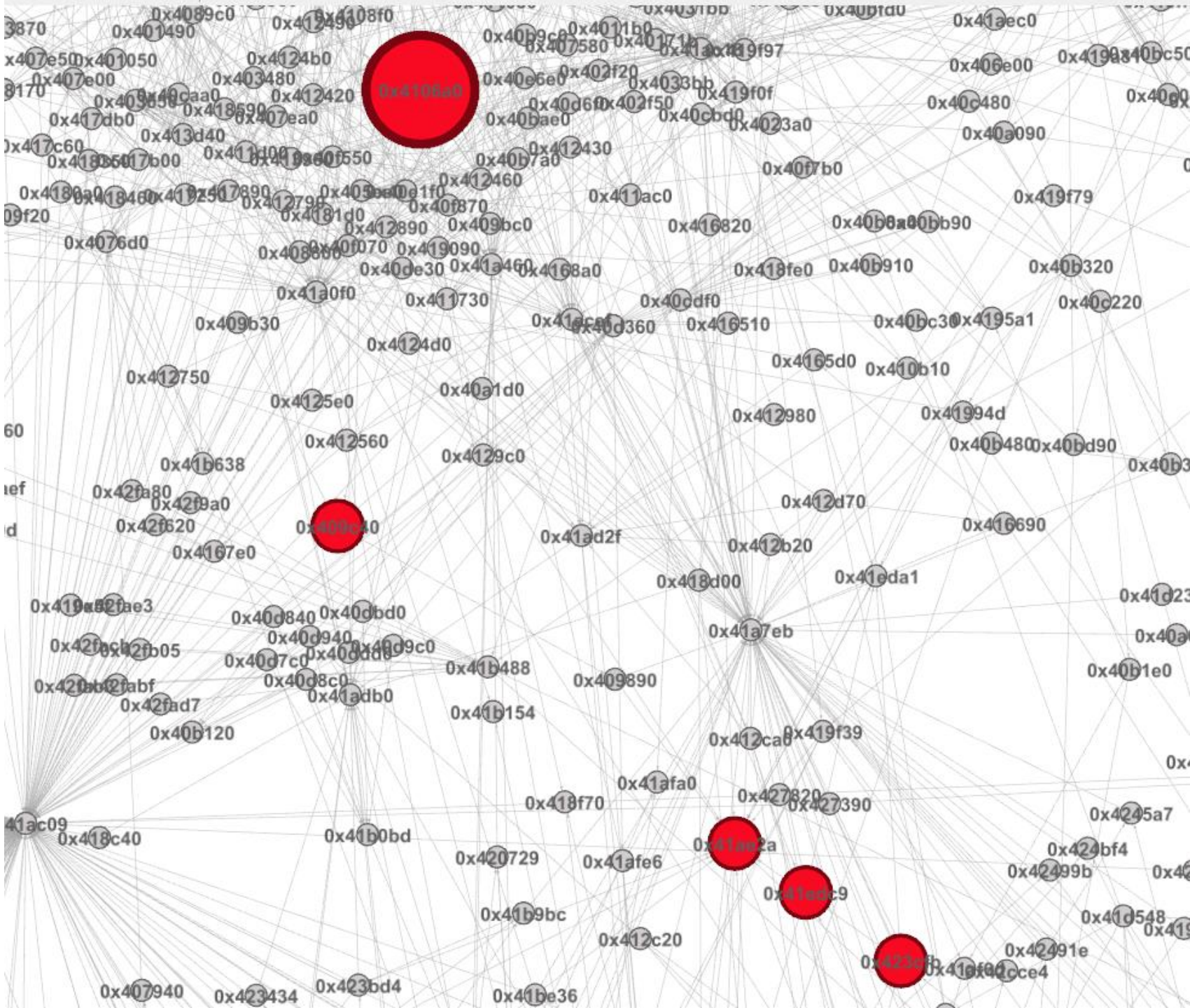


yellow: 0 API calls
gradually darker: plenty of API calls
node size: out-degree



green: 0 API calls
gradually darker: plenty of API calls

Highlighting memory allocation habits



How to deal with large graphs & too much information

Data reduction and simplification

How to pick features for visualization

- *know what your tools support*
- *what your algorithms support*
- *what your data has to say*

Layout algorithms

Graph transformations

API gadgets & highlighting

String evaluation

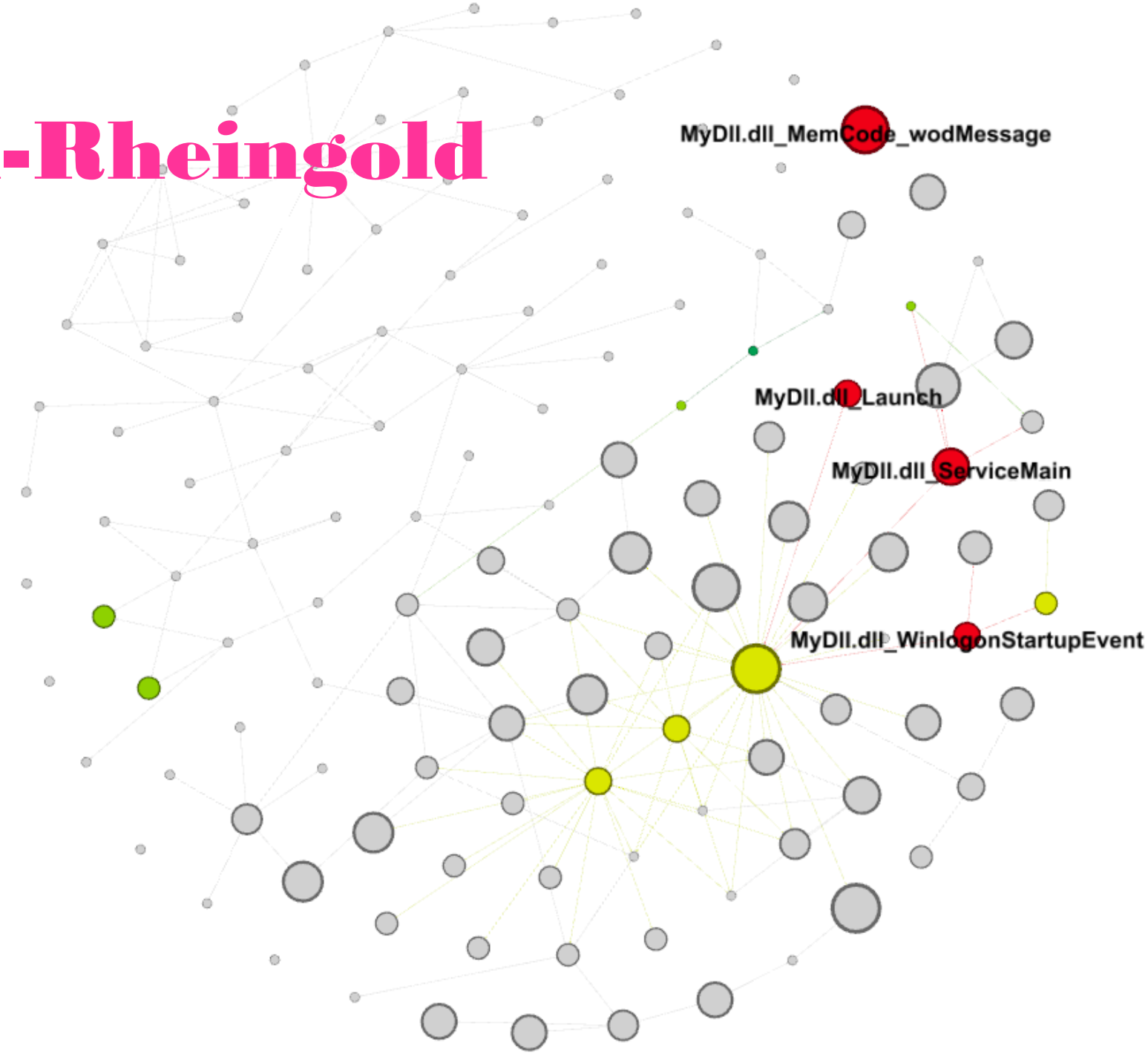
Fruchterman-Rheingold

Force directed

Neat overview

Slooow²

Find most important nodes at a glance



Force-directed graph layouts

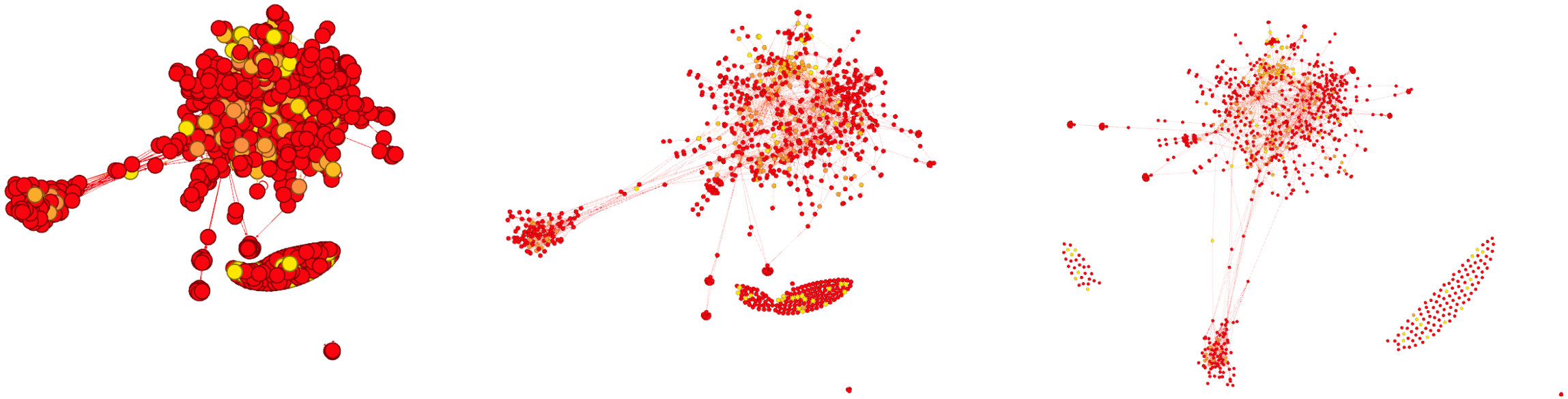
Position graph nodes in a way, that edges are in equal length and cross as little as possible

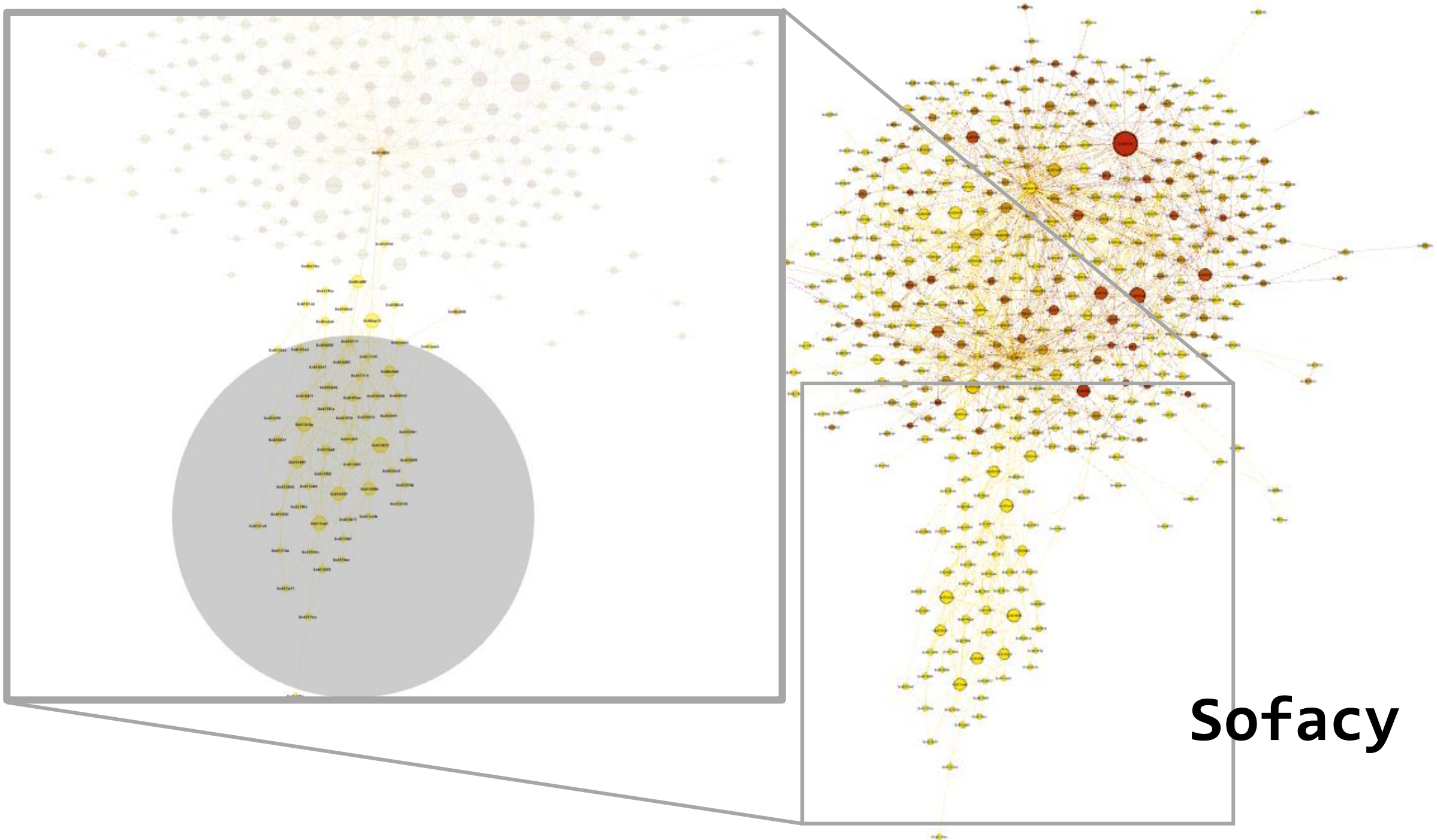
Forces can be applied, to pull less connected nodes further apart

High running time, high number of iterations

ForceAtlas

Repulsion and gravity



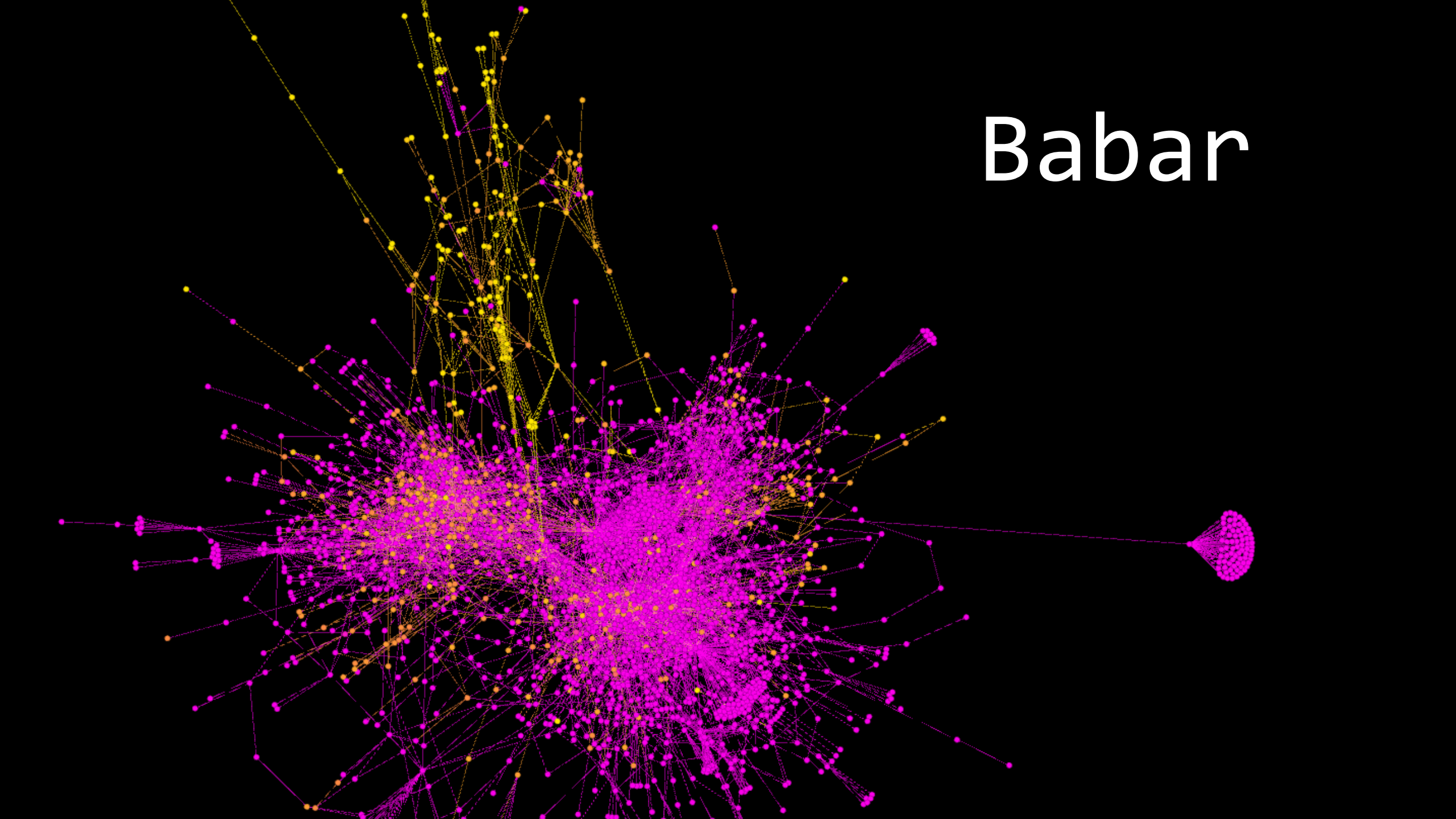

Sofacy

Mnemonicism

Arithmetic instructions as indicator for cryptography, compression or codecs

Leveraging radare2‘s instruction type

```
shl
shr
mul
div
rol
ror
sar
load
store
```

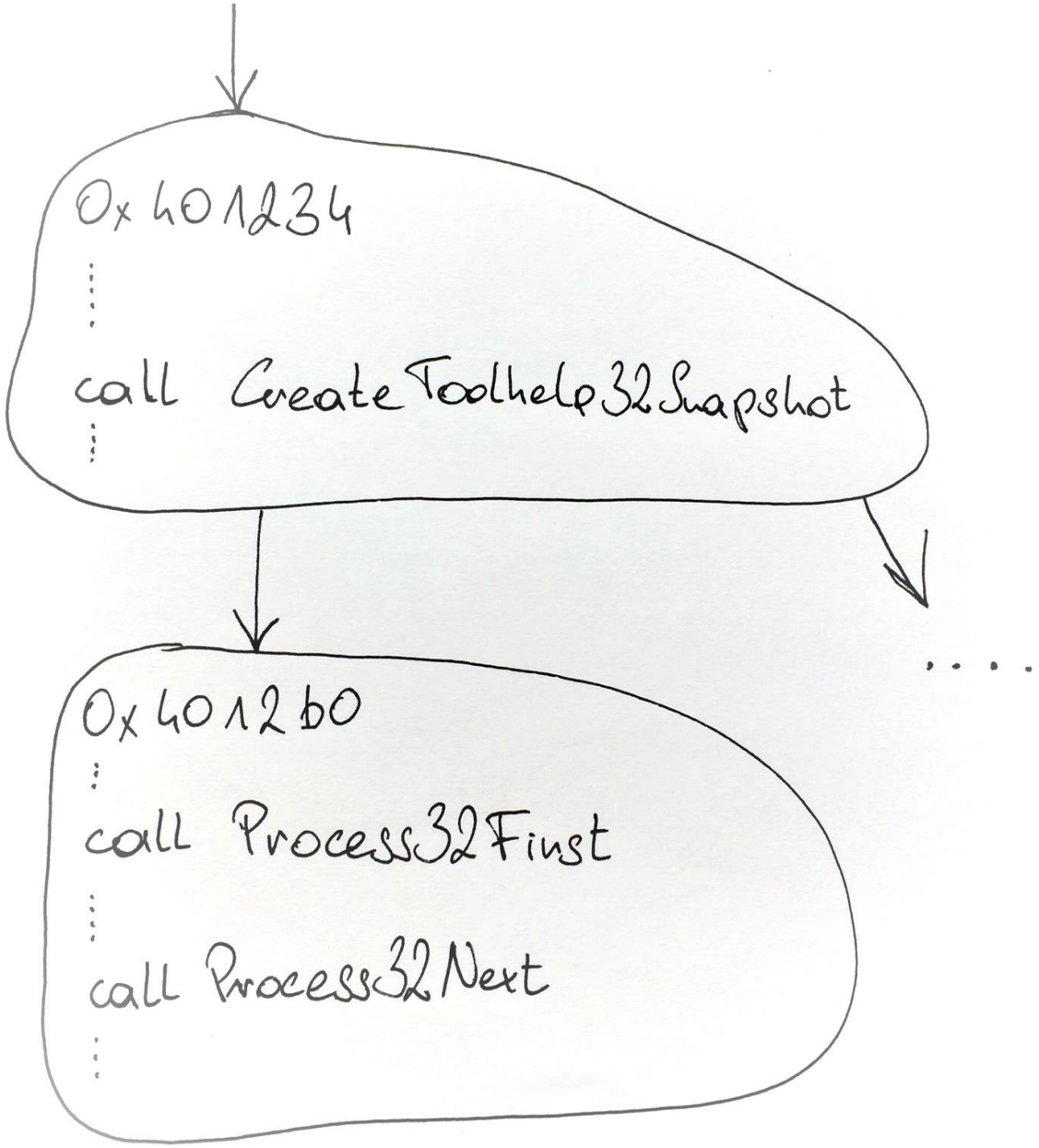

Babar

```
funcDict = {
    'DRIVERCOMM': ['DeviceIoControl'],
    'CREATESTARTSERVICE': ['OpenSCManager', 'CreateService', 'OpenService', 'StartService'],
    'CREATETHREAD': ['CreateThread'],
    'PROCESSITER': ['CreateToolhelp32Snapshot', 'Process32First', 'Process32Next'],
    'APILOADING': ['LoadLibrary', 'GetProcAddress'],
    'WRITEFILE': ['CreateFile', 'WriteFile'],
    'READFILE': ['CreateFile', 'ReadFile'],
    'WINHOOK': ['SetWindowsHookEx'],
    'DRIVESITER': ['GetLogicalDriveStrings', 'GetDriveType'],
    'FILEITER': ['FindFirstFile', 'FindNextFile', 'FindClose'],
    'REGSETVAL': ['RegOpenKey', 'RegSetValue'],
    'REGQUERY': ['RegOpenKey', 'RegQueryValue'],
    'DUMPRSRC': ['FindResource', 'LoadResource', 'CreateFile', 'WriteFile'],
    'LOADRSRC': ['FindResource', 'LoadResource', 'LockResource'],
    'WSASEND': ['WSAStartup', 'gethostbyname', 'send'],
    'RECV': ['recv', 'send'],
    'RETROINJECTION': ['GetCurrentProcess', 'CreatePipe', 'DuplicateHandle'],
    'WINEXEC': ['WinExec'],
    'SHELLEXEC': ['ShellExecute'],
    'CREATEPROC': ['CreateProcess'],
    'WINDOW': ['CreateWindow', 'RegisterClass', 'DispatchMessage'],
    'EXITSYSTEM': ['ExitWindows'],
    'TEMPFILEWRITE': ['GetTempFileName', 'CreateFile', 'WriteFile'],
    'REMTHREAD': ['CreateThread', 'WriteProcessMemory', 'ReadProcessMemory', 'ResumeThread'],
    'FPRINT': ['fopen', 'fprintf', 'fclose'],
    'UPDATERESOURCE': ['BeginUpdateResource', 'UpdateResource', 'EndUpdateResource'],
    'SCREENSHOT': ['CreateCompatibleDC', 'GetDeviceCaps', 'CreateCompatibleBitmap', 'BitBlt'],
    'CRYPT': ['CryptAcquireContext', 'CryptGenKey', 'CryptEncrypt']
}
```

“Behavior” Gadgets

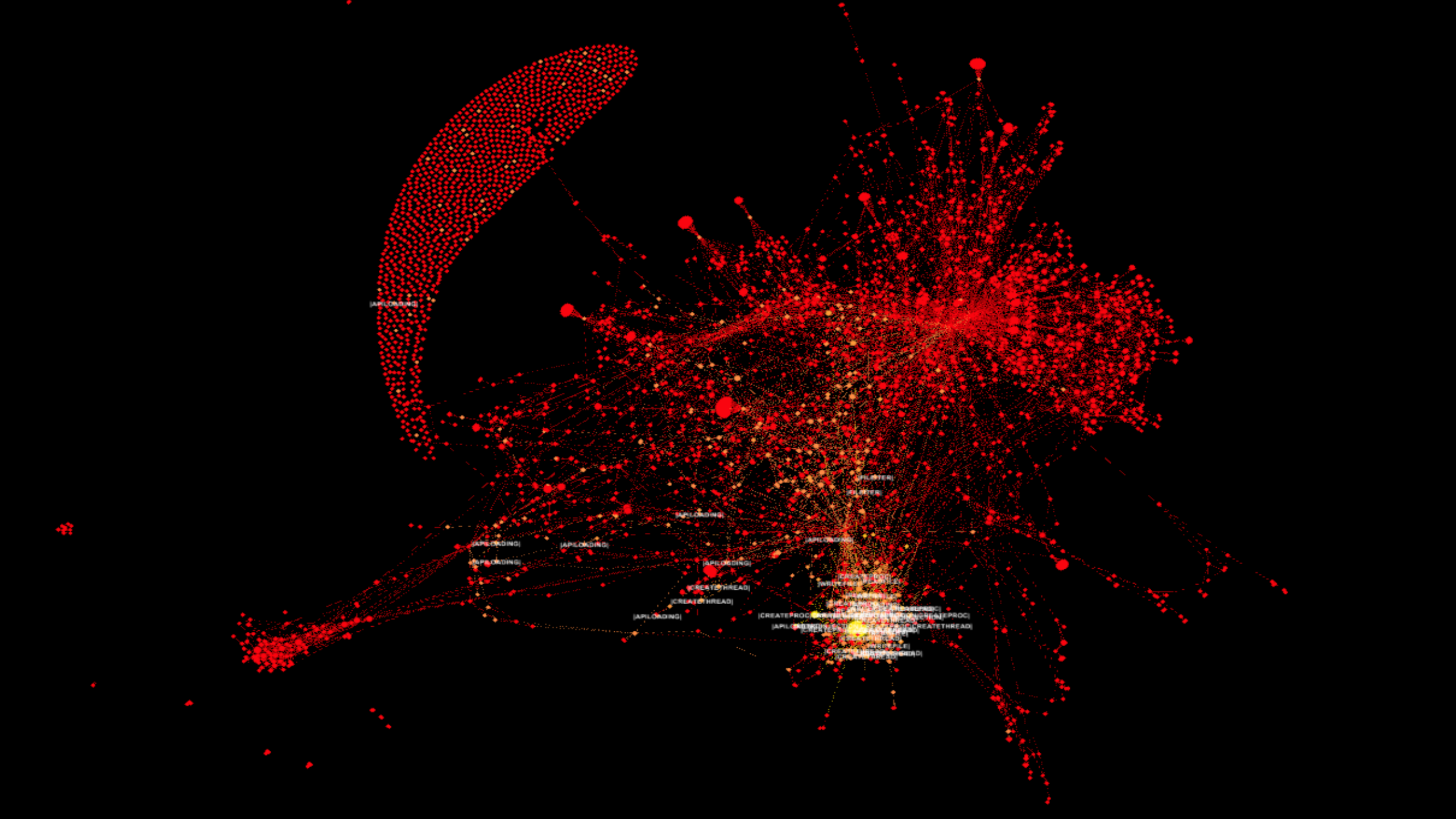
Scanning for Gadgets

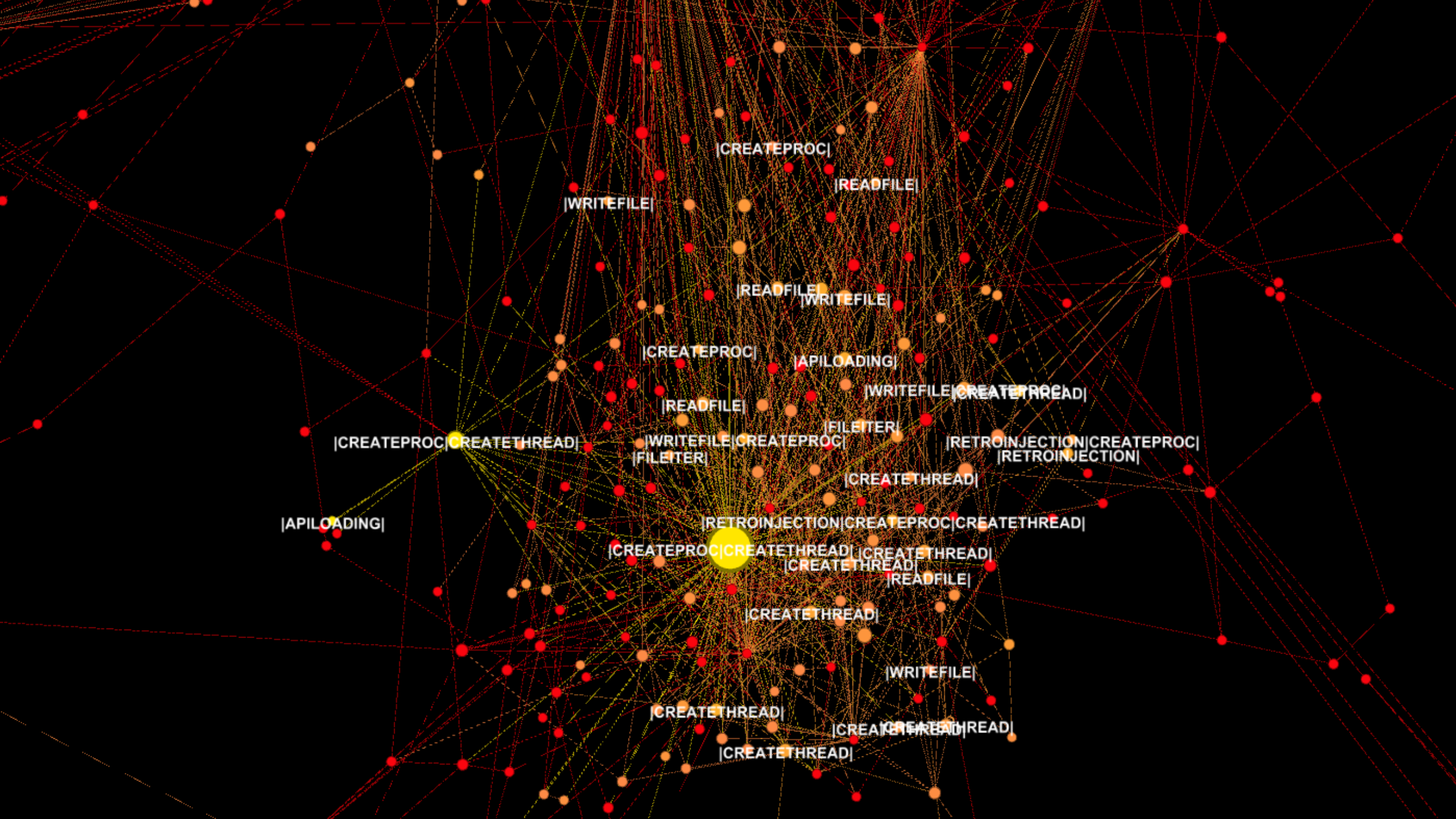
Pre-defined API patterns
Searching the graph for anchor
Scanning nodes in close vicinity

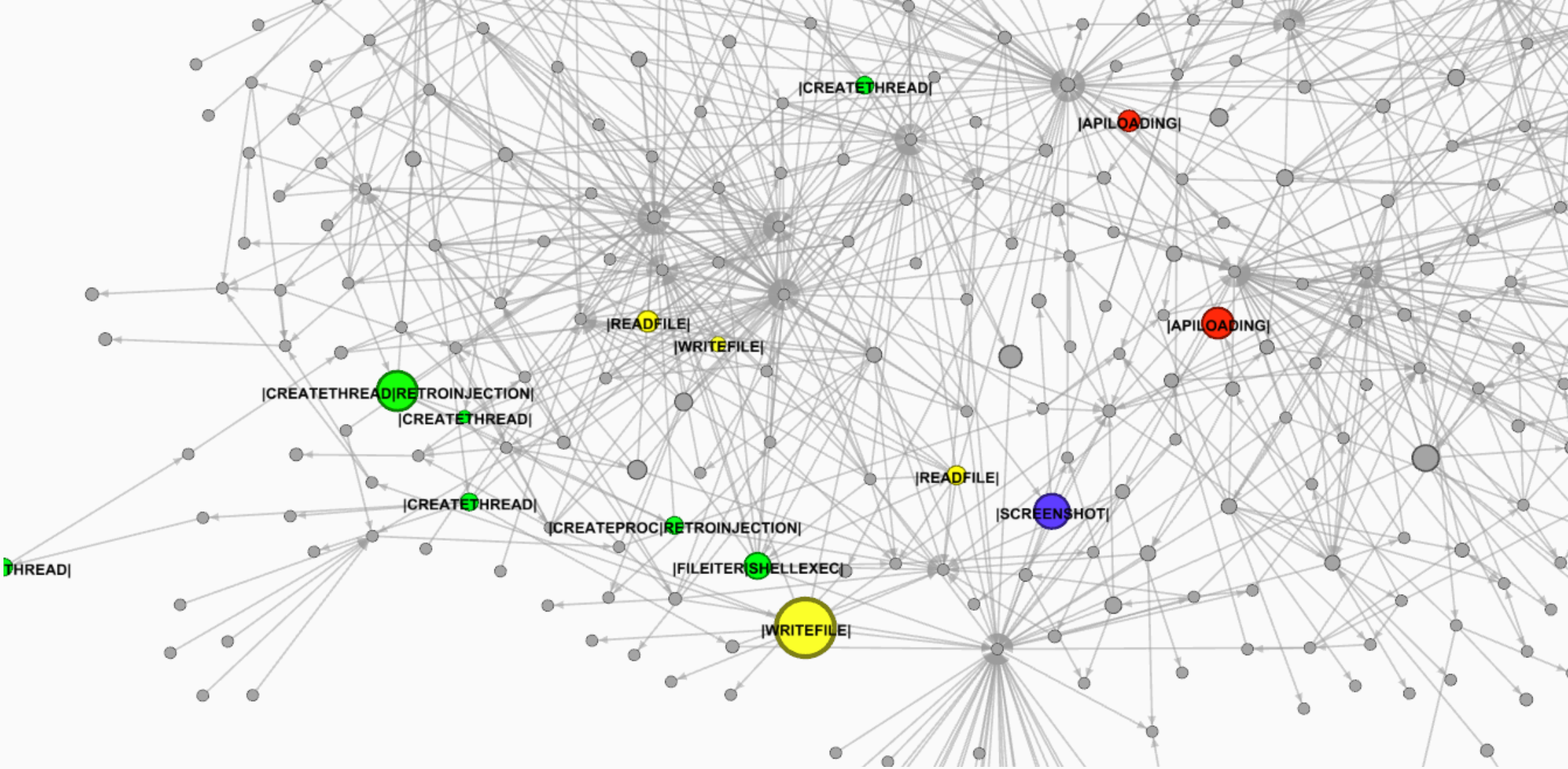


“Behavior” Gadgets

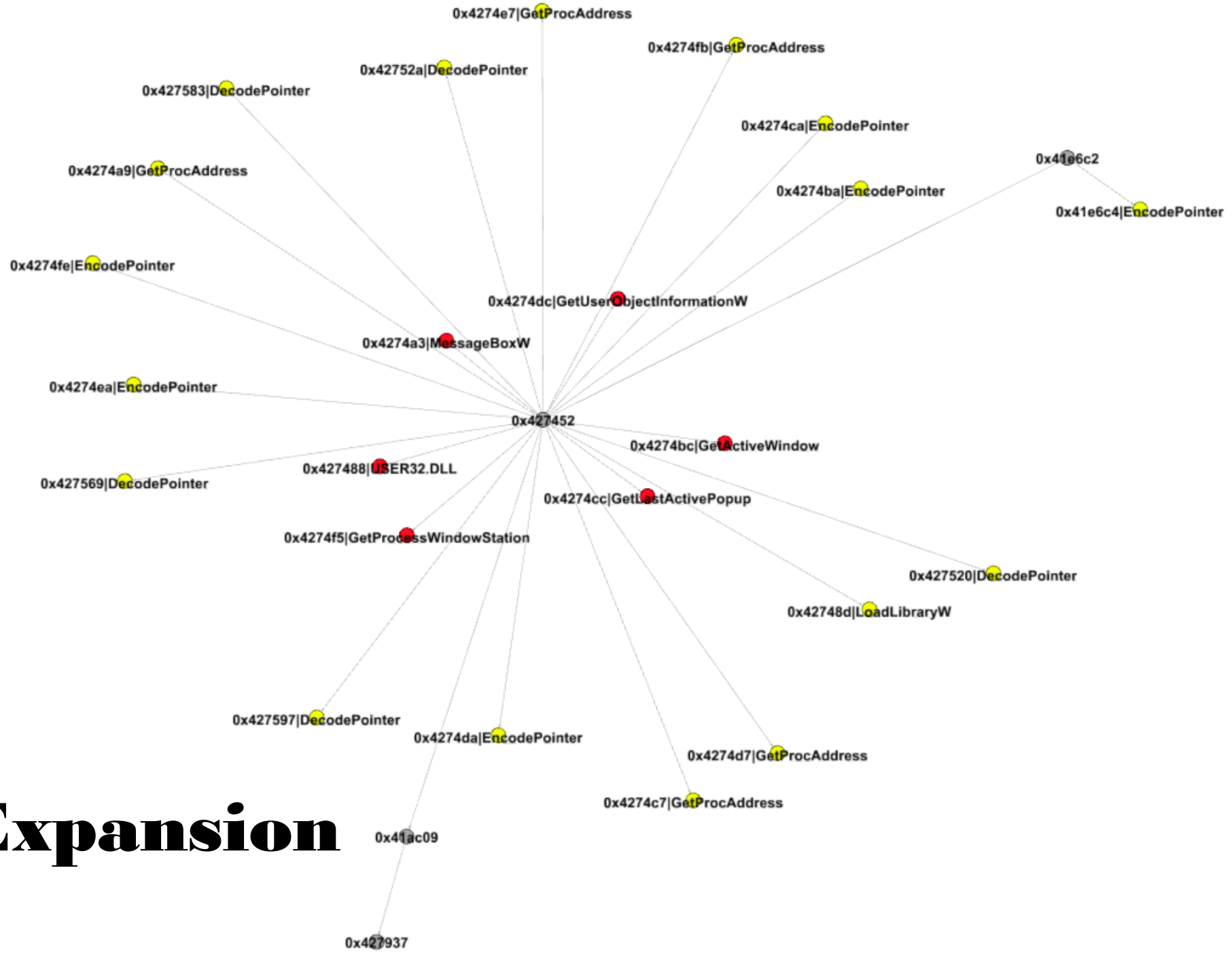
```
For APILOADING found {'GetProcAddress': '0x1000def8', 'LoadLibrary': '0x1000def8'}
For APILOADING found {'GetProcAddress': '0x10014e88', 'LoadLibrary': '0x10014e88'}
For READFILE found {'ReadFile': '0x100032a0', 'CreateFile': '0x100032a0'}
For READFILE found {'ReadFile': '0x1000d6b0', 'CreateFile': '0x1000d6b0'}
For APILOADING2 found {'GetModuleHandle': '0x1000fbd3', 'GetProcAddress': '0x1000fbd3'}
For APILOADING2 found {'GetModuleHandle': '0x1000f8ef', 'GetProcAddress': '0x1000fbd3'}
For APILOADING2 found {'GetModuleHandle': '0x10012552', 'GetProcAddress': '0x10012552'}
For SHELLEXEC found {'ShellExecute': '0x1000d330'}
For FILEITER found {'FindClose': '0x1000d330', 'FindFirstFile': '0x1000d330', 'FindNextFile': '0x1000d330'}
For CREATETHREAD found {'CreateThread': '0x1000ebc2'}
For CREATETHREAD found {'CreateThread': '0x10009b10'}
For CREATETHREAD found {'CreateThread': '0x10002190'}
For CREATETHREAD found {'CreateThread': '0x1000a050'}
For CREATETHREAD found {'CreateThread': '0x10001820'}
For CREATETHREAD found {'CreateThread': '0x10001000'}
For WRITEFILE found {'WriteFile': '0x1000d880', 'CreateFile': '0x1000d880'}
For WRITEFILE found {'WriteFile': '0x1000a4f0', 'CreateFile': '0x1000a4f0'}
For WRITEFILE found {'WriteFile': '0x10001f80', 'CreateFile': '0x10001f80'}
For RECV found {'recv': '0x1000b290', 'send': '0x1000b290'}
For SCREENSHOT found {'GetDeviceCaps': '0x100094d0', 'CreateCompatibleBitmap':
      '0x100094d0', 'BitBlt': '0x100094d0', 'CreateCompatibleDC': '0x100094d0'}
For REGQUERY found {'RegOpenKey': '0x10001000', 'RegQueryValue': '0x10001000'}
```


[APILOADING]
[APILOADING]
[APILOADING]
[APILOADING]
[APILOADING]
[APILOADING]
[APILOADING]
[CREATETHREAD]
[CREATETHREAD]
[APILOADING]
[CREATEPROC
[CREATETHREAD]


|CREATEPROC|
|READFILE|
|WRITEFILE|
|READFILE|
|WRITEFILE|
|CREATEPROC|
|APILOADING|
|WRITEFILE|CREATETHREAD|
|READFILE|
|FILEITER|
|CREATEPROC|CREATETHREAD|
|WRITEFILE|CREATEPROC|
|FILEITER|
|RETROINJECTION|CREATEPROC|
|RETROINJECTION|
|CREATETHREAD|
|APILOADING|
|RETROINJECTION|CREATEPROC|CREATETHREAD|
|CREATEPROC|CREATETHREAD|
|CREATETHREAD|
|CREATETHREAD|
|READFILE|
|CREATETHREAD|
|WRITEFILE|
|CREATETHREAD|
|CREATETHREAD|
|CREATETHREAD|



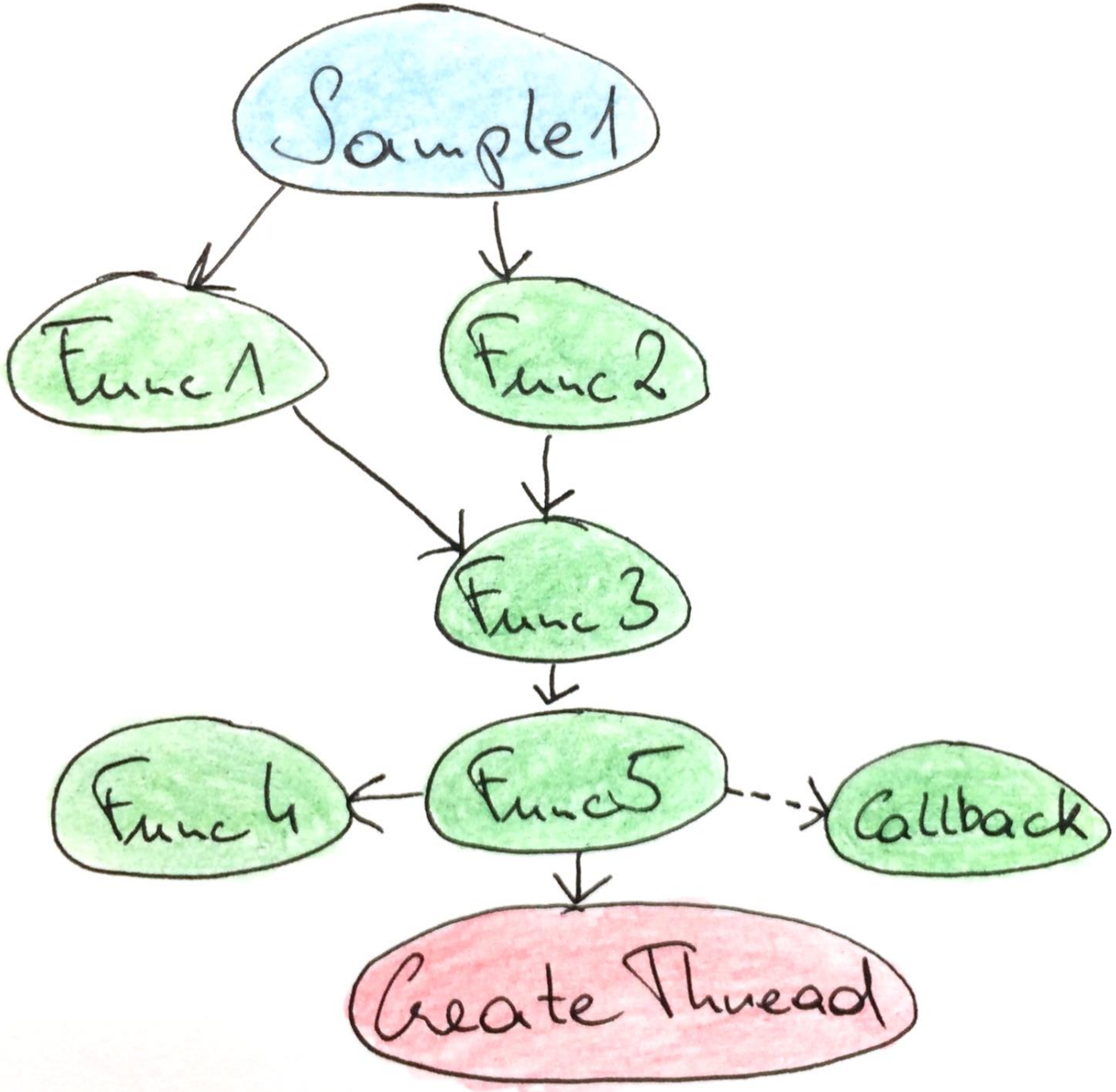
Color-code functionality families

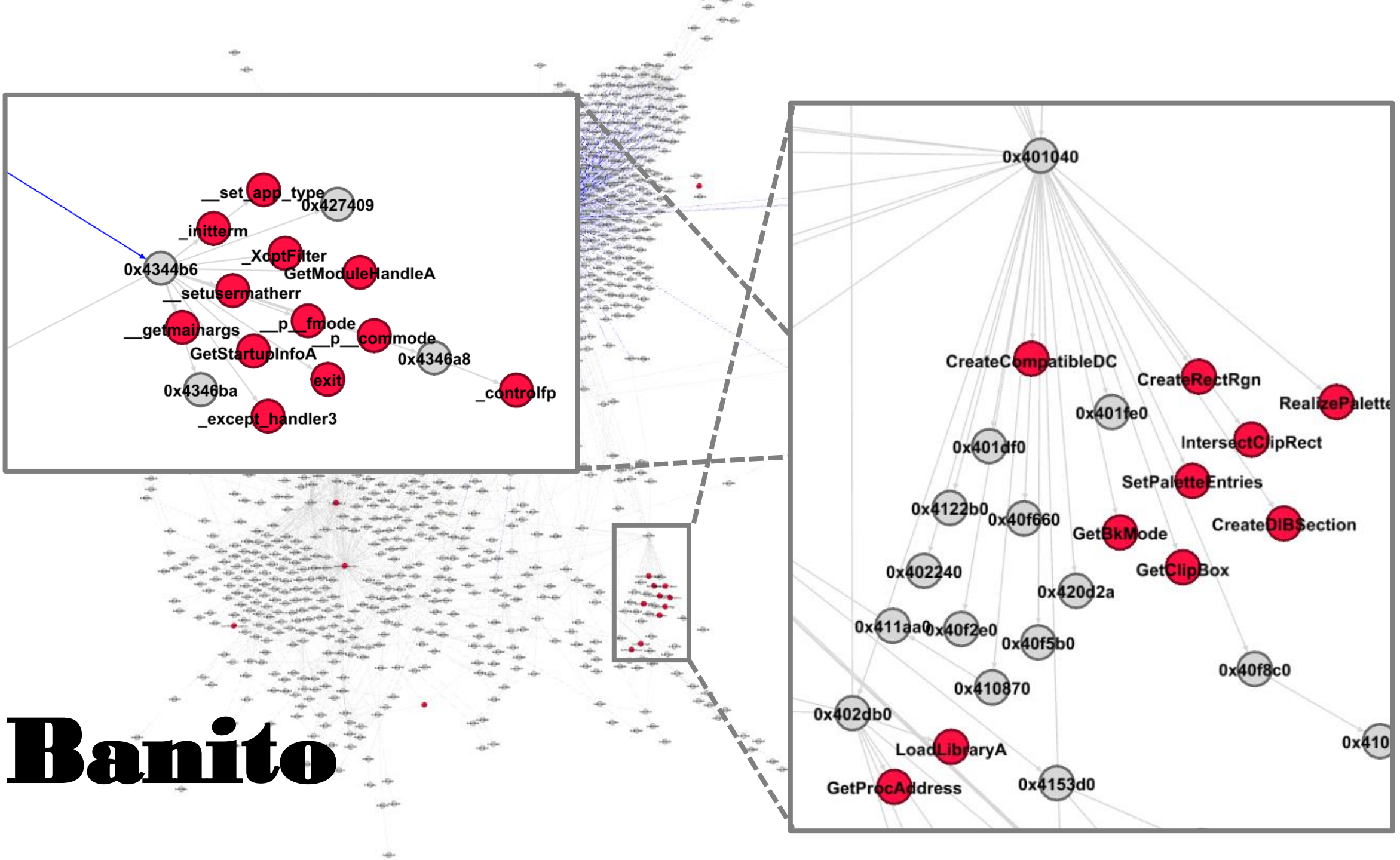


Grey: functions
Yellow: API calls
Red: strings

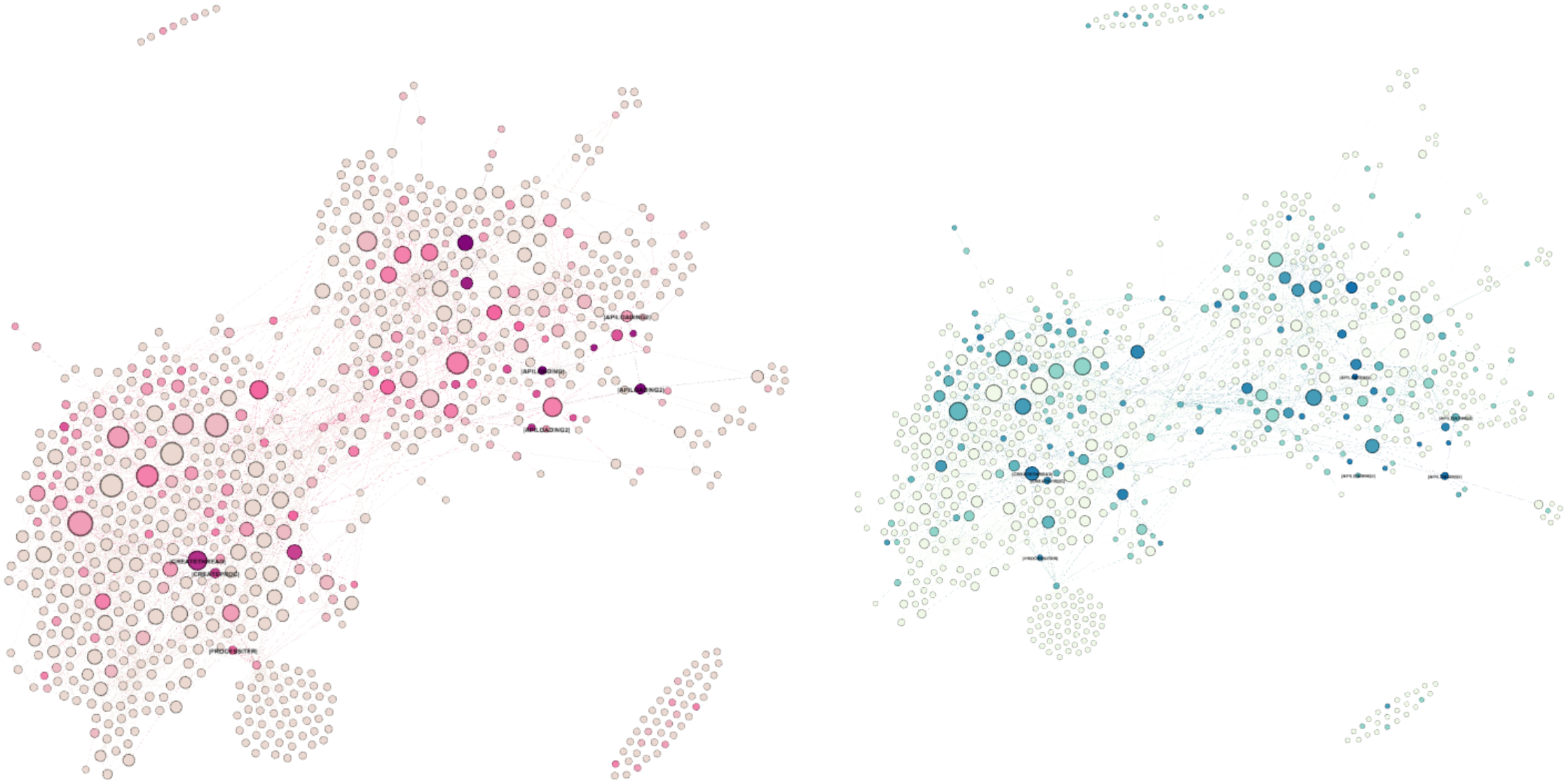
Subgraph Expansion

Expansion Transformation




__set_app_type
0x427409
_initterm
_XcptFilter
0x4344b6
GetModuleHandleA
__setusermatherr
__getmainargs
__p__fmode
__p__commode
GetStartupInfoA
0x4346a8
exit
0x4346ba
_controlfp
_except_handler3
0x401040
CreateCompatibleDC
CreateRectRgn
RealizePalette
0x401fe0
0x401df0
IntersectClipRect
SetPaletteEntries
0x4122b0
0x40f660
GetBkMode
CreateDIBSection
0x402240
GetClipBox
0x420d2a
0x411aa0
0x40f2e0
0x40f5b0
0x40f8c0
0x410870
0x402db0
LoadLibraryA
0x410
GetProcAddress
0x4153d0
Banito

Similarity Visualization: Animalfarm Binaries



AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX
functionstotal	refslocal	refsglobalva	refsunknown	apitotal	apimisses	stringsreferenced	stringsdangling	stringsnoref	ratiofunc	ratioapi	ratiostring	getprocaddress	memallocation	createthread	ctshortestpath	callbackcount	cbaveragesize	cblargestsize	stringsrefhisto
124	715	1	0	183	3	30	2	264	9.6875	14.296875	2.34375	0	88	3	2	2	467	612	2-0-2-4-1-4-2-5-1
543	1730	3	1	437	0	100	0	1178	7.798138786764706	6.275850183823529	1.4361213235294117	11	8	11	2	11	145	556	2-8-2-17-17-6-13-17-14
1611	4311	4	1	601	1	100	0	2646	8.620505136986301	3.2159674657534247	0.5351027397260274	12	10	8	2	8	181	551	4-2-5-10-35-1-0-4-36
1218	3091	3	1	409	0	84	0	1739	9.079794847328245	3.0489623091603053	0.6261927480916031	11	5	7	2	7	196	551	2-0-7-9-31-0-0-3-30
1712	4431	4	1	584	1	106	0	2666	8.845899470899472	3.017526455026455	0.5477017195767195	11	10	9	2	8	180	551	2-2-7-12-37-1-0-4-38
1650	4317	4	1	583	0	114	0	2786	8.733485772357724	3.0858316395663956	0.6034044715447154	11	10	9	2	8	180	551	2-8-8-11-39-1-0-4-38
1503	3825	3	1	563	3	107	0	2220	8.86872167673716	3.3220827039274923	0.6313727341389728	15	10	9	2	8	170	469	2-4-7-13-37-5-0-3-34
1788	4649	4	1	598	0	123	0	2736	8.774340452261306	2.934594849246231	0.6036039572864321	13	10	9	2	8	170	469	3-5-7-16-40-6-0-4-38
1678	4331	4	1	530	0	115	0	2868	8.739583333333334	2.7604166666666665	0.5989583333333334	11	10	8	2	8	163	469	2-5-7-14-36-5-0-3-38
1304	3331	3	1	425	0	102	0	1950	8.93640350877193	2.9125548245614037	0.699013157894737	11	5	7	2	7	184	469	3-6-7-12-35-4-0-3-30
1513	3082	3	1	384	1	94	0	2316	10.90434732472324	2.7675276752767526	0.6774677121771218	14	5	4	2	4	118	219	2-4-7-13-29-5-0-3-29
1436	2921	3	1	371	0	81	0	2317	10.78725961538461	2.7869591346153846	0.6084735576923077	12	5	4	2	4	118	219	2-4-7-7-27-0-0-3-29
1445	2936	3	1	374	7	86	0	2537	10.81327825670498	2.798730842911877	0.6435584291187739	12	5	4	2	4	118	219	3-4-7-10-27-0-0-3-29
1511	3095	3	1	376	0	91	0	2459	10.84989659926470	2.699908088235294	0.6534352022058824	12	5	4	2	4	118	219	3-4-7-12-27-4-0-3-29
4255	20499	21	30	690	2	5134	21	30583	2.724769467213115	0.4418545081967213	3.2876536885245904	63	5	2	5	2	119	185	13-233-274-464-276-1381-1895-265-190
4255	20499	21	30	690	2	5134	21	30638	2.724769467213115	0.4418545081967213	3.2876536885245904	63	5	2	5	2	119	185	13-233-274-464-276-1381-1895-265-190
3624	6273	3	1	869	0	117	0	4252	10.70820726172466	2.5677240922844176	0.3457119894099849	66	4	3	2	1	173	173	2-2-5-11-25-1-0-4-46
3623	6272	3	1	866	0	117	0	4239	10.72147253787878	2.562736742424242	0.3462357954545454	66	4	4	2	1	173	173	2-2-5-11-25-1-0-4-46
3638	6696	3	1	875	0	117	0	6986	8.102016818700115	1.9486708950969214	0.2605651368301026	66	3	4	2	2	119	173	2-2-5-11-25-1-0-4-46
3639	6698	3	1	873	0	117	0	6995	8.095013525056947	1.9420024202733486	0.2602683656036446	66	3	3	2	2	119	173	2-2-5-11-25-1-0-4-46
3639	6698	3	1	873	0	117	0	7005	8.095013525056947	1.9420024202733486	0.2602683656036446	66	3	3	2	2	119	173	2-2-5-11-25-1-0-4-46
295	859	6	1	305	2	37	0	1367	8.864182692307692	9.164663461538462	1.111778846153846	15	16	1	5	1	161	161	3-0-3-8-13-0-1-3-2
247	720	6	1	296	0	38	0	1245	8.614676339285714	10.323660714285714	1.3253348214285714	15	21	1	5	1	161	161	3-1-3-8-13-0-1-3-2
246	699	6	1	289	0	38	0	1245	8.579799107142858	10.079520089285714	1.3253348214285714	15	21	1	5	1	161	161	3-1-3-8-13-0-1-3-2
3940	17932	15	30	627	1	2950	27	24859	2.989631895881896	0.4757612179487179	2.238429972804973	63	5	1	5	1	125	125	12-195-121-175-177-769-1319-75-49
3950	17779	15	30	627	1	2973	2	27944	2.971819626348228	0.471729343220339	2.2367644934514637	63	5	1	5	1	125	125	12-195-122-175-177-784-1324-76-50
3572	15520	15	30	589	0	2935	5	17840	4.794888316151202	0.7906464776632302	3.9398088487972505	35	6	1	7	1	101	101	12-194-123-163-184-786-1308-81-49
3573	15503	15	30	591	0	2919	14	18760	4.792936555631869	0.7927863152472527	3.9156400240384617	35	6	1	5	1	101	101	12-195-120-156-188-781-1308-76-47
3700	16162	15	31	536	2	2908	1	19165	4.638358472400514	0.6719351732991014	3.6454990372272142	35	6	1	6	1	101	101	12-195-121-158-163-785-1323-73-43
3475	15342	15	30	578	0	2856	49	19078	4.67431775137741	0.7774836432506887	3.8416838842975207	35	6	1	5	1	101	101	12-195-122-157-187-770-1255-76-48
3486	15385	15	30	567	0	2919	13	18950	4.685886958017893	0.7621623365450791	3.9237246214728145	35	6	1	5	1	101	101	12-195-123-156-183-769-1324-73-49
3551	15594	15	29	526	0	2789	52	17439	4.763425051510989	0.7055932348901099	3.741253863324176	35	6	1	5	1	101	101	9-193-101-134-160-757-1277-76-48
3573	15686	15	30	591	24	2931	16	18491	4.783081305688827	0.79115618574366	3.923652758738862	33	6	1	5	1	101	101	12-195-121-160-189-786-1304-81-49
434	1247	3	1	318	0	37	0	991	5.927666083916084	4.3433129370629375	0.505354020979021	10	5	1	6	1	101	101	2-1-2-10-5-0-0-3-9
430	1244	3	1	317	0	39	0	1003	5.873033216783217	4.329654720279721	0.5326704545454546	10	5	1	6	1	101	101	2-1-2-10-7-0-0-3-9
823	2836	4	2	669	0	135	0	1467	5.007544781931464	4.070531542056075	0.8214077102803738	10	6	1	6	1	101	101	2-7-25-22-10-6-4-20-27
823	2836	4	2	669	0	135	0	1467	5.007544781931464	4.070531542056075	0.8214077102803738	10	6	1	6	1	101	101	2-7-25-22-10-6-4-20-27
238	937	0	0	158	0	44	0	537	9.486607142857142	6.297831632653061	1.753826530612245	0	1	2	2	1	100	100	11-15-9-2-4-1-0-0-1
234	829	0	0	291	0	31	0	583	8.161272321428571	10.149274553571429	1.0811941964285714	3	1	3	2	2	69	92	4-11-6-4-5-0-0-0-0
234	829	0	0	291	0	34	0	568	8.161272321428571	10.149274553571429	1.1858258928571428	3	1	3	2	2	78	92	5-11-6-5-5-0-0-0-0
234	829	0	0	291	0	34	0	568	8.161272321428571	10.149274553571429	1.1858258928571428	3	1	3	2	2	78	92	5-11-6-5-5-0-0-0-0
235	830	0	0	295	0	38	0	560	8.196149553571429	10.288783482142858	1.3253348214285714	3	1	3	2	2	78	92	5-9-11-5-6-0-0-0-0
225	781	0	0	138	0	29	0	480	10.21984011627907	6.268168604651163	1.3172238372093024	0	1	3	2	2	69	92	7-9-4-2-4-1-0-0-0
1299	3376	4	1	417	0	96	0	2350	8.513789848993289	2.733064177852349	0.6291946308724833	11	10	4	2	4	67	86	3-1-6-11-32-1-0-3-36
1298	3371	4	1	415	0	106	0	2391	8.53587962962963	2.7291140572390575	0.6970749158249159	11	10	4	2	4	67	86	3-4-13-11-32-1-0-3-36
222	780	0	0	118	0	33	0	487	10.32366071428571	5.48735119047619	1.5345982142857142	0	1	1	2	1	85	85	5-16-7-0-3-0-0-0-0
214	840	0	0	255	0	24	0	458	8.359375	9.9609375	0.9375	3	1	3	2	2	68	84	3-4-7-4-4-0-0-0-1
196	768	0	0	255	0	29	0	477	8.144946808510639	10.596742021276595	1.2051196808510638	3	1	3	2	2	68	84	3-5-12-5-2-0-0-0-1
195	765	0	0	260	0	33	0	477	7.9345703125	10.579427083333334	1.3427734375	3	2	3	2	2	68	84	3-5-12-5-4-1-0-0-0
228	915	0	0	298	0	40	0	536	7.952008928571429	10.393415178571429	1.3950892857142856	3	1	3	2	2	82	84	4-8-13-6-5-1-0-0-1
229	909	0	0	140	0	30	0	460	9.318033854166666	5.696614583333333	1.220703125	0	1	1	2	1	57	57	4-5-4-13-2-0-0-1-0
229	909	0	0	140	0	30	0	460	9.318033854166666	5.696614583333333	1.220703125	0	1	1	2	1	57	57	4-5-4-13-2-0-0-1-0
228	922	0	0	121	0	34	0	457	9.474734042553191	5.0282579787234045	1.4128989361702127	0	1	1	2	1	57	57	4-8-5-12-2-1-0-1-0

String Constants

Human readable strings give information away

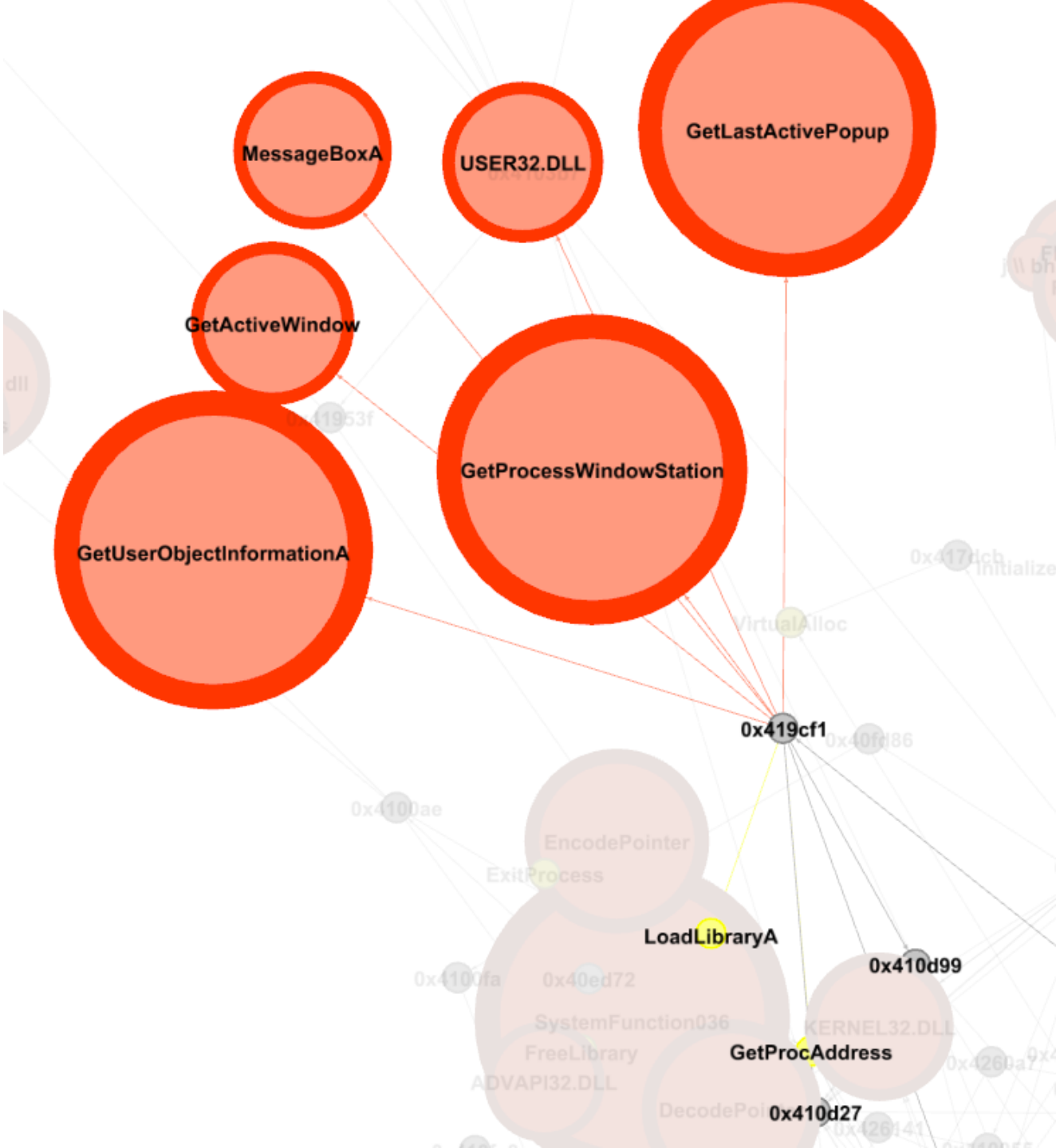
Presence or absence of readable strings is relevant information

Graph structure, character frequency and character repetition allow string constant evaluation

```
freqs = {
    'a': 0.0651738,
    'b': 0.0124248,
    'c': 0.0217339,
    'd': 0.0349835,
    'e': 0.1041442,
    'f': 0.0197881,
    'g': 0.0158610,
    'h': 0.0492888,
    'i': 0.0558094,
    'j': 0.0109033,
    'k': 0.0150529,
    'l': 0.0331490,
    'm': 0.0202124,
    'n': 0.0564513,
    'o': 0.0596302,
    'p': 0.0137645,
    'q': 0.0058606,
    'r': 0.0497563,
    's': 0.0515760,
    't': 0.0729357,
    'u': 0.0225134,
    'v': 0.0182903,
    'w': 0.0271272,
    'x': 0.0013692,
    'y': 0.0145984,
    'z': 0.0017836,
    ' ': 0.0500000,
    '0': 0.0500000,
    '1': 0.0500000,
    '2': 0.0500000,
    '3': 0.0500000,
    '4': 0.0500000,
    '5': 0.0500000,
    '6': 0.0500000,
    '7': 0.0500000,
    '8': 0.0500000,
    '9': 0.0500000,
    '.': 0.0400000,
    '_': 0.0400000
}
```

lnkfile \\ shellex \\ IconHandler	0.0897536969696697
OptionFlags	0.0457972
Progman	0.04012135714285714
_^Ã‹L$	0.014629799999999998
<A| \\ b<Z	0.017938219999999998
<A| \\ b<Z	0.017938219999999998
0123456789abcdefghijklmnopqrstuvwxyzABC	0.0702613625
_^][	0.01
_^][	0.01
SUVW	0.029876725
\\ *.*	0.02
X_^[	0.0103423
\\ StringFileInfo \\ %s \\ FileVersion	0.08549147692307693
%08X	0.0253423
\\ VarFileInfo \\ Translation	0.09178884
_^Ã‹L$	0.014629799999999998
SHELL32.DLL	0.04659895454545454534
SHGetFolderLocation	0.1073442631578947
State	0.07335308
_^][	0.01
3É,`|	0.0125
3É,`|	0.0125
3É,`|	0.0125

Sizing string nodes by „readability“



String character frequency histogram per sample

Bucketsize of 0.01

Count of strings per bucket

0.04 is a reasonable edge

Resilient to little changes

Subset of Sofacy

2-0-7-9-31-0-0-3-30
2-2-7-12-37-1-0-4-38
2-8-8-11-39-1-0-4-38
2-4-7-13-37-5-0-3-34
3-5-7-16-40-6-0-4-38
2-5-7-14-36-5-0-3-38
3-6-7-12-35-4-0-3-30
2-4-7-13-29-5-0-3-29
2-4-7-7-27-0-0-3-29
3-4-7-10-27-0-0-3-29
3-4-7-12-27-4-0-3-29
13-233-274-464-276-1381-1895-265-190
13-233-274-464-276-1381-1895-265-190
2-2-5-11-25-1-0-4-46
2-2-5-11-25-1-0-4-46
2-2-5-11-25-1-0-4-46
2-2-5-11-25-1-0-4-46
2-2-5-11-25-1-0-4-46
3-0-3-8-13-0-1-3-2
3-1-3-8-13-0-1-3-2
3-1-3-8-13-0-1-3-2
12-195-121-175-177-769-1319-75-49
12-195-122-175-177-784-1324-76-50
12-194-123-163-184-786-1308-81-49
12-195-120-156-188-781-1308-76-47
12-195-121-158-163-785-1323-73-43
12-195-122-157-187-770-1255-76-48
12-195-123-156-183-769-1324-73-49
9-193-101-134-160-757-1277-76-48
12-195-121-160-189-786-1304-81-49

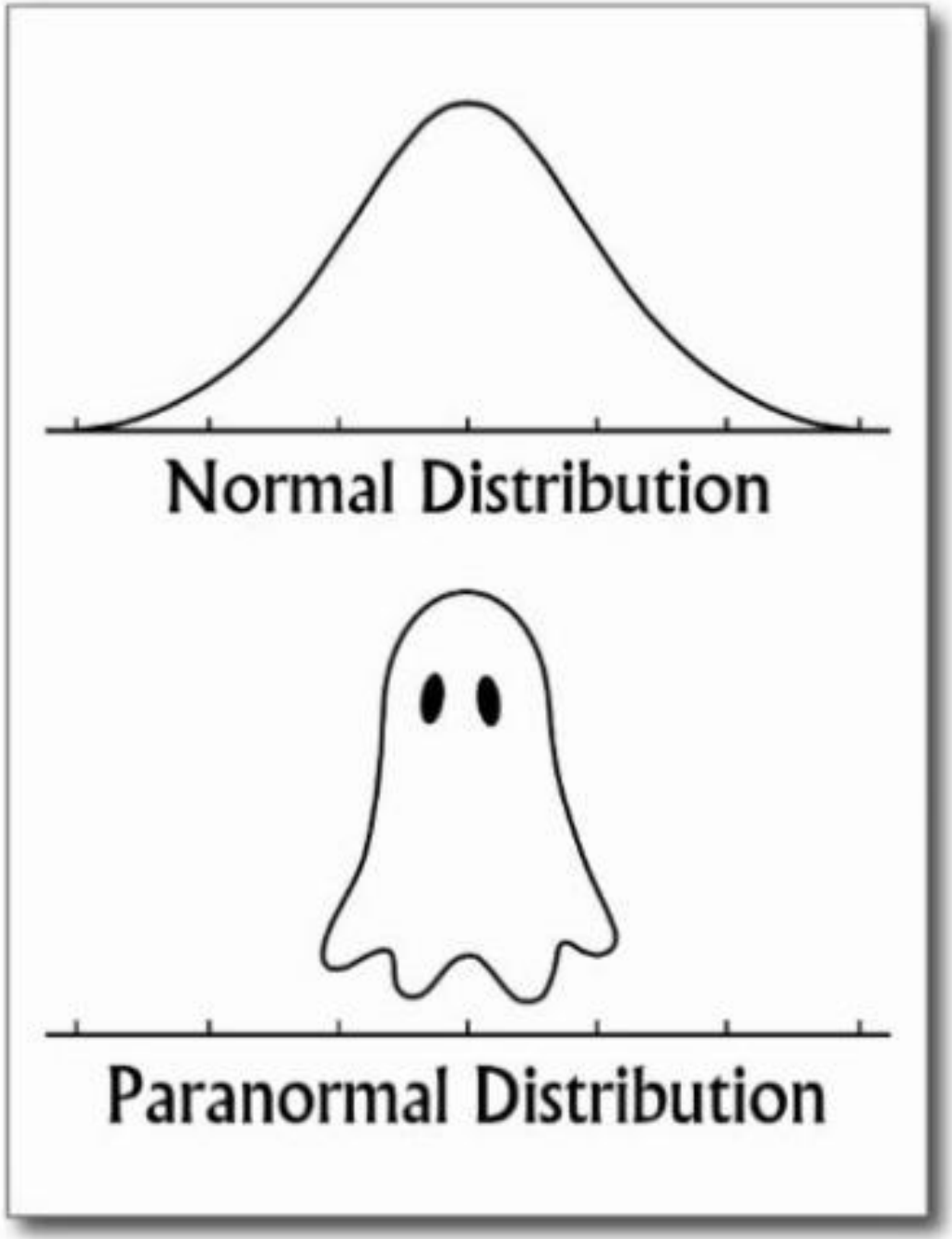
String character frequency histogram per sample

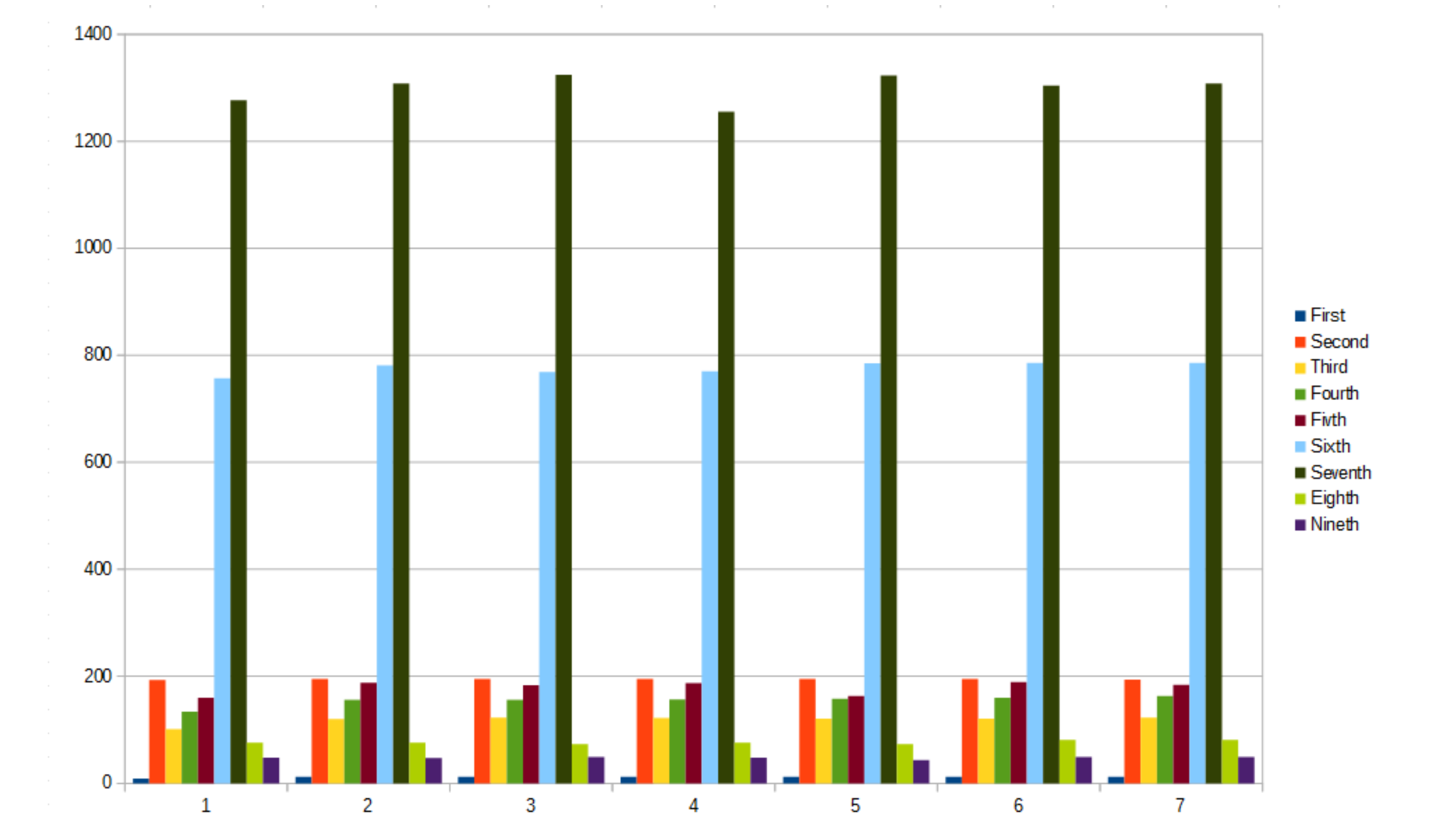
Bucketsize of 0·01

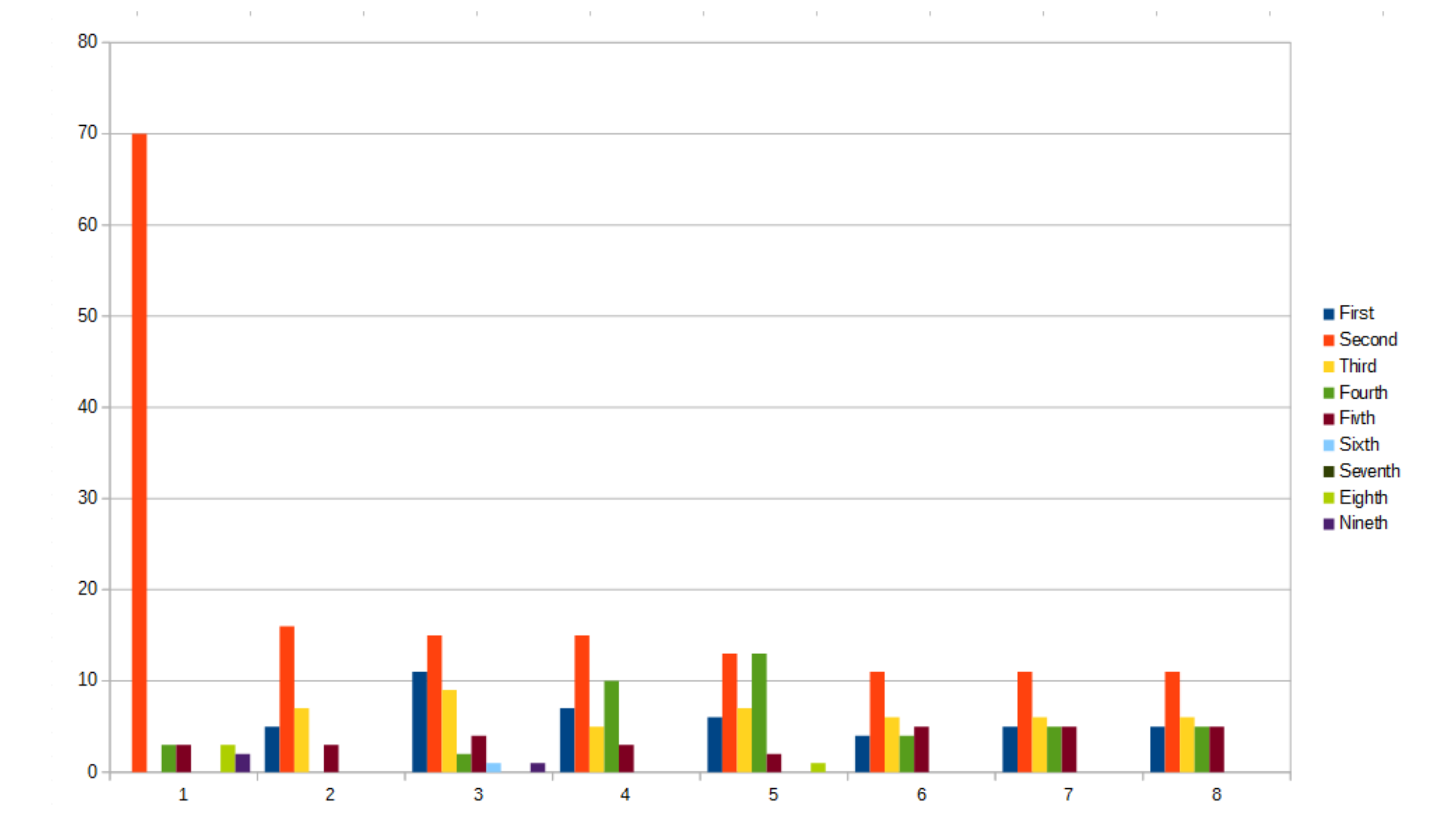
Count of strings per bucket

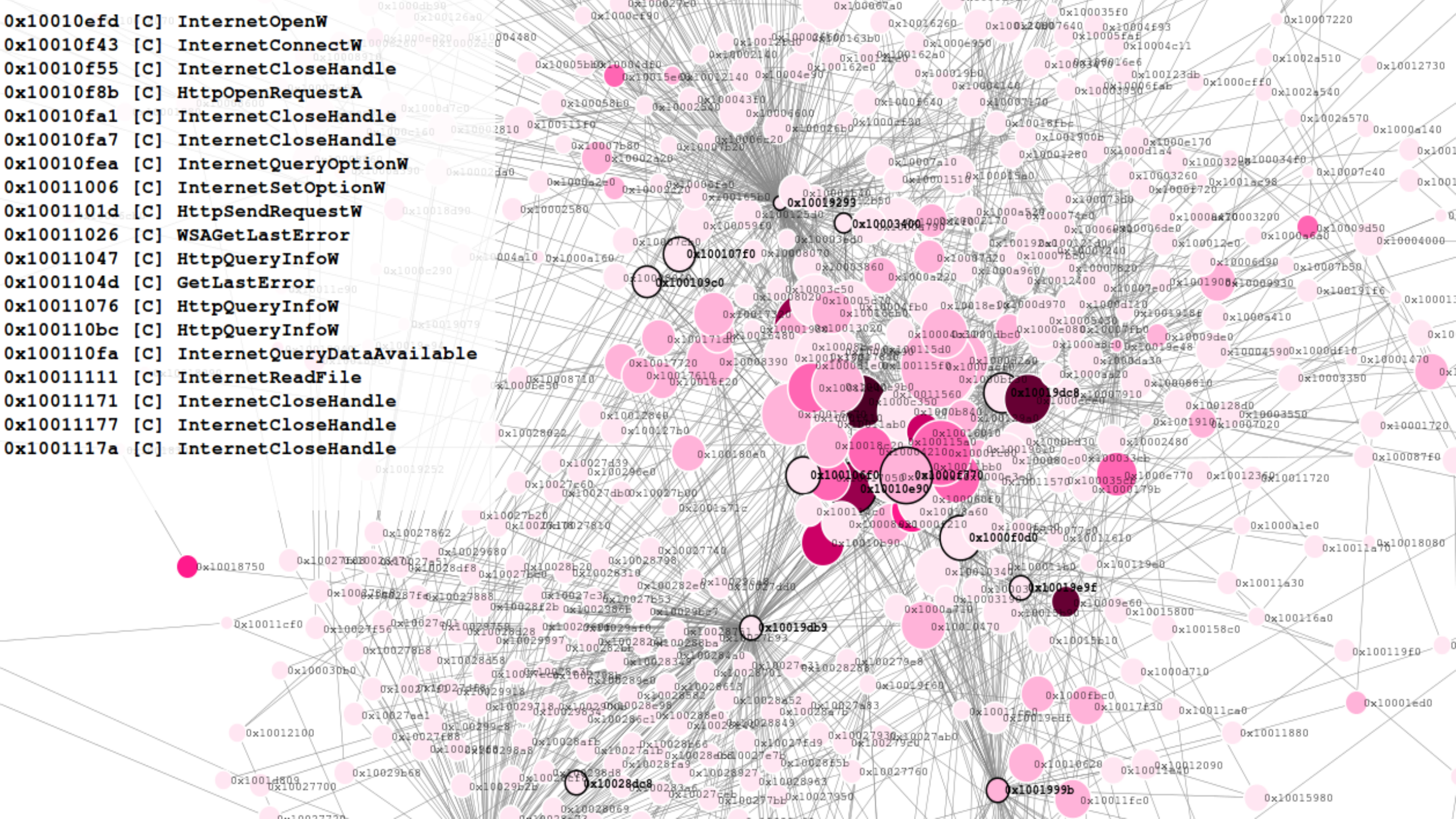
0·04 is a reasonable edge

Resilient to little changes




1400
1200
1000
800
600
400
200
0
1
2
3
4
5
6
7
First
Second
Third
Fourth
Fivth
Sixth
Seventh
Eighth
Nineth


80
70
60
50
40
30
20
10
0
1
2
3
4
5
6
7
8
First
Second
Third
Fourth
Fivth
Sixth
Seventh
Eighth
Nineth


0x10010efd [C] InternetOpenW
0x10010f43 [C] InternetConnectW
0x10010f55 [C] InternetCloseHandle
0x10010f8b [C] HttpOpenRequestA
0x10010fa1 [C] InternetCloseHandle
0x10010fa7 [C] InternetCloseHandle
0x10010fea [C] InternetQueryOptionW
0x10011006 [C] InternetSetOptionW
0x1001101d [C] HttpSendRequestW
0x10011026 [C] WSAGetLastError
0x10011047 [C] HttpQueryInfoW
0x1001104d [C] GetLastError
0x10011076 [C] HttpQueryInfoW
0x100110bc [C] HttpQueryInfoW
0x100110fa [C] InternetQueryDataAvailable
0x10011111 [C] InternetReadFile
0x10011171 [C] InternetCloseHandle
0x10011177 [C] InternetCloseHandle
0x1001117a [C] InternetCloseHandle

Corner Cases and Issues

C++

VB/.NET

Delphi xD

Other exotic compilers

Large binaries

Loops

Inner programming logic


WHY?

Help in static analysis

Borderline foolproof packer detection

Persisting of analysis results

(Unintentional) disassembly framework bug report factory

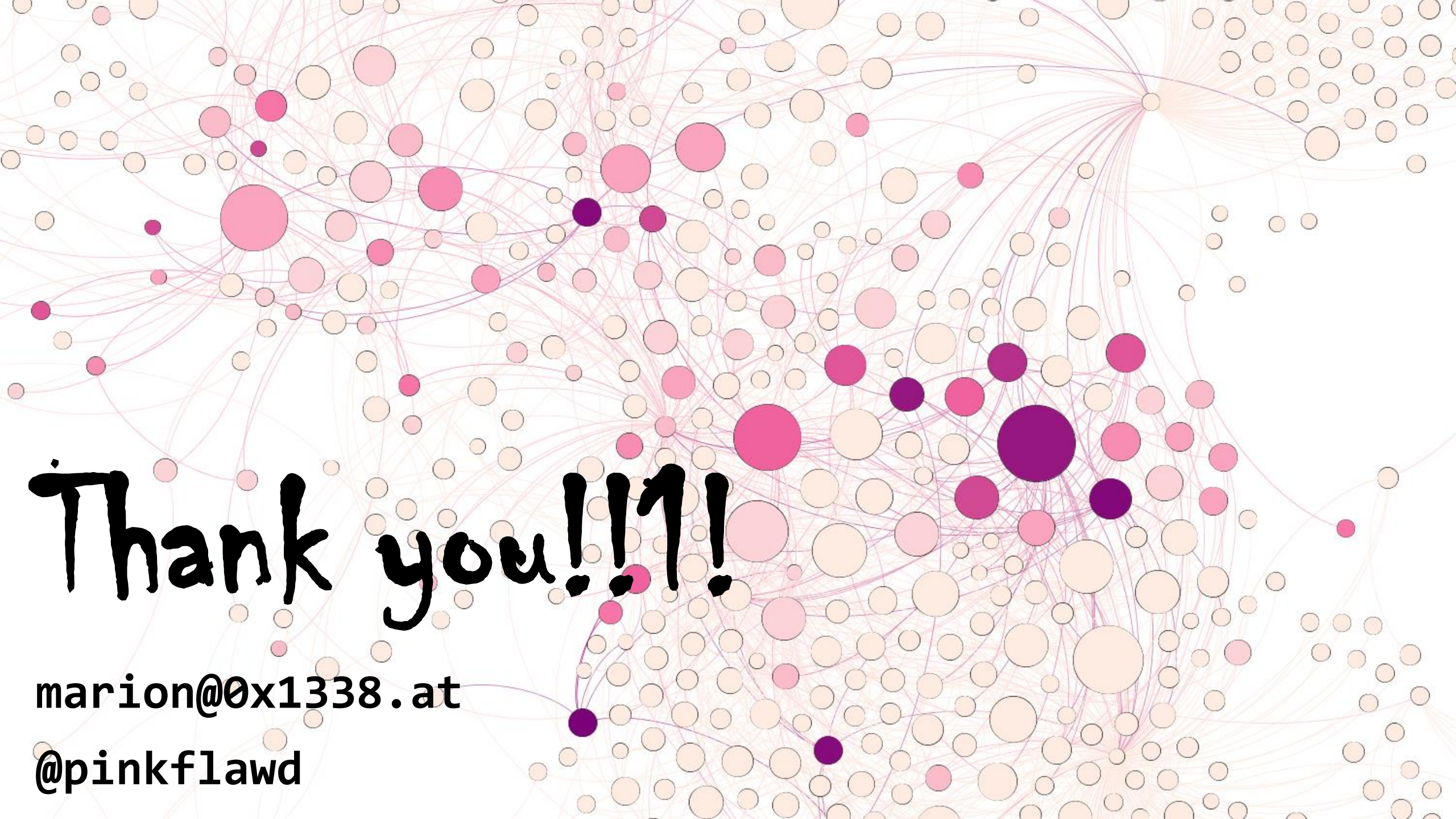
Marketing will faint, I swear

Scales

Open source

Lightweight

Parse once, analyse forevaaa



Thank you!!1!

marion@0x1338.at

@pinkflawd