
50 Shades of
VisualStudio

Disclaimer

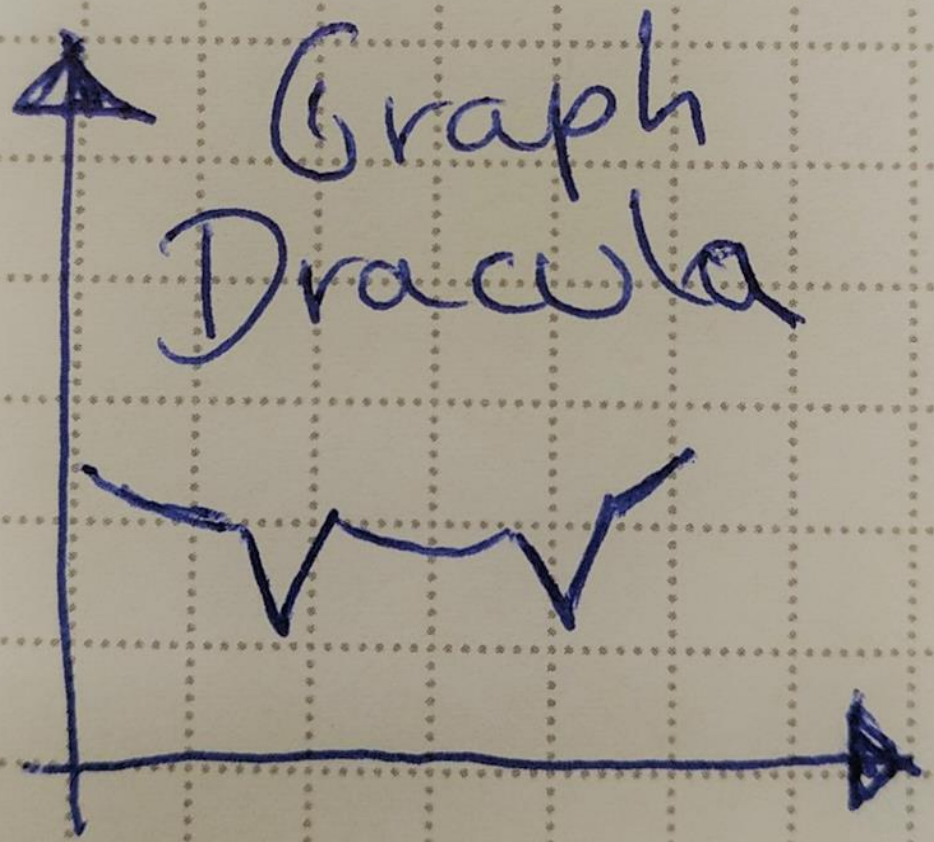
I don’t speak for my employer(duh!). All the opinions and information presented here are my responsibility.

IMPORTANT: No, I am *not* part of the Intel Security Group (McAfee)

Marion Marschalek

marion@0x1338.at

@pinkflawd



Moarrrr IoC

We can‘t find the needle in the haystack,
give us more hay!

Threat Detection: 50 Shades of Hay

File hashes	System behavior	Known-bad
File fragments	Network patterns	Non known-good
File behavior	Abnormal system behavior	Known-bad origin
File properties	Abnormal network patterns	Non known-good origin

Threat detection metrics heavily **build on known fragments**, while aiming to **find the largely unknown**.

DATA

File Geometry

Graph Geometry

API & Behavior Indicators

String Constant Evaluation

	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO	AP	AQ	AR	AS	AT	AU	AV	AW	AX
1	functionstotal	refslocal	refsglobalva	refsunknown	apitotal	apimisses	stringsreferenced	stringsdangling	stringsnoref	ratiofunc	ratioapi	ratiostring	getprocaddress	memallocation	createthread	cshortestpath	callbackcount	cbaveragesize	cblargestsize	stringsrefhisto
2	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
3	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
4	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
5	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
6	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
7	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
8	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
9	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
10	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
11	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
12	1513	3082	3	1	384	1	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.6774677121771218	14	5	4	2	4	118	219	2-4-7-13-29-5-0-3-29
13	1436	2921	3	1	371	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.6084735576923077	12	5	4	2	4	118	219	2-4-7-7-27-0-0-3-29
14	1445	2936	3	1	374	7	[illegible]	[illegible]	2537	[illegible]	[illegible]	0.6435584291187739	12	5	4	2	4	118	219	3-4-7-10-27-0-0-3-29
15	1511	3095	3	1	376	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	0.6534352022058824	12	5	4	2	4	118	219	3-4-7-12-27-4-0-3-29
16	4255	20499	21	30	690	2	[illegible]	21	[illegible]	[illegible]	[illegible]	3.2876536885245904	63	5	2	5	2	119	185	13-233-274-464-276-1381-1895-265-19
17	4255	20499	21	30	690	2	[illegible]	21	[illegible]	[illegible]	[illegible]	3.2876536885245904	63	5	2	5	2	119	185	13-233-274-464-276-1381-1895-265-19
18	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
19	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
20	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
21	3639	6698	3	1	873	0	[illegible]	0	6995	8.095013525056947	[illegible]	0.2602683656036446	66	3	3	2	2	119	173	2-2-5-11-25-1-0-4-46
22	3639	6698	3	1	873	0	[illegible]	0	[illegible]	[illegible]	[illegible]	[illegible]	66	3	3	2	2	119	173	2-2-5-11-25-1-0-4-46
23	295	859	6	1	305	2	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	15	16	1	5	1	161	161	3-0-3-8-13-0-1-3-2
24	247	720	6	1	296	0	38	0	1245	8.614676339285714	10.323660714285714	[illegible]	15	21	1	5	1	161	161	3-1-3-8-13-0-1-3-2
25	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
26	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
27	3950	17779	15	30	627	1	2973	2	27944	[illegible]	0.471729343220339	[illegible]	63	5	1	5	1	125	125	12-195-122-175-177-784-1324-76-50
28	3572	15520	15	30	589	0	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	35	6	1	7	1	101	101	12-194-123-163-184-786-1308-81-49
29	3573	15503	15	30	591	0	[illegible]	14	[illegible]	[illegible]	[illegible]	[illegible]	35	6	1	5	1	101	101	12-195-120-156-188-781-1308-76-47
30	3700	16162	15	31	536	2	[illegible]	1	[illegible]	[illegible]	[illegible]	[illegible]	35	6	1	6	1	101	101	12-195-121-158-163-785-1323-73-43
31	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
32	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
33	3551	15594	15	29	526	0	[illegible]	52	17439	[illegible]	0.7055932348901099	3.741253863324176	35	6	1	5	1	101	101	9-193-101-134-160-757-1277-76-48
34	3573	15686	15	30	591	24	[illegible]	16	18491	[illegible]	[illegible]	[illegible]	[illegible]	6	[illegible]	5	1	101	101	12-195-121-160-189-786-1304-81-49
35	434	1247	3	1	318	0	[illegible]	[illegible]	991	[illegible]	[illegible]	[illegible]	[illegible]	5	[illegible]	6	1	101	101	2-1-2-10-5-0-0-3-9
36	430	1244	3	1	317	0	[illegible]	[illegible]	1003	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	6	1	101	101	2-1-2-10-7-0-0-3-9
37	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
38	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
39	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
40	234	829	0	0	291	0	31	0	583	[illegible]	[illegible]	[illegible]	3	1	3	2	2	69	92	4-11-6-4-5-0-0-0-0
41	234	829	0	0	291	0	[illegible]	[illegible]	568	[illegible]	[illegible]	[illegible]	3	1	3	2	2	78	92	5-11-6-5-5-0-0-0-0
42	234	829	0	0	291	0	[illegible]	[illegible]	568	[illegible]	[illegible]	[illegible]	3	1	3	2	2	78	92	5-11-6-5-5-0-0-0-0
43	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
44	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
45	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
46	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
47	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
48	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
49	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
50	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
51	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
52	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
53	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
54	228	922	0	0	121	0	34	0	457	9.474734042553191	5.028257978723404	1.4128989361702127	0	1	1	2	1	57	57	4-8-5-12-2-1-0-1-0

Function call graphs

Function cross references within code section

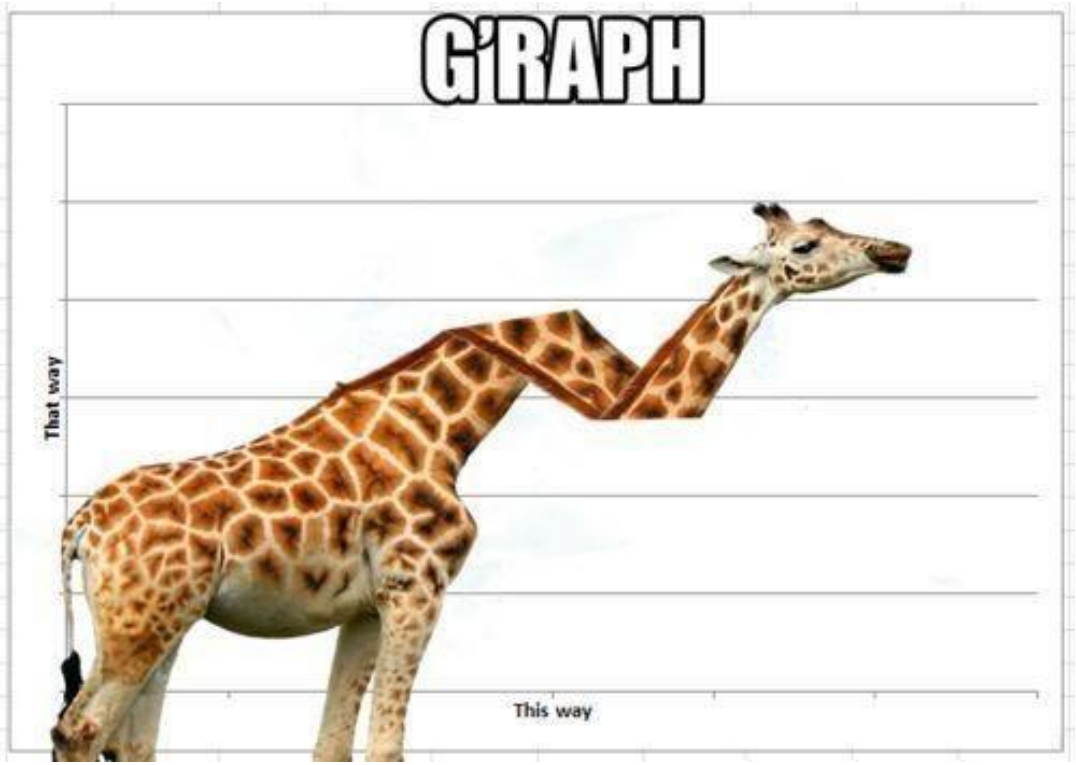
References to function offsets

Outside executable section(s)

Nodes: functions

=> Offset, size, calling convention

Edges: calls, indirect calls



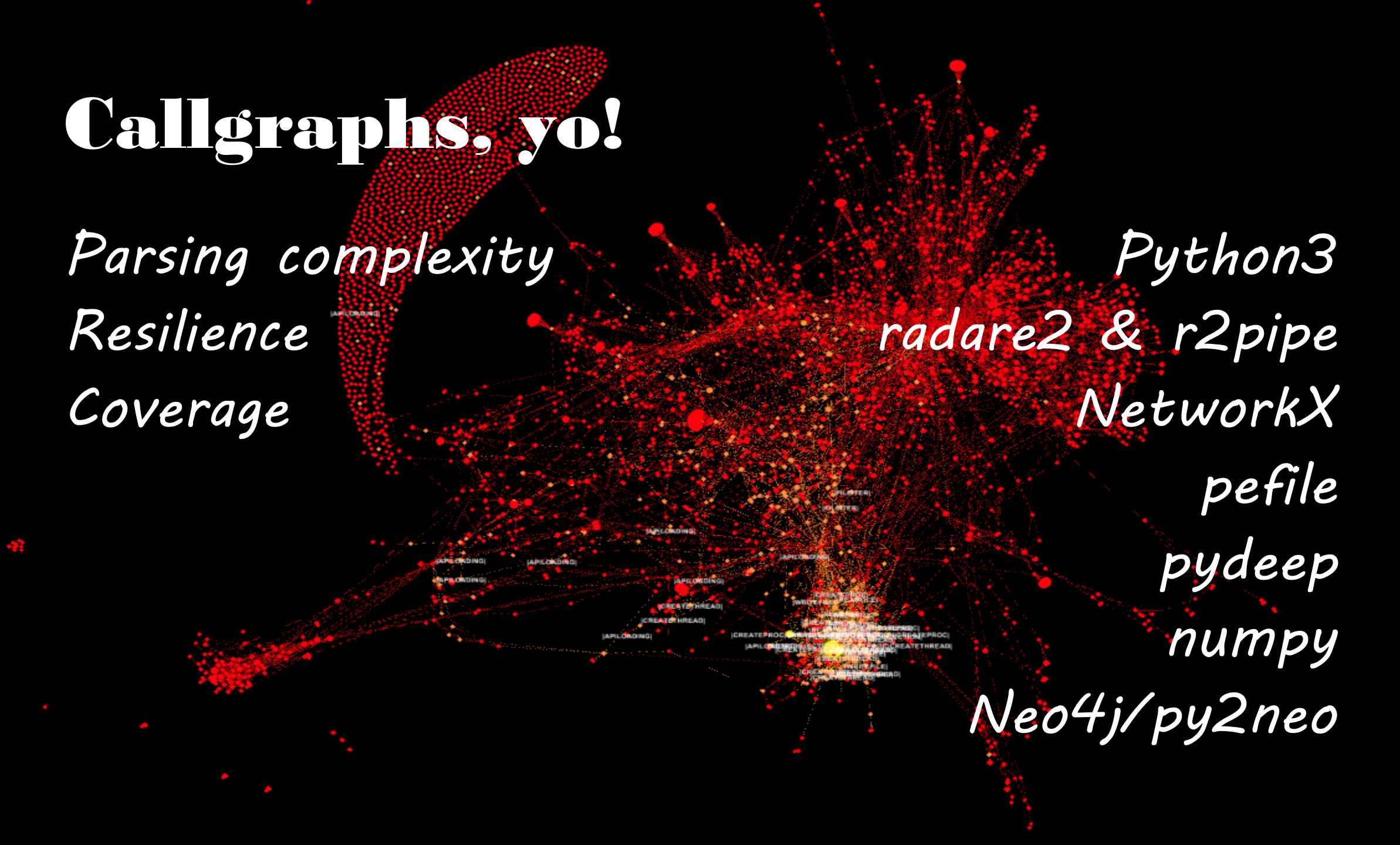
r2graphity

```
    0x00403e99      8bc8            mov ecx, eax
    0x00403e9b      81e1ff000000    and ecx, 0xff
    0x00403ea1      890d14b04000    mov dword [0x40b014], ecx   ; [0x40b014:4]=0
    0x00403ea7      c1e108          shl ecx, 8
    0x00403eaa      03ca            add ecx, edx
    0x00403eac      890d10b04000    mov dword [0x40b010], ecx   ; [0x40b010:4]=0
    0x00403eb2      c1e810          shr eax, 0x10
    0x00403eb5      a30cb04000      mov dword [0x40b00c], eax   ; [0x40b00c:4]=0
    0x00403eba      33f6            xor esi, esi
    0x00403ebc      56              push esi
    0x00403ebd      e8fb190000      call sub.KERNEL32.DLL_HeapCreate_8bd ; heaphandle HeapCreate(HeapAllocationControl flOptions, SIZE_T dwInitialSi
    0x00403ec2      59              pop ecx
    0x00403ec3      85c0            test eax, eax
,=< 0x00403ec5      7508            jne 0x403ecf
|   0x00403ec7      6a1c            push 0x1c                   ; 28
|   0x00403ec9      e8b0000000      call sub.KERNEL32.DLL_ExitProcess_f7e ; void ExitProcess(UINT uExitCode)
|   0x00403ece      59              pop ecx
|      ; JMP XREF from 0x00403ec5 (entry0)
`-> 0x00403ecf      8975fc          mov dword [local_4h], esi
    0x00403ed2      e83b180000      call sub.KERNEL32.DLL_GetStartupInfoA_712 ; void GetStartupInfoA(void)
    0x00403ed7      ff15a0904000    call dword [sym.imp.KERNEL32.DLL_GetCommandLineA] ; 0x4090a0 ; "L\x99"
    0x00403edd      a304c64000      mov dword [0x40c604], eax   ; [0x40c604:4]=0
    0x00403ee2      e8f9160000      call sub.KERNEL32.DLL_GetEnvironmentStringsW_5e0 ; lpwstr GetEnvironmentStringsW(void)
    0x00403ee7      a34cb04000      mov dword [0x40b04c], eax   ; [0x40b04c:4]=0
    0x00403eec      e8a2140000      call sub.KERNEL32.DLL_GetModuleFileNameA_393 ; dword GetModuleFileNameA(HMODULE hModule, LPSTR lpFilename, DWORD
    0x00403ef1      e8e4130000      call fcn.004052da
    0x00403ef6      e8dcfcffff      call fcn.00403bd7
    0x00403efb      8975d0          mov dword [local_30h], esi
    0x00403efe      8d45a4          lea eax, [local_5ch]
    0x00403f01      50              push eax
    0x00403f02      ff159c904000    call dword [sym.imp.KERNEL32.DLL_GetStartupInfoA] ; 0x40909c ; ":\x99"
    0x00403f08      e875130000      call fcn.00405282
    0x00403f0d      89459c          mov dword [local_64h], eax
    0x00403f10      f645d001        test byte [local_30h], 1    ; [0x1:1]=255 ; 1
,=< 0x00403f14      7406            je 0x403f1c
|   0x00403f16      0fb745d4        movzx eax, word [local_2ch]
,==< 0x00403f1a      eb03            jmp 0x403f1f
||     ; JMP XREF from 0x00403f14 (entry0)
|`-> 0x00403f1c      6a0a            push 0xa                    ; 10
|   0x00403f1e      58              pop eax
|      ; JMP XREF from 0x00403f1a (entry0)
`--> 0x00403f1f      50              push eax
    0x00403f20      ff759c          push dword [local_64h]
```

Yes I still use radare2

Static Analysis is ~~King~~ Princess

Callgraphs, yo!



Parsing complexity
Resilience
Coverage

Python3
radare2 & r2pipe
NetworkX
pefile
pydeep
numpy
Neo4j/py2neo

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

Corner Cases and Issues

C++

VB/.NET

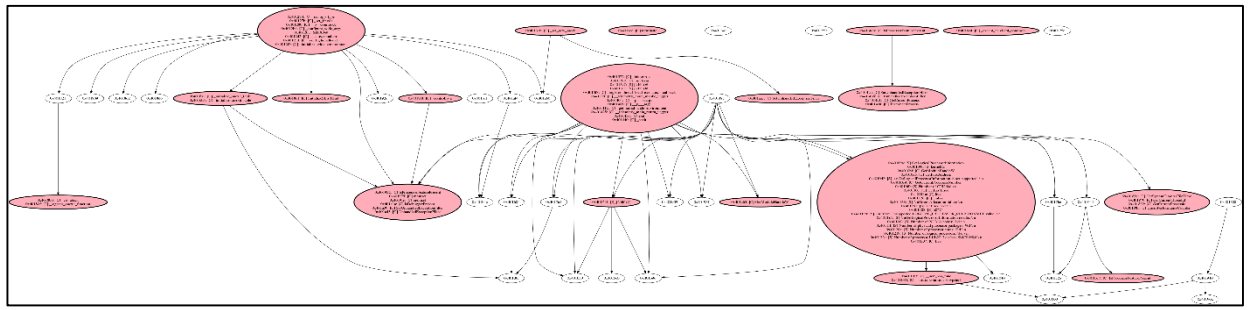
Delphi xD

Other exotic compilers

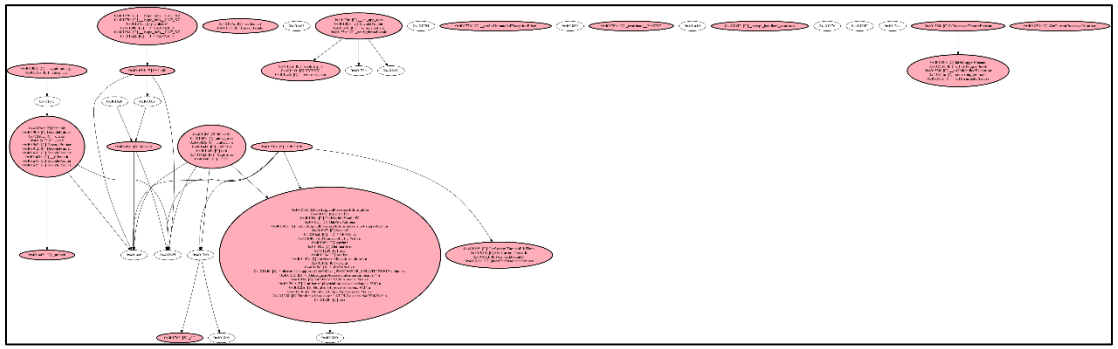
Large binaries

Loops

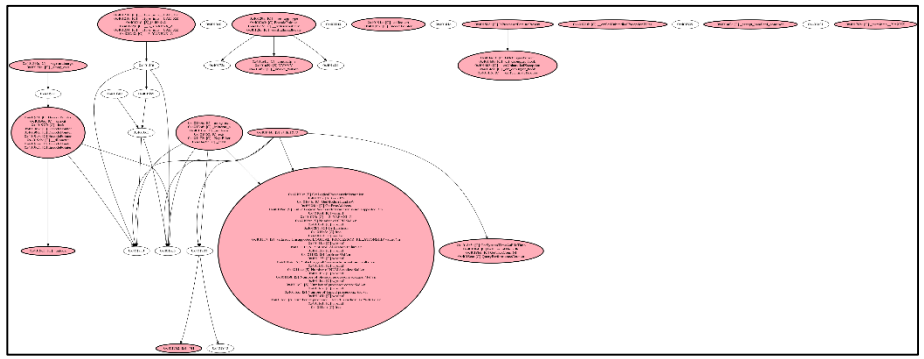
Inner programming logic



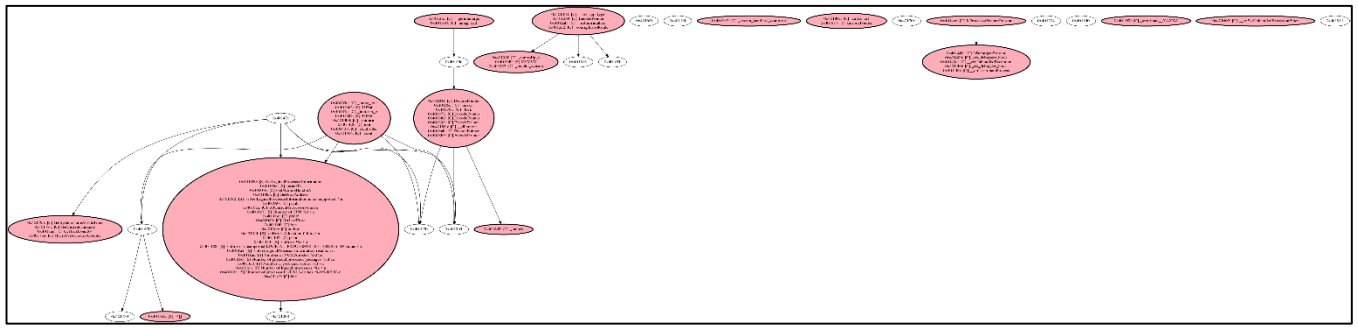
Default



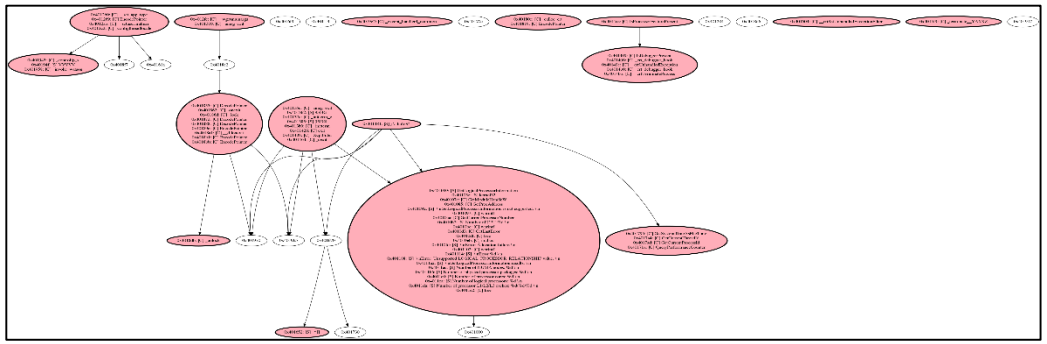
FullyOptimized



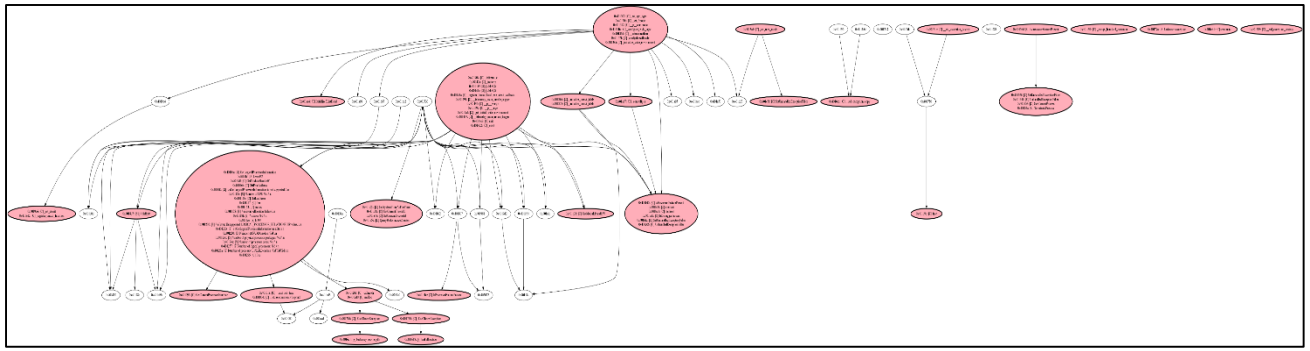
SizeOptimized



vc110



vc120



vc150Dbg

Compilers and Optimization

Scope: local, regional, global, interprocedural

Elimination of redundant expressions

Reuse of previously computed values

Helping instruction level parallelism

Loop unrolling

Basic block order optimization

Function order optimization

Data flow optimization

Inline substitution

„Engineering A Compiler, 2nd Edition“ by Cooper & Torczon

Optimization can do terrible things to assembly

```
mov     [ebp+var_14], eax
mov     [ebp+var_10], edx
call    ds:_Query_perf_counter
mov     [ebp+var_1C], eax
mov     [ebp+var_18], edx
mov     eax, [ebp+var_10]
push    eax
mov     ecx, [ebp+var_14]
push    ecx
mov     edx, [ebp+var_18]
push    edx
mov     eax, [ebp+var_1C]
push    eax
call    __alldiv
push    0
push    3B9ACA00h
push    edx
```

=

```
sub_4010CF proc near

var_8= dword ptr -8
var_4= dword ptr -4

push    ecx
push    ecx
push    ebx
push    ebp
push    esi
push    edi
mov     [esp+18h+var_4], ecx
call    ds:_Query_perf_frequency
mov     esi, edx
call    ds:_Query_perf_counter
push    esi
push    edi
push    edx
push    eax
call    __alldvrm
push    0
push    3B9ACA00h
push    ecx
mov     [esp+28h+var_8], eax
mov     ebp, edx
call    __allmul
push    esi
push    edi
push    edx
push    eax
call    __alldiv
push    0
push    3B9ACA00h
push    ebp
push    [esp+24h+var_8]
mov     esi, eax
mov     edi, edx
call    __allmul
add     esi, eax
mov     eax, [esp+18h+var_4]
adc     edi, edx
mov     [eax+4], edi
pop     edi
mov     [eax], esi
pop     esi
pop     ebp
pop     ebx
pop     ecx
pop     ecx
retn
sub_4010CF endp
```

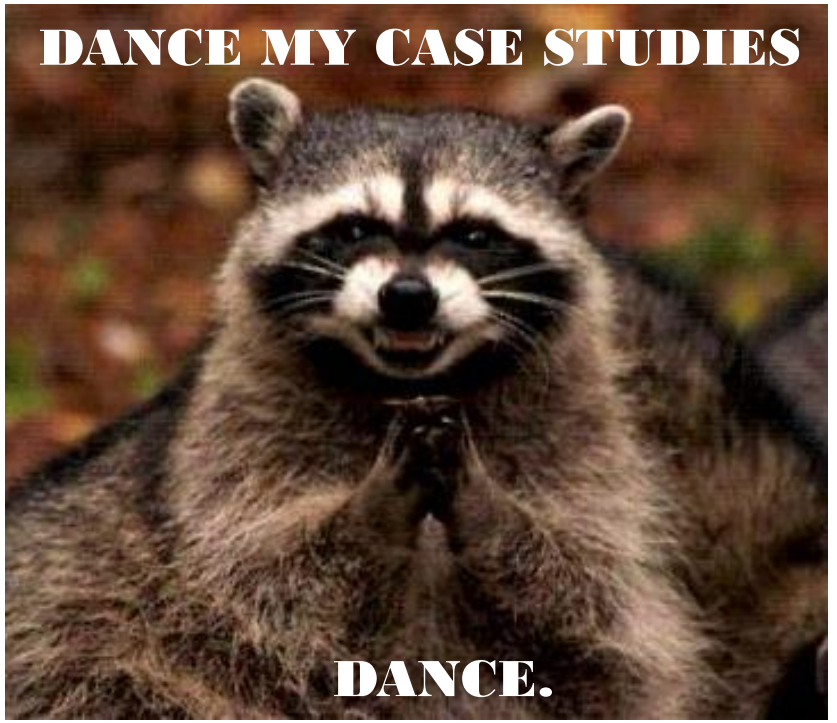
Win32Window - A simple Win32 window application

randomNG - A random number generator in C

LongPrimeSieve - A long prime number generator in C++

Visual Studio 2015

Optimization	/Od /O1 /O2
Function Expansion	/Ob1 /Ob2
Intrinsic Functions	/Oi
Size or Speed	/Os /Ot
Omit Frame Pointer	/Oy
Fiber-safe TLS	/GT
Whole Program Opt.	/GL
EVERYthing	all



What to look for?

Node and edge counts

Jumps and calls

Filesize

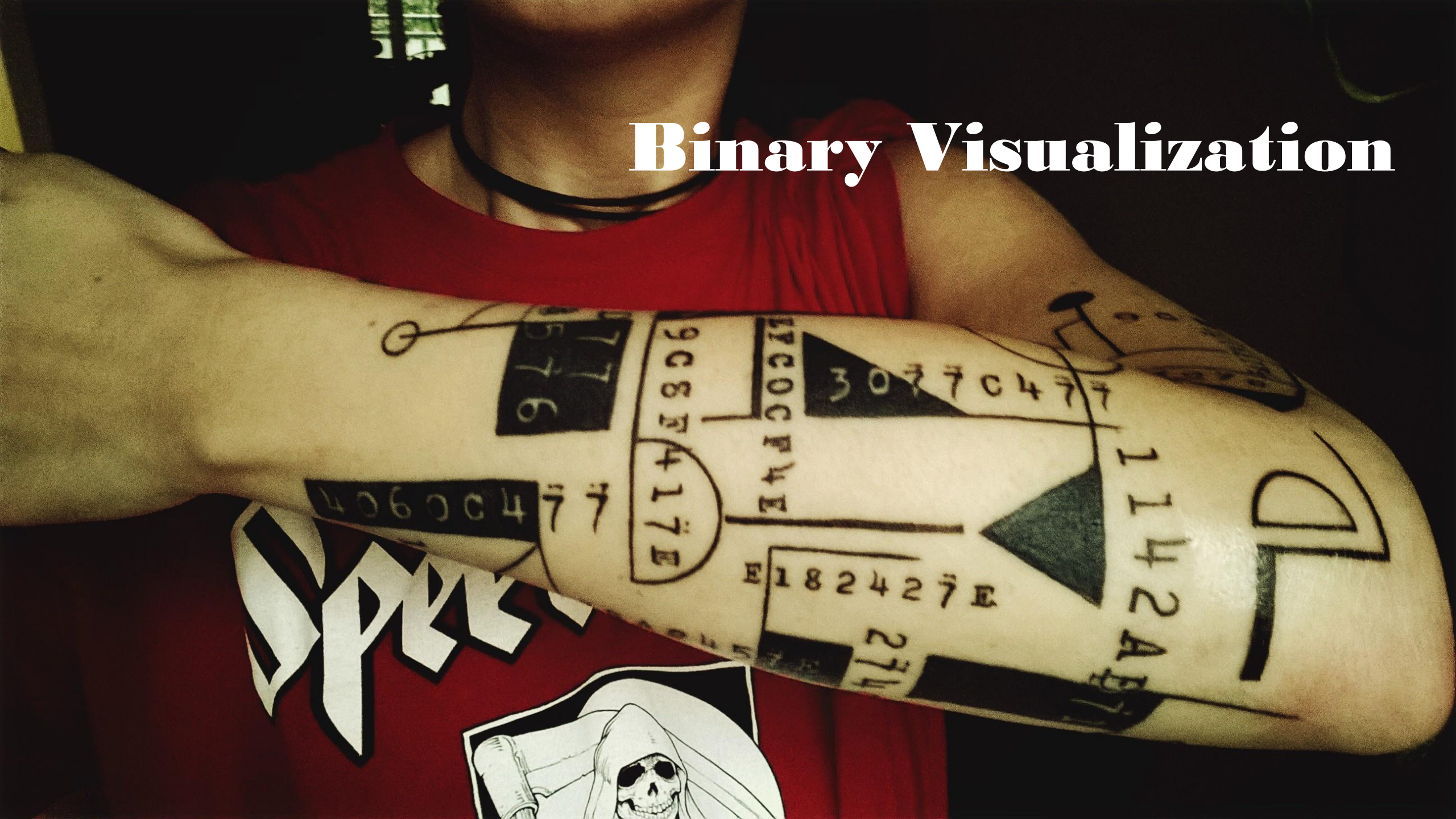
Instruction count and variance

Data references

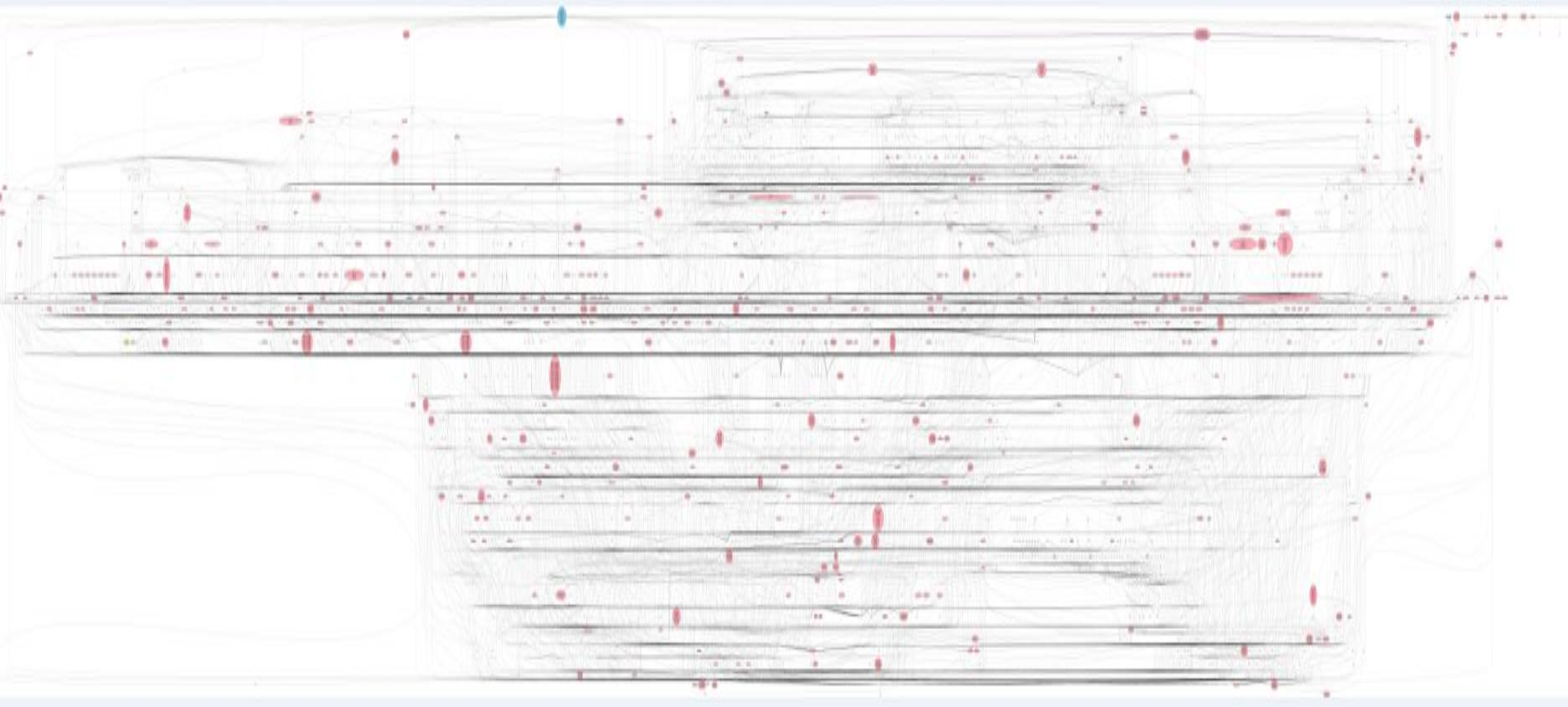
Memory references

… and how to look at it?

Binary Visualization



„Useful“ ain‘t easy



Visualization

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*

Graph visualization

Dot & directed graphs

Heatmaps

Histograms and diagrams

Distributions

The Attributes

Mnemonic types total

Mnemonic variance per function

Presence of instruction families

Graph node and edge count

API call count

Data references

Ratios abstracted by code section size

Variance & standard deviation of attributes per case study

Data

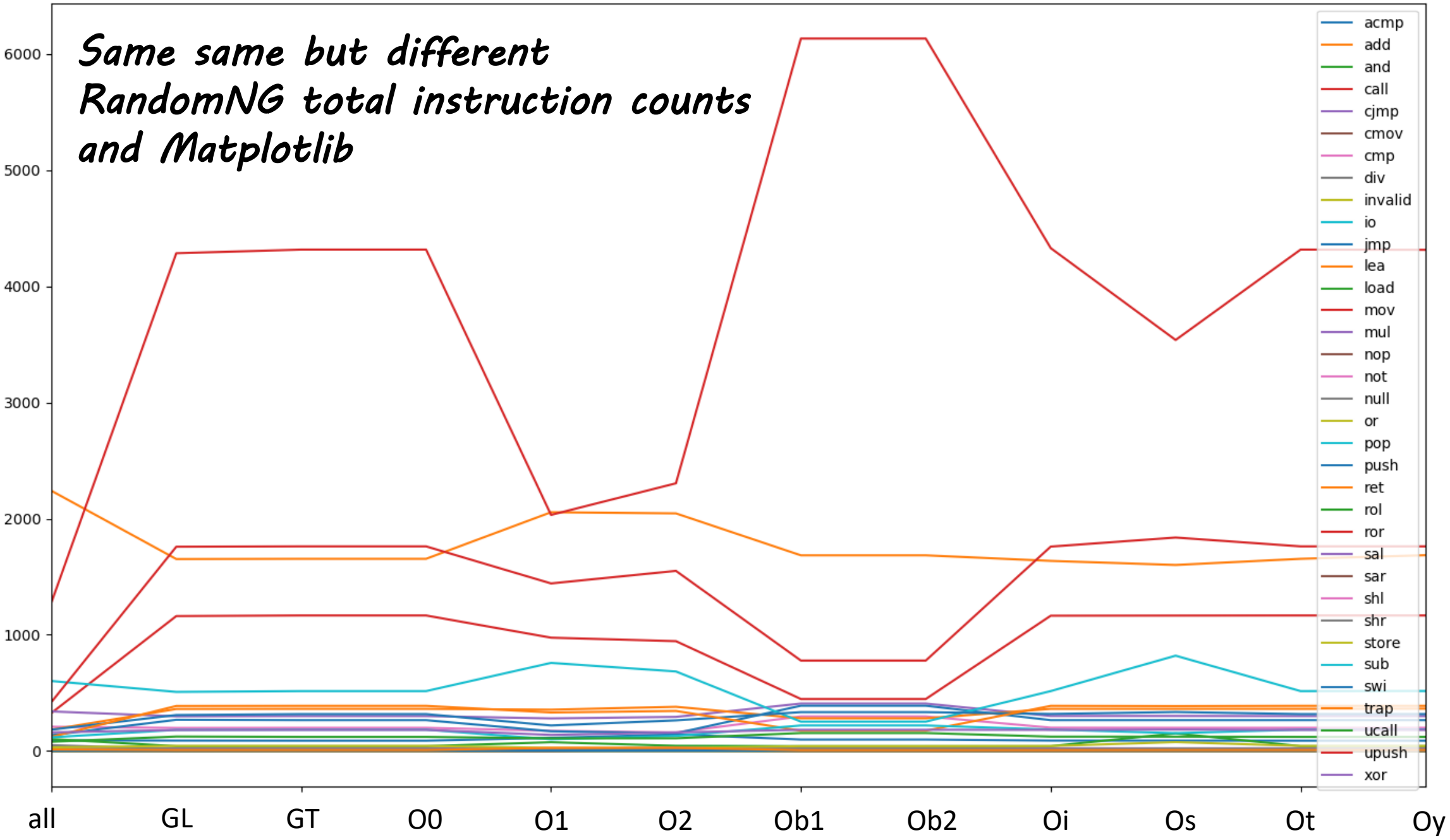
filename	acmp	add	and	call	cjmp	cmov	cmp	div	jmp	lea	load	mov	mul	nop	not	null	or	pop	push	ret	rol	ror	sar	shl	shr	store	sub	swi	trap	ucall	upush	xor
LongPrimeSieve_all.exe	75	1778	34	181	268	8	138	9	56	62	1	720	23	0	5	94	45	164	103	72	0	8	4	2	13	4	96	2	5	75	322	88
LongPrimeSieve_GL.exe	59	1444	31	228	269	3	154	9	91	90	14	1414	17	0	5	91	48	130	198	93	0	9	4	4	13	21	106	2	5	101	438	86
LongPrimeSieve_GT.exe	61	1930	88	231	366	3	171	9	91	91	17	1451	32	1	5	123	49	174	201	94	1	10	4	3	13	28	126	2	5	100	498	94
LongPrimeSieve_O0.exe	60	1927	88	231	364	3	171	9	91	91	17	1453	33	1	5	124	52	177	201	94	1	10	4	6	13	24	121	2	5	100	499	94
LongPrimeSieve_O1.exe	85	1177	34	214	284	6	139	9	56	88	4	813	21	0	5	99	44	286	148	96	0	9	4	2	13	22	98	2	5	94	447	91
LongPrimeSieve_O2.exe	96	734	29	217	290	6	131	12	49	94	6	880	23	4	5	100	47	168	183	99	0	11	4	12	12	22	130	2	5	98	423	90
LongPrimeSieve_Ob1.exe	62	1540	69	197	325	3	172	9	90	67	14	1517	23	1	5	109	48	127	195	78	1	8	4	4	13	21	114	2	7	102	379	92
LongPrimeSieve_Ob2.exe	62	1540	69	197	325	3	172	9	90	67	14	1517	23	1	5	109	48	127	195	78	1	8	4	4	13	21	114	2	7	102	379	92
LongPrimeSieve_Oi.exe	59	1387	29	229	267	3	153	9	90	91	16	1443	17	0	5	88	45	125	198	94	0	8	4	3	13	20	102	2	4	100	434	85
LongPrimeSieve_Os.exe	58	1690	53	231	266	3	152	9	90	91	14	1181	17	0	5	88	48	262	198	93	0	9	4	3	13	20	95	2	4	100	448	87
LongPrimeSieve_Ot.exe	60	1929	89	231	366	3	171	9	91	91	17	1453	32	1	6	123	49	172	201	94	1	10	4	3	13	27	121	5	5	100	498	94
LongPrimeSieve_Ox.exe	96	731	29	218	290	6	130	12	49	94	6	879	23	4	5	100	46	167	182	100	0	10	4	12	12	23	127	2	4	98	423	90
LongPrimeSieve_Oy.exe	59	1750	29	230	266	3	152	9	90	93	14	1429	17	0	5	88	45	132	198	93	0	10	4	3	13	21	102	2	4	100	438	85

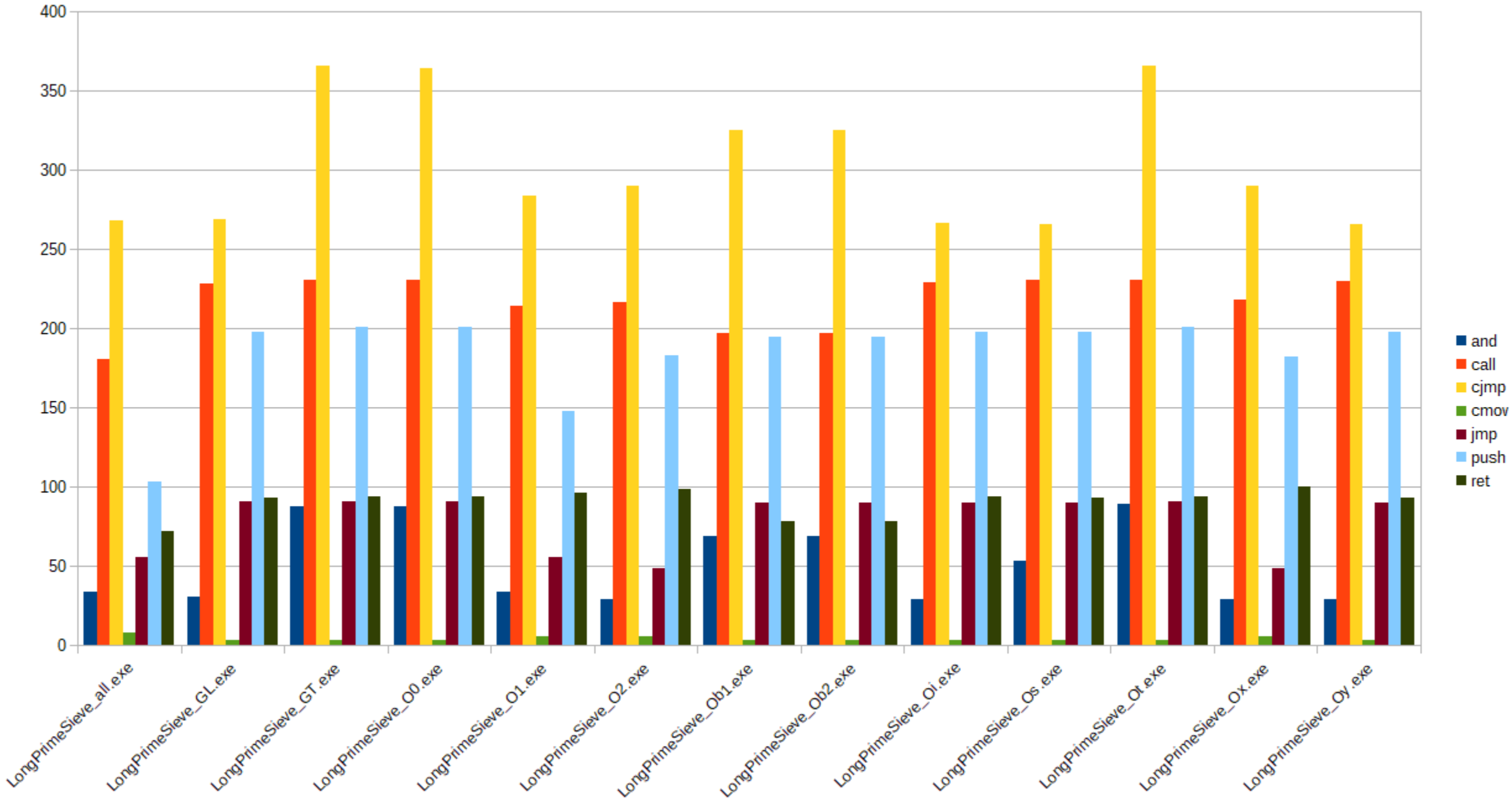
Heatmaps

filename	acmp	add	and	call	cjmp	cmov	cmp	div	jmp	lea	load	mov	mul	nop	not	null	or	pop	push	ret	rol	ror	sal	sar	shl	shr	store	sub	swi	trap	ucall	upush	xor
randomNG_all.exe	91	2240	102	326	340	18	208	30	136	183	0	1277	34	1	4	50	30	602	183	121	0	2	1	1	6	10	1	117	2	12	79	420	160
randomNG_GL.exe	89	1652	41	1161	298	3	198	17	268	361	0	4285	21	0	4	16	43	508	309	387	0	2	1	0	10	13	1	181	2	7	124	1758	180
randomNG_GT.exe	88	1654	41	1166	300	3	199	18	265	362	0	4316	23	0	4	17	43	515	317	388	0	2	0	1	11	13	1	183	3	7	121	1761	181
randomNG_O0.exe	88	1654	41	1166	300	3	199	18	265	362	0	4316	23	0	4	17	43	515	317	388	0	2	0	1	11	13	1	183	3	7	121	1761	181
randomNG_O1.exe	112	2055	75	975	280	20	173	14	169	355	0	2032	17	0	7	13	26	758	219	331	0	3	0	0	8	10	2	100	3	26	108	1442	138
randomNG_O2.exe	142	2045	44	945	293	22	160	5	152	380	0	2304	26	16	7	6	24	684	263	344	1	2	0	8	8	19	1	129	3	27	109	1550	159
randomNG_Ob1.exe	98	1684	41	447	408	3	297	27	389	281	0	6133	23	0	4	24	41	252	334	174	0	2	0	0	12	14	1	219	3	6	154	778	182
randomNG_Ob2.exe	98	1684	41	447	408	3	297	27	389	281	0	6133	23	0	4	24	41	252	334	174	0	2	0	0	12	14	1	219	3	6	154	778	182
randomNG_Oi.exe	89	1636	41	1164	301	3	200	17	265	362	0	4329	23	0	4	14	42	515	318	388	0	2	0	0	11	13	1	184	3	6	123	1759	181
randomNG_Os.exe	89	1601	151	1165	301	3	199	17	265	362	1	3538	23	0	4	21	76	820	335	386	0	2	0	0	11	14	1	153	3	7	123	1837	184
randomNG_Ot.exe	88	1654	41	1166	300	3	199	18	265	362	0	4316	23	0	4	17	43	515	317	388	0	2	0	1	11	13	1	183	3	7	121	1761	181
randomNG_Oy.exe	88	1685	41	1166	300	3	199	18	265	364	0	4314	23	0	4	16	43	516	317	388	0	2	0	1	11	13	1	185	3	7	121	1761	181

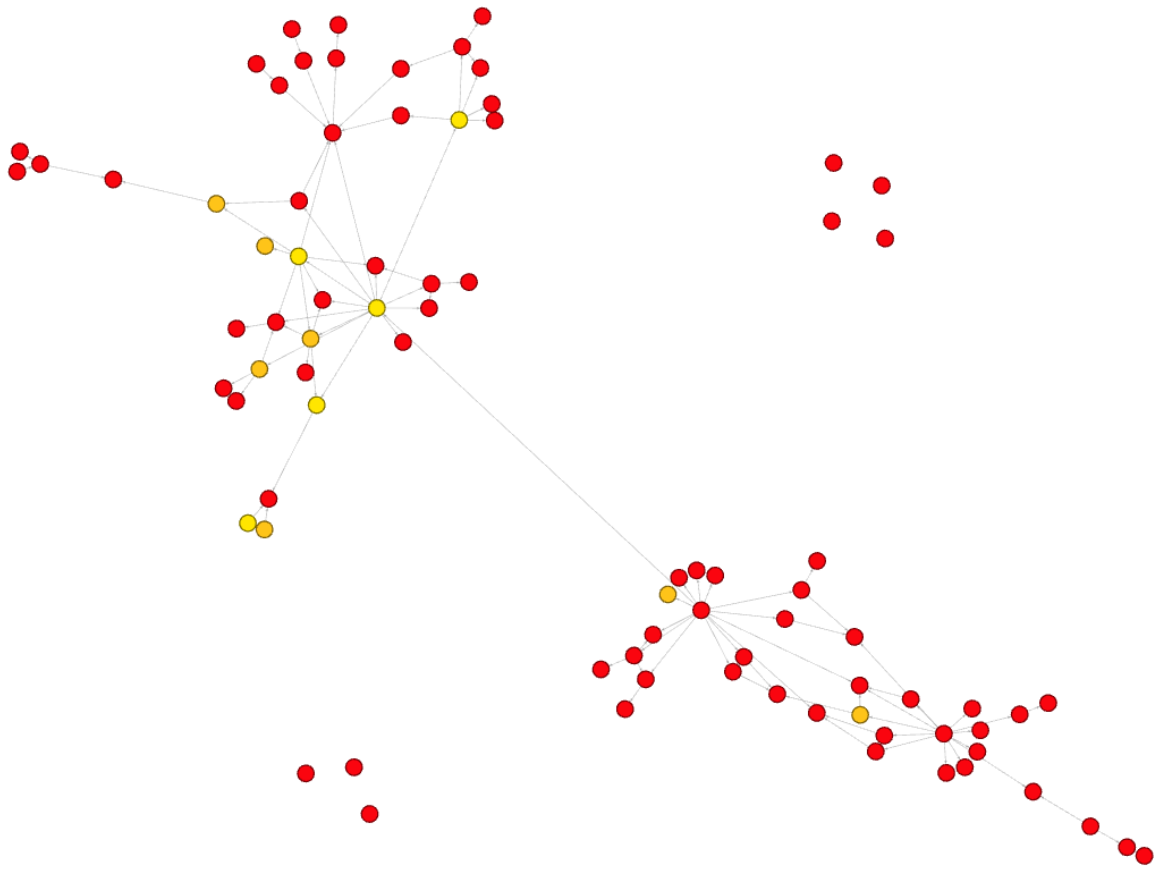
filename	acmp	add	and	call	cjmp	cmov	cmp	div	jmp	lea	load	mov	mul	nop	not	null	or	pop	push	ret	rol	ror	sal	sar	shl	shr	store	sub	swi	trap	ucall	upush	xor
LongPrimeSieve_all.exe	75	1778	34	181	268	8	138	9	56	62	1	720	23	0	5	94	45	164	103	72	0	8	4	2	13	4	96	2	5	75	322	88
LongPrimeSieve_GL.exe	59	1444	31	228	269	3	154	9	91	90	14	1414	17	0	5	91	48	130	198	93	0	9	4	4	13	21	106	2	5	101	438	86
LongPrimeSieve_GT.exe	61	1930	88	231	366	3	171	9	91	91	17	1451	32	1	5	123	49	174	201	94	1	10	4	3	13	28	126	2	5	100	498	94
LongPrimeSieve_O0.exe	60	1927	88	231	364	3	171	9	91	91	17	1453	33	1	5	124	52	177	201	94	1	10	4	6	13	24	121	2	5	100	499	94
LongPrimeSieve_O1.exe	85	1177	34	214	284	6	139	9	56	88	4	813	21	0	5	99	44	286	148	96	0	9	4	2	13	22	98	2	5	94	447	91
LongPrimeSieve_O2.exe	96	734	29	217	290	6	131	12	49	94	6	880	23	4	5	100	47	168	183	99	0	11	4	12	12	22	130	2	5	98	423	90
LongPrimeSieve_Ob1.exe	62	1540	69	197	325	3	172	9	90	67	14	1517	23	1	5	109	48	127	195	78	1	8	4	4	13	21	114	2	7	102	379	92
LongPrimeSieve_Ob2.exe	62	1540	69	197	325	3	172	9	90	67	14	1517	23	1	5	109	48	127	195	78	1	8	4	4	13	21	114	2	7	102	379	92
LongPrimeSieve_Oi.exe	59	1387	29	229	267	3	153	9	90	91	16	1443	17	0	5	88	45	125	198	94	0	8	4	3	13	20	102	2	4	100	434	85
LongPrimeSieve_Os.exe	58	1690	53	231	266	3	152	9	90	91	14	1181	17	0	5	88	48	262	198	93	0	9	4	3	13	20	95	2	4	100	448	87
LongPrimeSieve_Ot.exe	60	1929	89	231	366	3	171	9	91	91	17	1453	32	1	6	123	49	172	201	94	1	10	4	3	13	27	121	5	5	100	498	94
LongPrimeSieve_Ox.exe	96	731	29	218	290	6	130	12	49	94	6	879	23	4	5	100	46	167	182	100	0	10	4	12	12	23	127	2	4	98	423	90
LongPrimeSieve_Oy.exe	59	1750	29	230	266	3	152	9	90	93	14	1429	17	0	5	88	45	132	198	93	0	10	4	3	13	21	102	2	4	100	438	85

filename	acmp	add	and	call	cjmp	cmov	cmp	div	jmp	lea	load	mov	mul	nop	not	null	or	pop	push	ret	rol	ror	sal	sar	shl	shr	store	sub	swi	trap	ucall	upush	xor
Win32Window_all.exe	40	337	15	66	97	4	53	0	24	32	1	313	5	0	3	10	15	96	73	49	0	2	0	0	5	2	5	67	3	8	47	147	54
Win32Window_GL.exe	37	1990	17	70	97	3	72	1	34	32	0	393	5	0	3	8	15	86	79	53	0	2	2	1	3	1	1	29	3	5	49	133	51
Win32Window_GT.exe	37	2039	16	71	94	3	86	2	31	32	0	388	5	1	3	6	15	87	81	52	0	2	2	4	3	1	3	28	3	7	48	133	53
Win32Window_O0.exe	37	2039	16	71	94	3	86	2	31	32	0	388	5	1	3	6	15	87	81	52	0	2	2	4	3	1	3	28	3	7	48	133	53
Win32Window_O1.exe	40	324	15	70	93	4	50	0	23	32	0	316	5	1	3	6	15	95	73	51	0	2	0	0	5	4	7	65	3	7	46	155	56
Win32Window_O2.exe	41	2052	11	76	92	4	55	0	14	33	0	325	5	2	3	5	14	107	82	59	0	2	0	0	3	0	1	30	3	4	49	130	58
Win32Window_Ob1.exe	37	2200	15	67	95	3	86	2	31	32	0	382	5	1	3	6	15	84	81	49	0	2	2	4	3	1	7	30	3	6	48	133	52
Win32Window_Ob2.exe	41	7365	38	70	324	3	124	2	32	32	1	397	60	5	3	56	31	187	92	49	1	2	4	5	3	1	5	85	3	7	49	216	87
Win32Window_Oi.exe	37	2039	16	71	94	3	86	2	31	32	0	388	5	1	3	6	15	87	81	52	0	2	2	4	3	1	3	28	3	7	48	133	53
Win32Window_Os.exe	37	2037	19	70	92	3	65	0	31	32	0	338	5	0	3	6	14	92	82	52	0	2	2	1	3	1	1	27	3	4	48	128	52
Win32Window_Ot.exe	37	2039	16	71	94	3	86	2	31	32	0	388	5	1	3	6	15	87	81	52	0	2	2	4	3	1	3	28	3	7	48	133	53
Win32Window_Ox.exe	41	2052	11	76	92	4	55	0	14	33	0	325	5	2	3	5	14	107	82	59	0	2	0	0	3	0	1	30	3	4	49	130	58
Win32Window_Oy.exe	37	2177	17	71	94	3	86	2	31	33	1	389	5	1	3	6	16	91	81	52	0	2	2	4	3	1	6	32	3	6	48	136	56

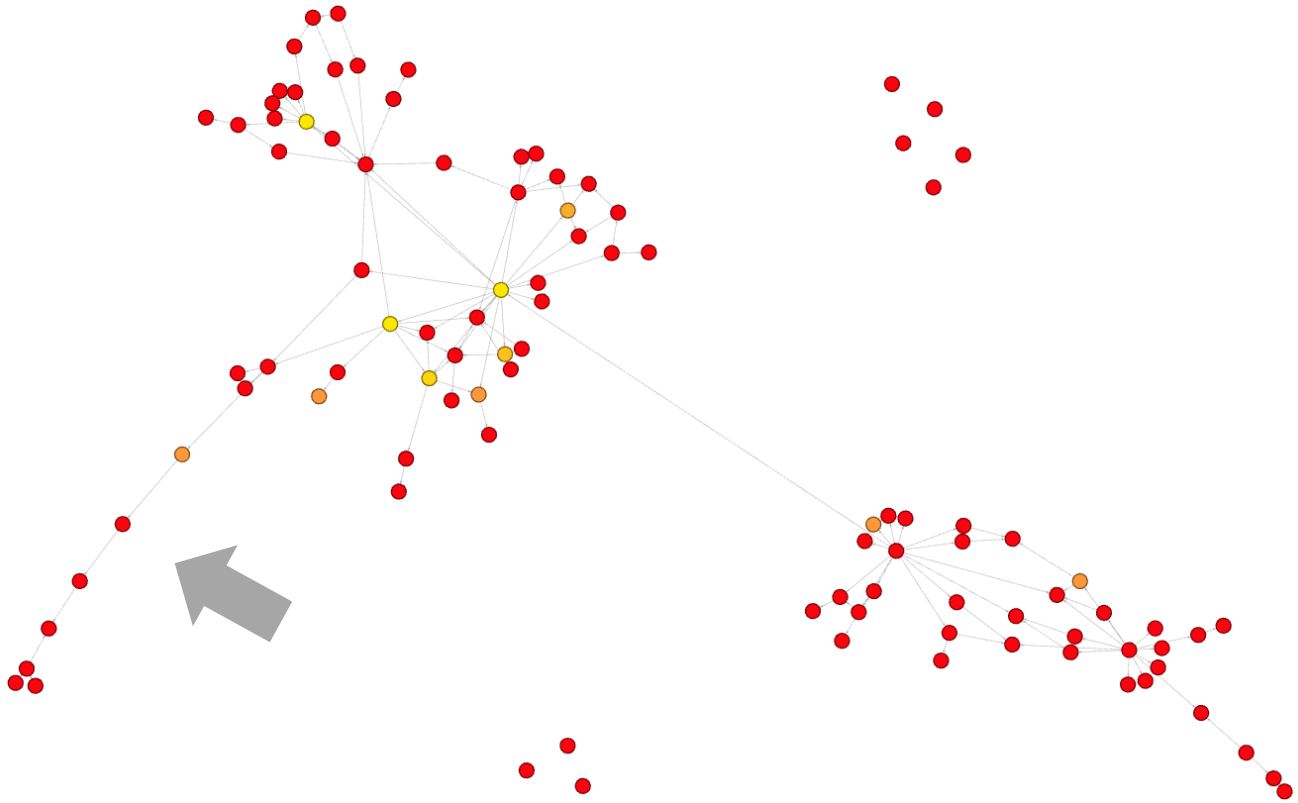

Same same but different
RandomNG total instruction counts
and Matplotlib
6000
5000
4000
3000
2000
1000
0
all
GL
GT
O0
O1
O2
Ob1
Ob2
Oi
Os
Ot
Oy
acmp
add
and
call
cjmp
cmov
cmp
div
invalid
io
jmp
lea
load
mov
mul
nop
not
null
or
pop
push
ret
rol
ror
sal
sar
shl
shr
store
sub
swi
trap
ucall
upush
xor


400
350
300
250
200
150
100
50
0
and
call
cjmp
cmov
jmp
push
ret
LongPrimeSieve_all.exe
LongPrimeSieve_GL.exe
LongPrimeSieve_GT.exe
LongPrimeSieve_O0.exe
LongPrimeSieve_O1.exe
LongPrimeSieve_O2.exe
LongPrimeSieve_Ob1.exe
LongPrimeSieve_Ob2.exe
LongPrimeSieve_Oi.exe
LongPrimeSieve_Os.exe
LongPrimeSieve_Ot.exe
LongPrimeSieve_Ox.exe
LongPrimeSieve_Oy.exe

LongPrimeSieve all vs. O0: jmp instruction

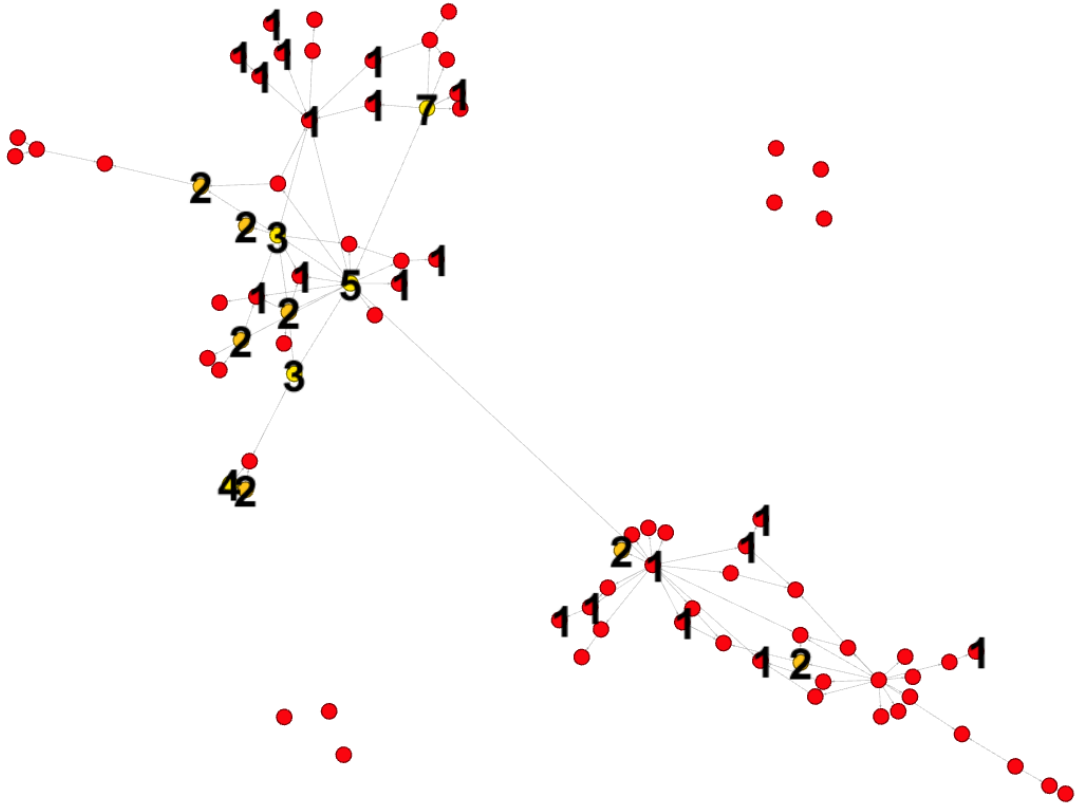


all: 83 nodes / 101 edges

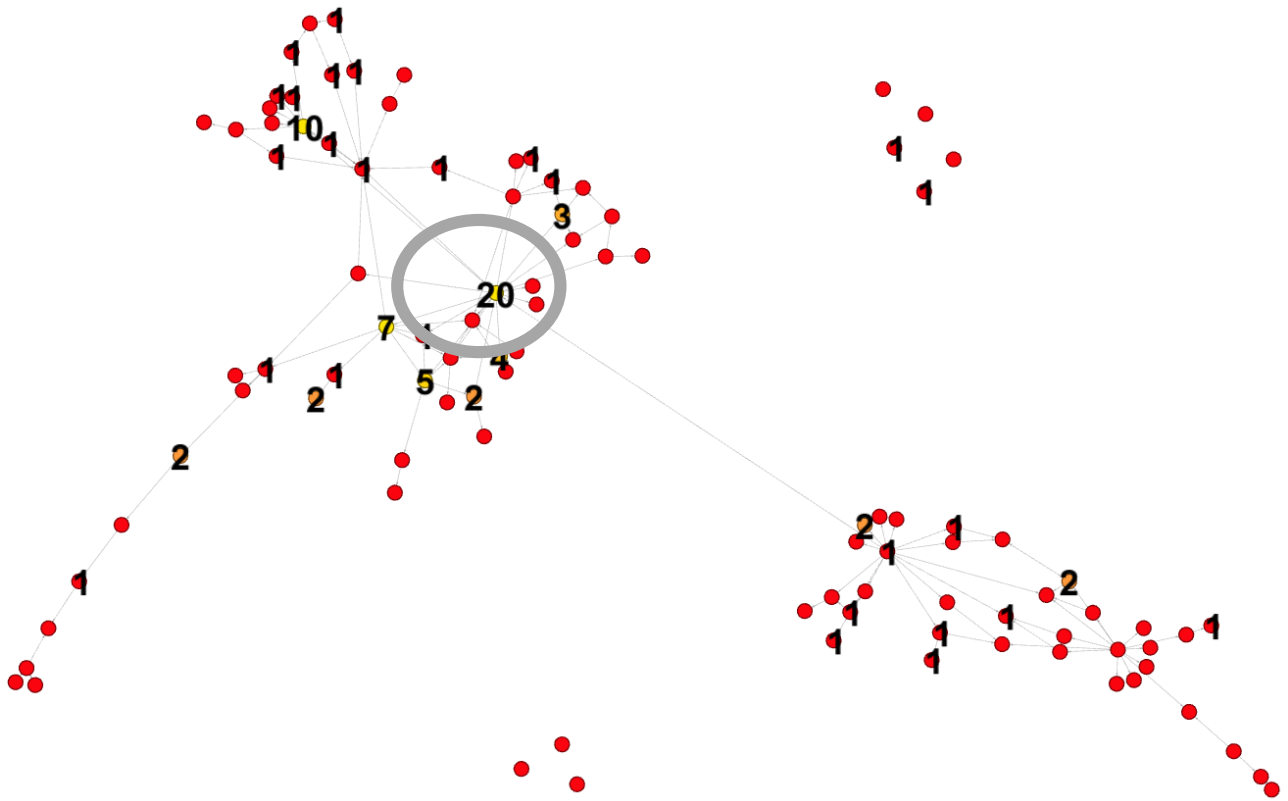


O0: 103 nodes / 129 edges

LongPrimeSieve all vs. O0: jmp instruction



all: 83 nodes / 101 edges

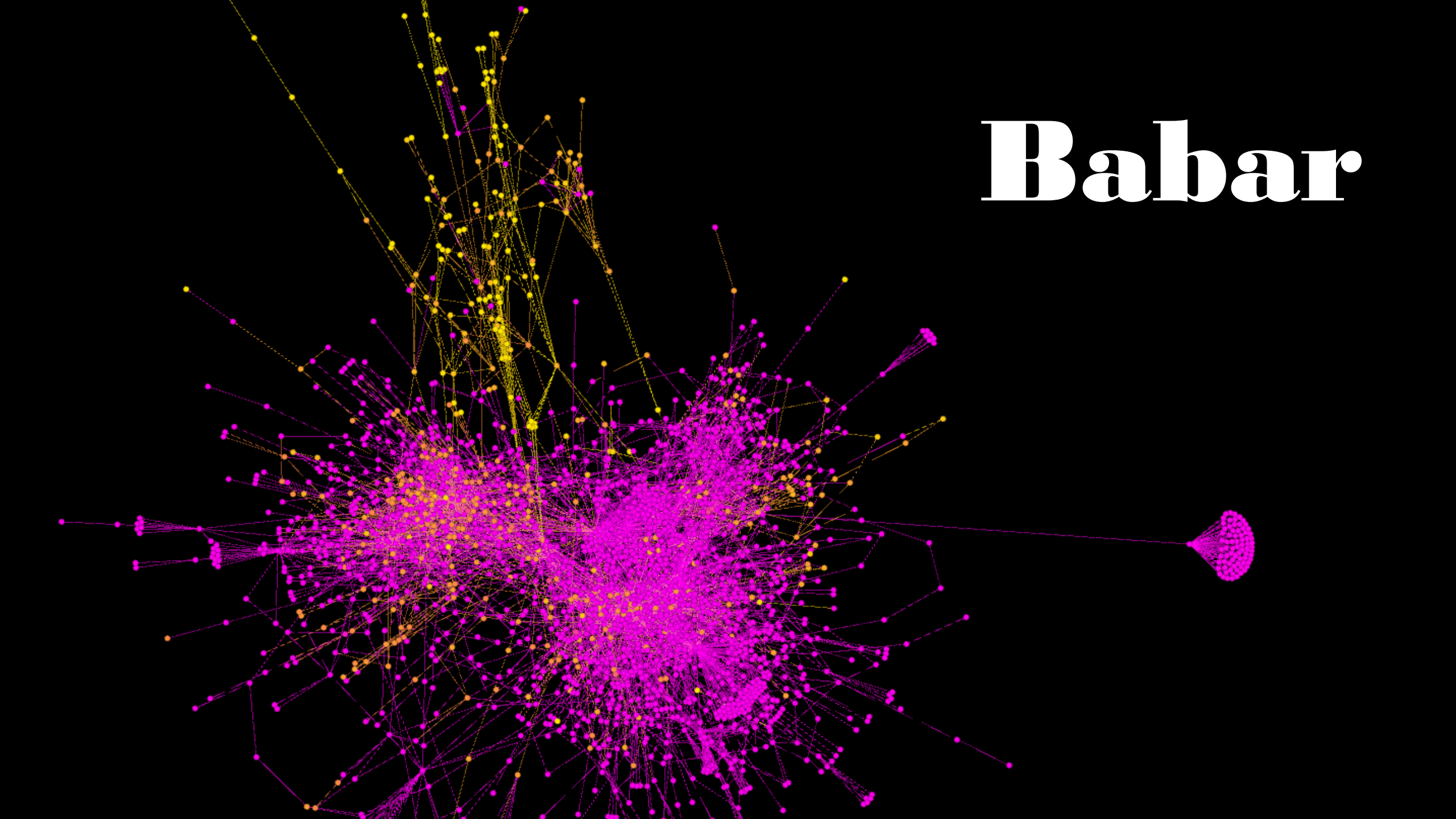


O0: 103 nodes / 129 edges

Mnemonicism

Arithmetic instructions as indicator for cryptography, compression or codecs

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


Babar

Mnemonic variance distribution

Variance of mnemonics per function

Varying bucket sizes

Unified comparable values

Resilient to little changes

Flag	Variance distribution
All	22.32.8.11.3.1.2.0.0.0.0.0.0.0.2.0.0.0.0
GL	26.38.7.15.11.2.0.0.0.0.0.0.0.1.2.1.2.0.0
GT	24.38.7.16.10.2.0.0.0.0.0.0.0.1.1.0.1.0.0
O0	24.38.7.16.10.2.0.0.0.0.0.0.0.1.1.0.1.0.0
O1	25.43.12.14.3.1.2.1.0.0.0.0.0.0.1.0.0.0.0
O2	25.41.9.15.4.2.0.1.0.1.0.0.0.0.1.0.0.0.1
Ob1	23.29.5.10.10.2.0.0.0.0.0.0.0.2.1.0.1.0.1
Ob2	23.29.5.10.10.2.0.0.0.0.0.0.0.2.1.0.1.0.1
Oi	24.37.7.18.10.2.0.0.0.0.0.0.0.1.1.0.2.0.0
Os	24.40.7.21.3.2.0.0.0.0.0.0.0.2.0.1.0.0.0
Ot	24.38.7.16.10.2.0.0.0.0.0.0.0.1.1.0.1.0.0
Ox	25.41.10.15.3.1.0.2.1.1.0.0.0.0.1.0.0.0.1
Oy	25.38.7.16.10.2.0.0.0.0.0.0.0.1.1.1.1.0.0

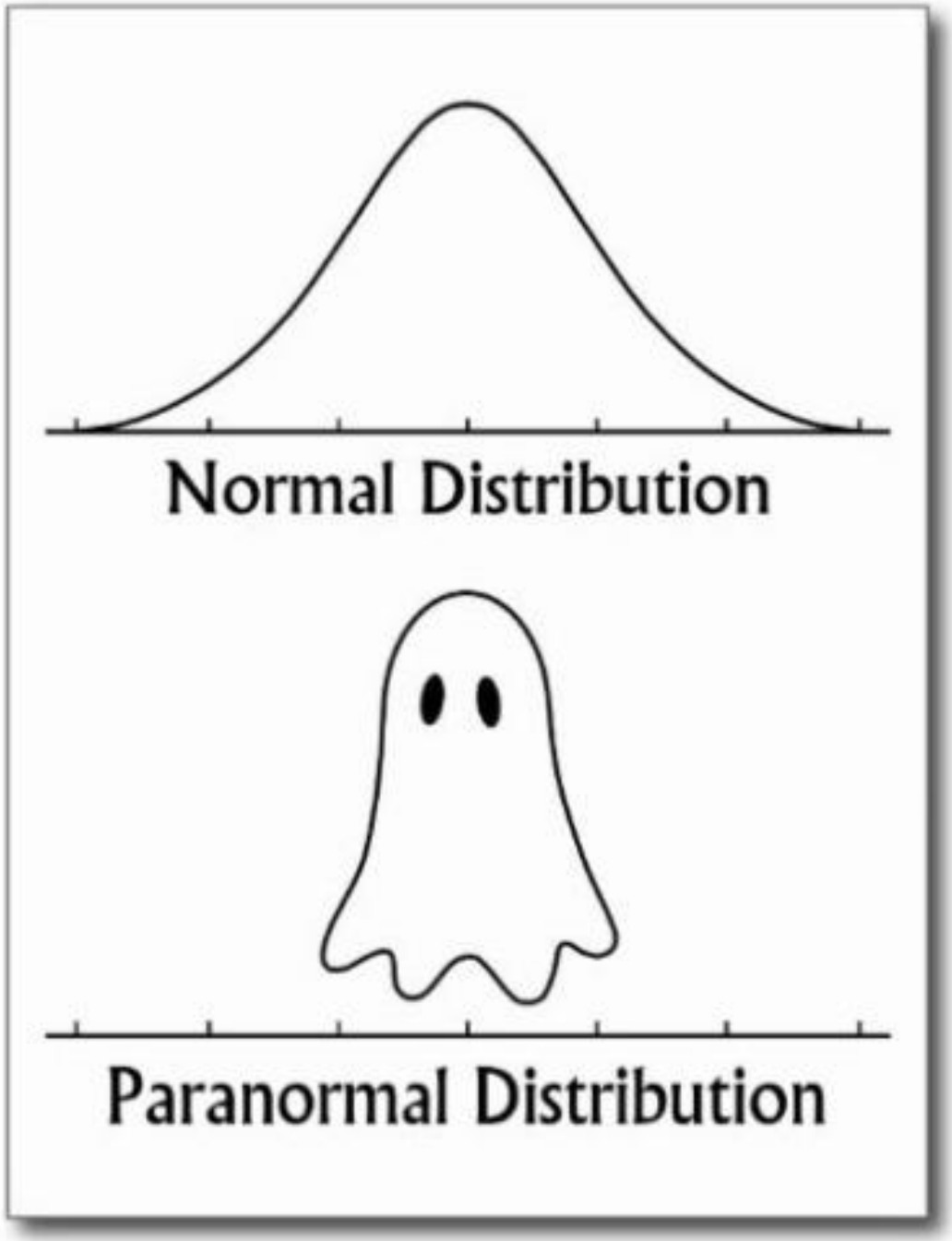
Mnemonic variance distribution

Variance of mnemonics per function

Varying bucket sizes

Unified comparable values

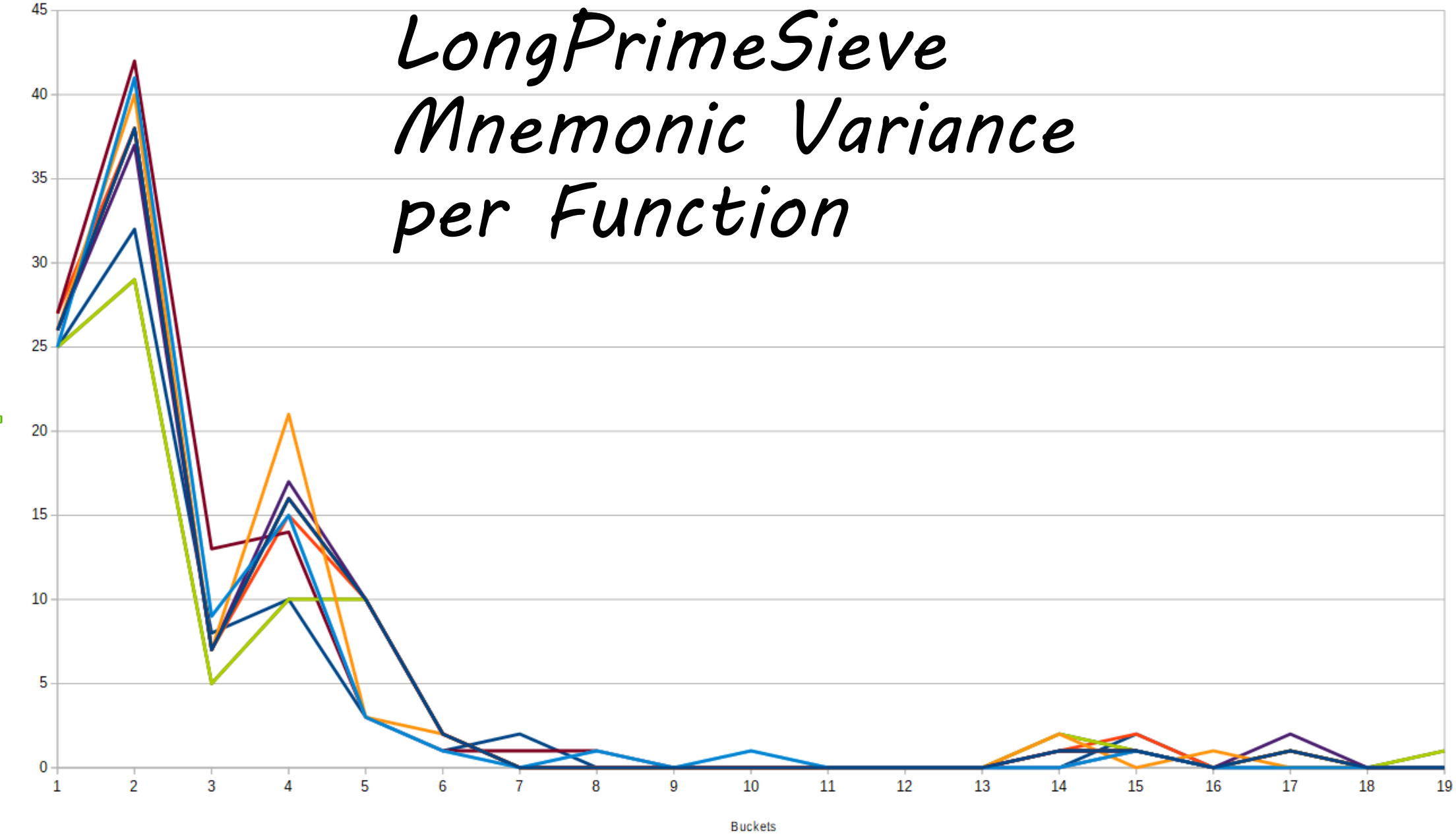
Resilient to little changes



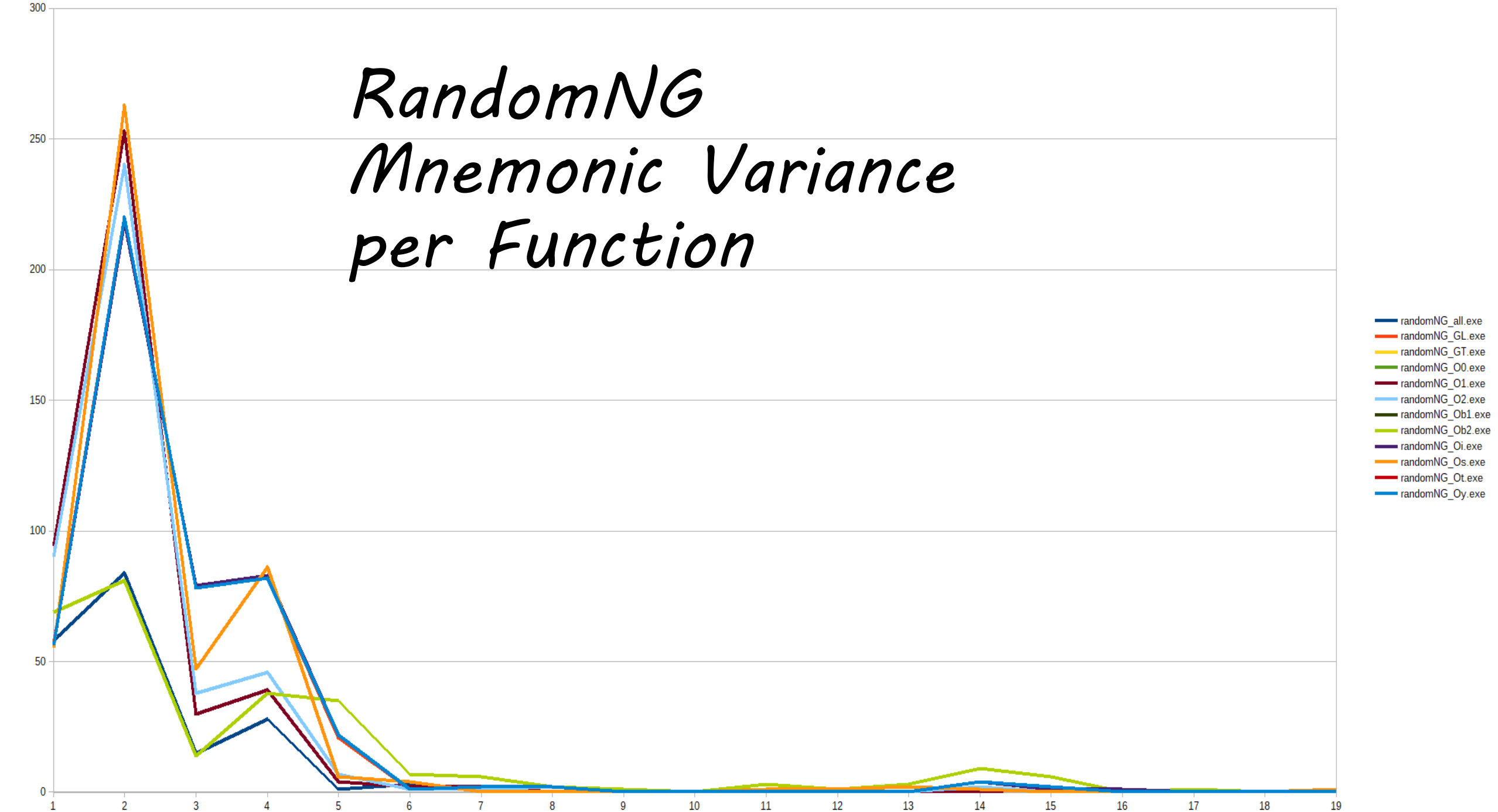
LongPrimeSieve
Mnemonic Variance per Function

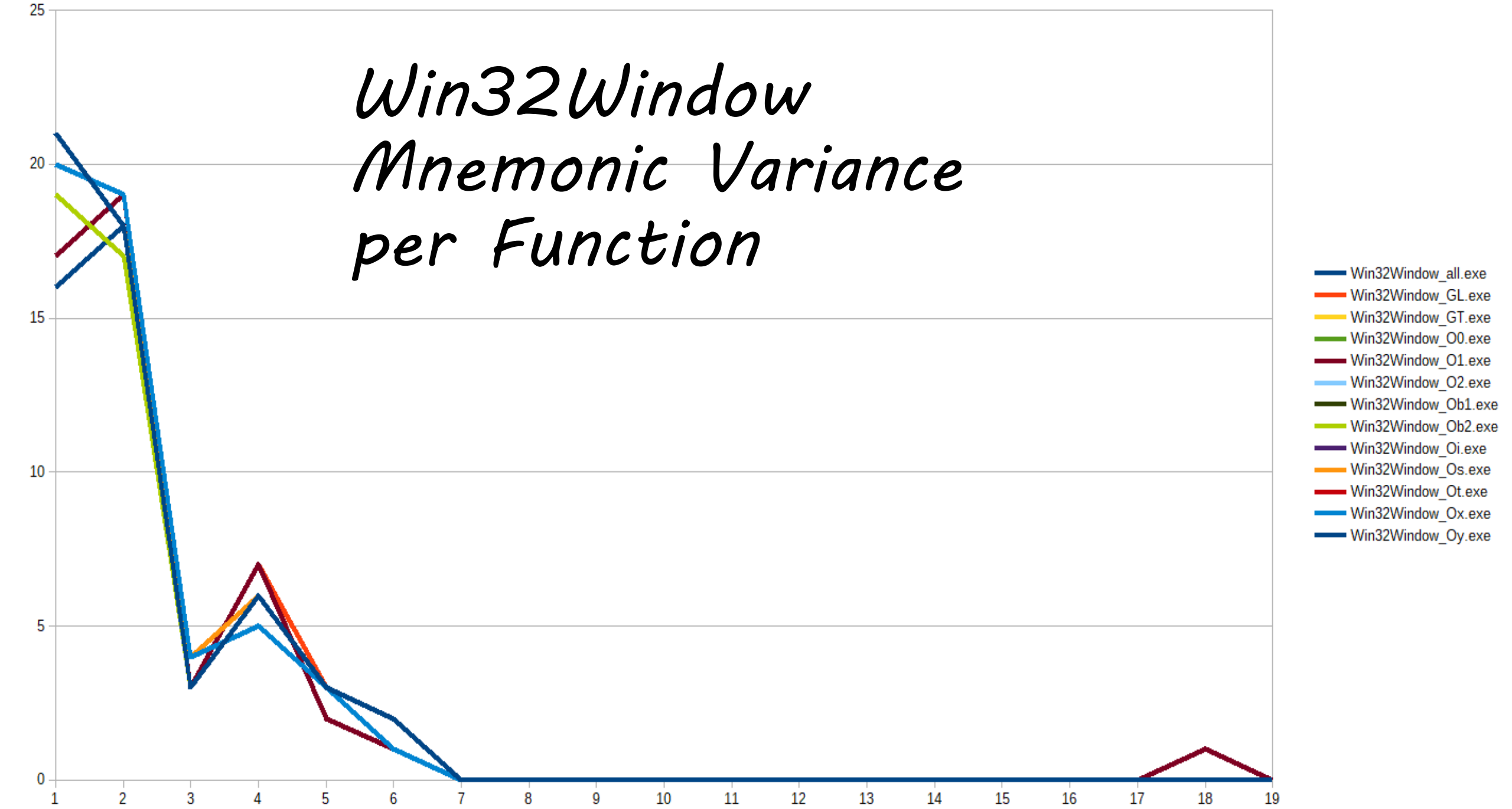
	0	5	10	50	100	150	200	250	300	350	400	450	500	1000	2000	3000	4000	5000	10000
LongPrimeSieve_all.exe	25	32	8	10	3	1	2	0	0	0	0	0	0	0	2	0	0	0	0
LongPrimeSieve_GL.exe	27	38	7	15	10	2	0	0	0	0	0	0	0	1	2	0	1	0	0
LongPrimeSieve_GT.exe	26	38	7	16	10	2	0	0	0	0	0	0	0	1	1	0	1	0	0
LongPrimeSieve_O0.exe	26	38	7	16	10	2	0	0	0	0	0	0	0	1	1	0	1	0	0
LongPrimeSieve_O1.exe	27	42	13	14	3	1	1	1	0	0	0	0	0	0	1	0	0	0	0
LongPrimeSieve_O2.exe	25	41	9	15	3	1	0	1	0	1	0	0	0	0	1	0	0	0	0
LongPrimeSieve_Ob1.exe	25	29	5	10	10	2	0	0	0	0	0	0	0	2	1	0	1	0	1
LongPrimeSieve_Ob2.exe	25	29	5	10	10	2	0	0	0	0	0	0	0	2	1	0	1	0	1
LongPrimeSieve_Oi.exe	26	37	7	17	10	2	0	0	0	0	0	0	0	1	1	0	2	0	0
LongPrimeSieve_Os.exe	26	40	7	21	3	2	0	0	0	0	0	0	0	2	0	1	0	0	0
LongPrimeSieve_Ot.exe	26	38	7	16	10	2	0	0	0	0	0	0	0	1	1	0	1	0	0
LongPrimeSieve_Ox.exe	25	41	9	15	3	1	0	1	0	1	0	0	0	0	1	0	0	0	0
LongPrimeSieve_Oy.exe	26	38	7	16	10	2	0	0	0	0	0	0	0	1	1	0	1	0	0

LongPrimeSieve Mnemonic Variance per Function

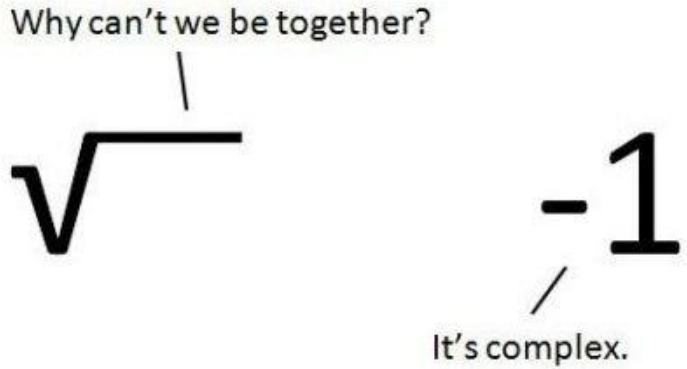


RandomNG Mnemonic Variance per Function




Win32Window
Mnemonic Variance
per Function
25
20
15
10
5
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
Win32Window_all.exe
Win32Window_GL.exe
Win32Window_GT.exe
Win32Window_O0.exe
Win32Window_O1.exe
Win32Window_O2.exe
Win32Window_Ob1.exe
Win32Window_Ob2.exe
Win32Window_Oi.exe
Win32Window_Os.exe
Win32Window_Ot.exe
Win32Window_Ox.exe
Win32Window_Oy.exe

Numbers <3



Comparing by count only makes sense on the same source base
No, I still don't diff binaries

filesize
codesecsize
functionstotal
refslocal
refsglobalvar
refsunknown
refsindirect

apitotal
datarefcount
ratiofunc
ratioapi
getprocaddress
memallocation
createthread

RandomNG

attribute	largest	smallest	average	variance	stddeviation	relstddeviation
filesize	52224.00000	24064.00000	43690.66667	62535907.55556	7907.96482	18.10%
codesecsize	38912.00000	12800.00000	31402.66667	55152184.88889	7426.45170	23.65%
functionstotal	505.00000	227.00000	445.41667	8275.24306	90.96836	20.42%
refslocal	1214.00000	364.00000	985.83333	103856.30556	322.26744	32.69%
refsglobalvar	0.00000	0.00000	0.00000	0.00000	0.00000	nan%
refsunknown	0.00000	0.00000	0.00000	0.00000	0.00000	nan%
refsindirect	0.00000	0.00000	0.00000	0.00000	0.00000	nan%
apitotal	184.00000	111.00000	153.75000	307.85417	17.54577	11.41%
datarefcount	350.00000	212.00000	316.83333	2236.97222	47.29664	14.93%
ratiofunc	21.57738	8.48067	14.83268	12.45490	3.52915	23.79%
ratioapi	8.67188	4.33709	5.19116	1.72738	1.31430	25.32%
getprocaddress	0.00000	0.00000	0.00000	0.00000	0.00000	nan%
memallocation	3.00000	2.00000	2.91667	0.07639	0.27639	9.48%
createthread	0.00000	0.00000	0.00000	0.00000	0.00000	nan%

LongPrimeSieve

attribute	largest	smallest	average	variance	stddeviation	relstddeviation
filesize	24064.00000	18432.00000	22803.69231	3412525.443	1847.30221	8.10%
codesecsize	14336.00000	9216.00000	13154.46154	2959590.248	1720.34597	13.08%
functionstotal	135.00000	114.00000	129.69231	46.82840	6.84313	5.28%
refslocal	262.00000	215.00000	250.30769	238.82840	15.45407	6.17%
refsglobalvar	0.00000	0.00000	0.00000	0.00000	0.00000	nan%
refsunknown	0.00000	0.00000	0.00000	0.00000	0.00000	nan%
refsindirect	0.00000	0.00000	0.00000	0.00000	0.00000	nan%
apitotal	133.00000	106.00000	127.92308	44.07101	6.63860	5.19%
datarefcount	239.00000	198.00000	225.61538	228.85207	15.12786	6.71%
ratiofunc	13.08594	8.37054	10.02846	1.88340	1.37237	13.68%
ratioapi	12.20703	8.99833	9.85776	1.11405	1.05548	10.71%
getprocaddress	0.00000	0.00000	0.00000	0.00000	0.00000	nan%
memallocation	2.00000	2.00000	2.00000	0.00000	0.00000	0.00%
createthread	1.00000	1.00000	1.00000	0.00000	0.00000	0.00%

Win32Window

attribute	largest	smallest	average	variance	stddeviation	relstddeviation
filesize	38400.00000	37888.00000	38321.23077	34125.25444	184.73022	0.48%
codesecsize	4608.00000	4096.00000	4529.23077	34125.25444	184.73022	4.08%
functionstotal	82.00000	75.00000	79.69231	3.90533	1.97619	2.48%
refslocal	99.00000	88.00000	92.84615	9.97633	3.15853	3.40%
refsglobalvar	1.00000	0.00000	0.07692	0.07101	0.26647	346.41%
refsunknown	0.00000	0.00000	0.00000	0.00000	0.00000	nan%
refsindirect	0.00000	0.00000	0.00000	0.00000	0.00000	nan%
apitotal	75.00000	72.00000	73.84615	0.74556	0.86346	1.17%
datarefcount	120.00000	114.00000	115.61538	3.77515	1.94297	1.68%
ratiofunc	18.79883	16.92708	17.61151	0.23085	0.48047	2.73%
ratioapi	17.57812	16.05903	16.32612	0.29093	0.53938	3.30%
getprocaddress	0.00000	0.00000	0.00000	0.00000	0.00000	nan%
memallocation	0.00000	0.00000	0.00000	0.00000	0.00000	nan%
createthread	0.00000	0.00000	0.00000	0.00000	0.00000	nan%

Larger code size - larger changes

Deviation between 3 - 30%

(Suspected) more optimization in C than C++ source

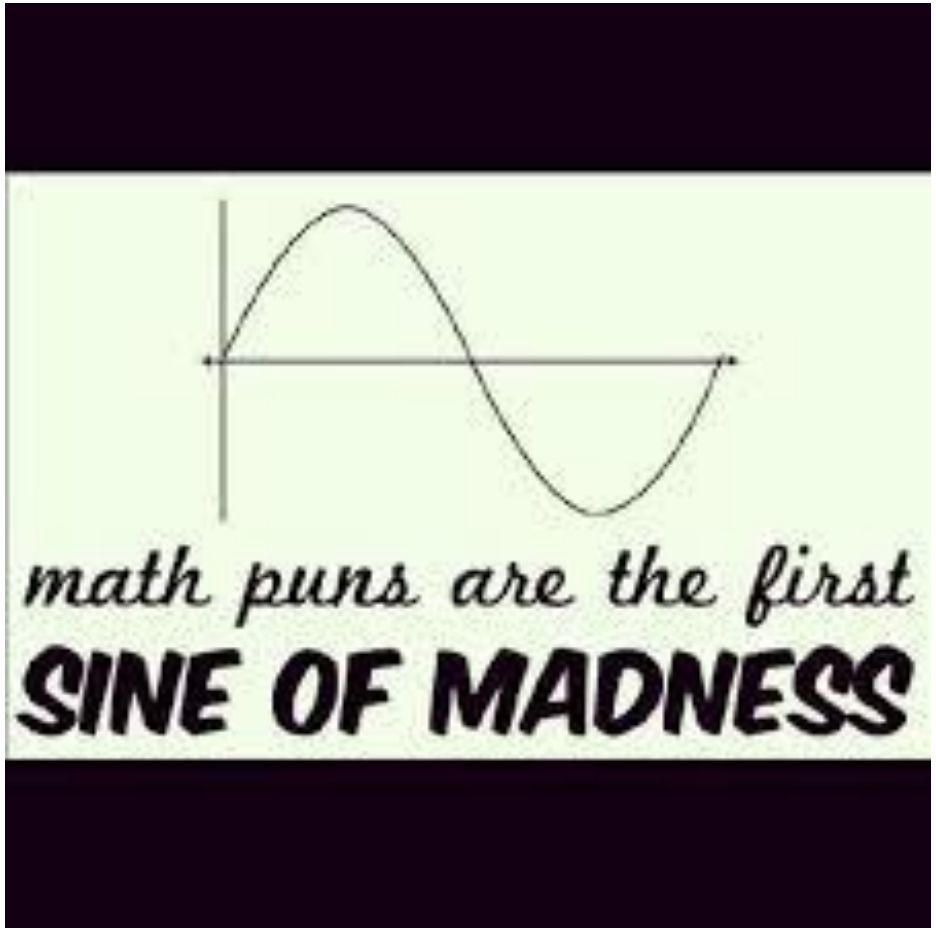
Small changes in small numbers show (false) big outliers


WHY?

Detection and analysis of optimization settings

Shown values are bound to one specific source base

Introducing abstraction by dividing by code size



	randomNG_all.exe	LongPrimeSieve_all.exe	Win32Window_all.exe
functionstotal	0.01773	0.01237	0.01831
refslocal	0.02844	0.02333	0.02148
refsglobalvar	0.00000	0.00000	0.00000
refsunknown	0.00000	0.00000	0.00000
refsindirect	0.00000	0.00000	0.00000
apitotal	0.00867	0.01150	0.01758
datarefcount	0.01656	0.02159	0.02808
ratiofunc	0.00139	0.00134	0.00447
ratioapi	0.00068	0.00125	0.00429
getprocaddress	0.00000	0.00000	0.00000
memallocation	0.00016	0.00022	0.00000
createthread	0.00000	0.00011	0.00000
push	0.04711	0.04612	0.05371
and	0.00797	0.00369	0.00366
call	0.03164	0.02778	0.02759
jmp	0.03719	0.03516	0.02954
ret	0.00945	0.00781	0.01196
mov	0.10117	0.07899	0.07739

	randomNG_O0.exe	LongPrimeSieve_O0.exe	Win32Window_O0.exe
functionstotal	0.01450	0.00942	0.01758
refslocal	0.03484	0.01828	0.02018
refsglobalvar	0.00000	0.00000	0.00000
refsunknown	0.00000	0.00000	0.00000
refsindirect	0.00000	0.00000	0.00000
apitotal	0.00437	0.00907	0.01606
datarefcount	0.01005	0.01653	0.02496
ratiofunc	0.00042	0.00066	0.00381
ratioapi	0.00013	0.00063	0.00349
getprocaddress	0.00000	0.00000	0.00000
memallocation	0.00009	0.00014	0.00000
createthread	0.00000	0.00007	0.00000
push	0.05969	0.04883	0.04644
and	0.00118	0.00614	0.00347
call	0.03697	0.02309	0.02582
jmp	0.01623	0.03174	0.02713
ret	0.01114	0.00656	0.01128
mov	0.12405	0.10156	0.08485

	randomNG_Ob2.ex	LongPrimeSieve_Ob2.ex	Win32Window_Ob2.exe
functionstotal	0.00848	0.00837	0.01693
refslocal	0.01252	0.01590	0.01931
refsglobalvar	0.00000	0.00000	0.00022
refsunknown	0.00000	0.00000	0.00000
refsindirect	0.00000	0.00000	0.00000
apitotal	0.00473	0.00928	0.01606
datarefcount	0.00876	0.01639	0.02474
ratiofunc	0.00022	0.00058	0.00367
ratioapi	0.00012	0.00065	0.00349
getprocaddress	0.00000	0.00000	0.00000
memallocation	0.00008	0.00014	0.00000
createthread	0.00000	0.00007	0.00000
push	0.02858	0.04004	0.06684
and	0.00105	0.00481	0.00825
call	0.01575	0.02086	0.02582
jmp	0.02048	0.02895	0.07726
ret	0.00478	0.00544	0.01063
mov	0.15769	0.10603	0.08681

	randomNG_Oi.exe	LongPrimeSieve_Oi.exe	Win32Window_Oi.exe
functionstotal	0.01448	0.00935	0.01758
refslocal	0.03484	0.01821	0.02018
refsglobalvar	0.00000	0.00000	0.00000
refsunknown	0.00000	0.00000	0.00000
refsindirect	0.00000	0.00000	0.00000
apitotal	0.00434	0.00900	0.01606
datarefcount	0.01000	0.01653	0.02496
ratiofunc	0.00042	0.00065	0.00381
ratioapi	0.00012	0.00063	0.00349
getprocaddress	0.00000	0.00000	0.00000
memallocation	0.00009	0.00014	0.00000
createthread	0.00000	0.00007	0.00000
push	0.05966	0.04408	0.04644
and	0.00118	0.00202	0.00347
call	0.03697	0.02295	0.02582
jmp	0.01626	0.02490	0.02713
ret	0.01114	0.00656	0.01128
mov	0.12443	0.10086	0.08485

fdiskyou / malware

Malware source code samples leaked online uploaded to GitHub for those who want to analyze the code.

malware malware-samples banking-trojan botnet sales-trojan

99 commits 1 branch 0 releases 1 contributor

Branch: master New pull request Find file Clone or download

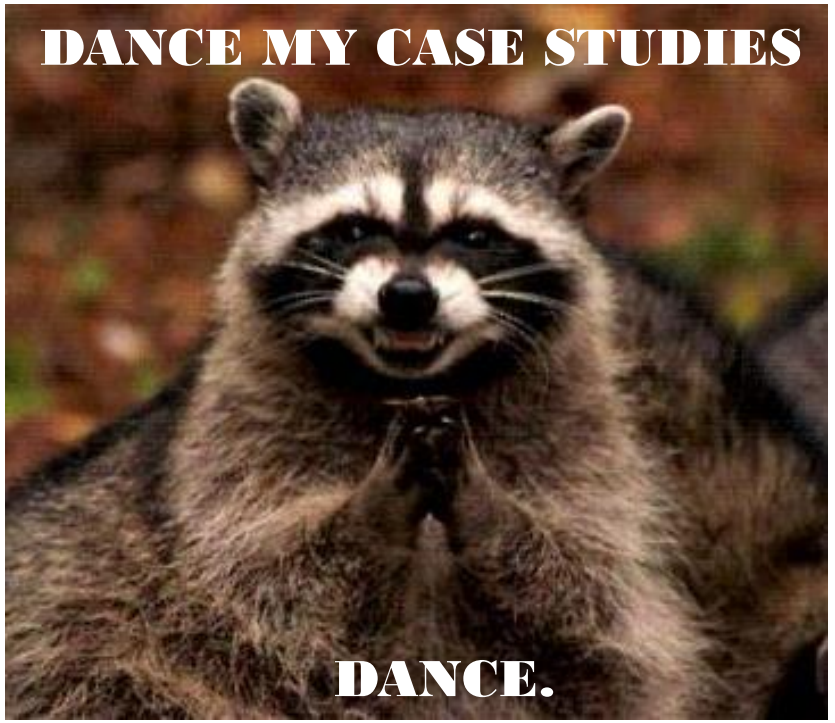
fdiskyou tweak Latest commit e51ec31 6 days ago

Alina	Update README.md	2 months ago
BleedingLife2	Update README.md	2 months ago
Carberp Botnet	add a few more references	5 months ago
Carberp	Update README.md	2 months ago
Crimepack3.1.3	Update README.md	2 months ago

Dexter - crappy POS scraper
Winkey - crappy Keylogger
Rbot - crappy .. Something something

VisualStudio 2017

Optimization	/Od /O1 /O2 /Ox
Function Expansion	/Ob0 /Ob1 /Ob2
Intrinsic Functions	/Oi
Size or Speed	/Os /Ot
Omit Frame Pointer	/Oy /Oy-
Whole Program Opt.	/GL



Rbot File Geometry

	largest	smallest	average	variance	std deviation	rel std dev
filesize	523776.00000	519168.00000	520661.33333	2685155.55556	1638.64443	0.31%
codesecsize	420352.00000	416256.00000	417578.66667	2029795.55556	1424.70894	0.34%
sectioncount	6.00000	6.00000	6.00000	0.00000	0.00000	0.00%
secsize1	420352.00000	416256.00000	417578.66667	2029795.55556	1424.70894	0.34%
secsize2	72192.00000	71680.00000	71722.66667	20024.88889	141.50932	0.20%
secsize3	11776.00000	11264.00000	11733.33333	20024.88889	141.50932	1.21%
secsize4	512.00000	512.00000	512.00000	0.00000	0.00000	0.00%
secsize5	512.00000	512.00000	512.00000	0.00000	0.00000	0.00%
secsize6	17920.00000	17408.00000	17578.66667	58254.22222	241.35911	1.37%
secent1	6.63672	6.62003	6.62644	0.00003	0.00545	0.08%
secent2	5.58557	5.54333	5.54810	0.00013	0.01132	0.20%
secent3	3.70221	3.46303	3.67129	0.00399	0.06317	1.72%
secent4	0.02039	0.02039	0.02039	0.00000	0.00000	0.00%
secent5	4.70824	4.70824	4.70824	0.00000	0.00000	0.00%
secent6	6.78160	6.71266	6.76198	0.00043	0.02066	0.31%

Dexter File Geometry

	largest	smallest	average	variance	std deviation	rel std dev
filesize	33280.00000	26624.00000	31402.66667	5432206.22222	2330.70938	7.42%
codesecsize	21504.00000	17408.00000	20480.00000	2402986.66667	1550.15698	7.57%
sectioncount	4.00000	3.00000	3.08333	0.07639	0.27639	8.96%
secsize1	21504.00000	17408.00000	20480.00000	2402986.66667	1550.15698	7.57%
secsize2	8192.00000	5632.00000	7765.33333	910222.22222	954.05567	12.29%
secsize3	2560.00000	512.00000	1962.66667	211171.55556	459.53406	23.41%
secsize4	0.00000	0.00000	0.00000	0.00000	0.00000	
secsize5	0.00000	0.00000	0.00000	0.00000	0.00000	
secsize6	0.00000	0.00000	0.00000	0.00000	0.00000	
secent1	6.32626	5.94436	6.11377	0.00955	0.09772	1.60%
secent2	4.56973	4.07940	4.48058	0.03222	0.17951	4.01%
secent3	6.18090	0.06116	5.54013	2.79873	1.67294	30.20%
secent4	0.00000	0.00000	0.00000	0.00000	0.00000	
secent5	0.00000	0.00000	0.00000	0.00000	0.00000	
secent6	0.00000	0.00000	0.00000	0.00000	0.00000	

Winkey File Geometry

	largest	smallest	average	variance	std deviation	rel std dev
filesize	34816.00000	25088.00000	27690.66667	6355171.55556	2520.94656	9.10%
codesecsize	20992.00000	12288.00000	14293.33333	5088142.22222	2255.69107	15.78%
sectioncount	5.00000	5.00000	5.00000	0.00000	0.00000	0.00%
secsize1	20992.00000	12288.00000	14293.33333	5088142.22222	2255.69107	15.78%
secsize2	9728.00000	8704.00000	9301.33333	254862.22222	504.83881	5.43%
secsize3	1024.00000	1024.00000	1024.00000	0.00000	0.00000	0.00%
secsize4	512.00000	512.00000	512.00000	0.00000	0.00000	0.00%
secsize5	1536.00000	1536.00000	1536.00000	0.00000	0.00000	0.00%
secsize6	0.00000	0.00000	0.00000	0.00000	0.00000	
secent1	6.36596	5.99111	6.26617	0.01040	0.10198	1.63%
secent2	5.08332	4.80283	4.93778	0.00638	0.07988	1.62%
secent3	2.91279	2.87559	2.90221	0.00017	0.01308	0.45%
secent4	4.70150	4.69612	4.70016	0.00001	0.00233	0.05%
secent5	6.14802	5.58756	5.81690	0.02771	0.16646	2.86%
secent6	0.00000	0.00000	0.00000	0.00000	0.00000	

Rbot Graph Geometry

	largest	smallest	average	variance	std deviation	rel std dev
functionstotal	3926.00000	3896.00000	3914.33333	109.88889	10.48279	0.27%
refslocal	9369.00000	8789.00000	9201.50000	55820.25000	236.26309	2.57%
refsglobalvar	782.00000	768.00000	778.50000	13.41667	3.66288	0.47%
refsunknown	2.00000	1.00000	1.91667	0.07639	0.27639	14.42%
refsindirect	0.00000	0.00000	0.00000	0.00000	0.00000	
apitotal	1024.00000	1011.00000	1017.58333	11.90972	3.45105	0.34%
apimisses	0.00000	0.00000	0.00000	0.00000	0.00000	
stringsreferenced	1514.00000	1427.00000	1450.00000	1373.66667	37.06301	2.56%
stringsdangling	0.00000	0.00000	0.00000	0.00000	0.00000	
stringsnoref	6225.00000	6208.00000	6214.41667	16.40972	4.05089	0.07%
ratiofunc	9.42011	9.30116	9.37403	0.00229	0.04780	0.51%
ratioapi	2.44021	2.42416	2.43687	0.00003	0.00548	0.23%
ratiostring	3.60614	3.42397	3.47214	0.00590	0.07680	2.21%

Dexter Graph Geometry

	largest	smallest	average	variance	std deviation	rel std dev
functionstotal	111.00000	91.00000	104.25000	22.35417	4.72802	4.54%
refslocal	536.00000	317.00000	405.33333	2181.72222	46.70891	11.52%
refsglobalvar	5.00000	3.00000	3.16667	0.30556	0.55277	17.46%
refsunknown	1.00000	0.00000	0.08333	0.07639	0.02763	31.66%
refsindirect	0.00000	0.00000	0.00000	0.00000	0.00000	
apitotal	357.00000	324.00000	350.00000	99.00000	9.94987	2.84%
apimisses	0.00000	0.00000	0.00000	0.00000	0.00000	
stringsreferenced	110.00000	106.00000	109.58333	1.24306	1.11492	1.02%
stringsdangling	0.00000	0.00000	0.00000	0.00000	0.00000	
stringsnoref	604.00000	499.00000	524.00000	785.33333	28.02380	5.35%
ratiofunc	6.02214	4.69680	5.11325	0.14216	0.37704	7.37%
ratioapi	18.69420	16.46205	17.15942	0.87618	0.93605	5.46%
ratiostring	6.08915	5.11533	5.38092	0.15993	0.39991	7.43%

Winkey Graph Geometry

	largest	smallest	average	variance	std deviation	rel std dev
functionstotal	298.00000	186.00000	246.08333	1553.40972	39.41332	16.02%
refslocal	564.00000	230.00000	413.00000	14865.83333	121.92552	29.52%
refsglobalvar	0.00000	0.00000	0.00000	0.00000	0.00000	
refsunknown	0.00000	0.00000	0.00000	0.00000	0.00000	
refsindirect	0.00000	0.00000	0.00000	0.00000	0.00000	
apitotal	190.00000	146.00000	171.66667	356.72222	18.88709	11.00%
apimisses	3.00000	1.00000	1.16667	0.30556	0.55277	47.38%
stringsreference	14.00000	12.00000	13.75000	0.35417	0.59512	4.33%
stringsdangling	0.00000	0.00000	0.00000	0.00000	0.00000	
stringsnoref	408.00000	367.00000	383.50000	127.41667	11.28790	2.94%
ratiofunc	20.58293	12.97433	17.37598	6.86519	2.62015	15.08%
ratioapi	14.04748	8.95579	12.14965	1.99400	1.41209	11.62%
ratiostring	1.13932	0.57165	0.98512	0.02216	0.14886	15.11%

API Level Indicators

		largest	smallest	average	variance	std deviation	rel std dev
Rbot	getprocaddress	59.00000	59.00000	59.00000	0.00000	0.00000	0.00%
	memallocation	13.00000	13.00000	13.00000	0.00000	0.00000	0.00%
	createthread	44.00000	44.00000	44.00000	0.00000	0.00000	0.00%
	ctshortestpath	3.00000	3.00000	3.00000	0.00000	0.00000	0.00%
	callbackcount	40.00000	37.00000	37.50000	0.75000	0.86603	2.31%
	cbaveragesize	623.00000	564.00000	613.41667	251.90972	15.87166	2.59%
	cblargestsize	2308.00000	1440.00000	2005.08333	137684.90972	371.05917	18.51%
Dexter	getprocaddress	4.00000	4.00000	4.00000	0.00000	0.00000	0.00%
	memallocation	13.00000	12.00000	12.08333	0.07639	0.27639	2.29%
	createthread	5.00000	5.00000	5.00000	0.00000	0.00000	0.00%
	ctshortestpath	0.00000	0.00000	0.00000	0.00000	0.00000	
	callbackcount	5.00000	5.00000	5.00000	0.00000	0.00000	0.00%
	cbaveragesize	272.00000	107.00000	168.16667	2216.97222	47.08473	28.00%
	cblargestsize	475.00000	291.00000	399.83333	2030.30556	45.05891	11.27%
Winkey	getprocaddress	0.00000	0.00000	0.00000	0.00000	0.00000	
	memallocation	4.00000	3.00000	3.75000	0.18750	0.43301	11.55%
	createthread	0.00000	0.00000	0.00000	0.00000	0.00000	
	ctshortestpath	0.00000	0.00000	0.00000	0.00000	0.00000	
	callbackcount	0.00000	0.00000	0.00000	0.00000	0.00000	
	cbaveragesize	0.00000	0.00000	0.00000	0.00000	0.00000	
	cblargestsize	0.00000	0.00000	0.00000	0.00000	0.00000	

String character frequency histogram

		largest	smallest	average	variance	std deviation	rel std dev
Rbot	str001	11.00000	11.00000	11.00000	0.00000	0.00000	0.00%
	str002	132.00000	126.00000	128.50000	2.75000	1.65831	1.29%
	str003	136.00000	117.00000	122.75000	59.85417	7.73655	6.30%
	str004	239.00000	207.00000	215.50000	185.41667	13.61678	6.32%
	str005	250.00000	220.00000	227.66667	166.55556	12.90564	5.67%
	str006	118.00000	109.00000	111.25000	15.18750	3.89711	3.50%
	str007	106.00000	105.00000	105.25000	0.18750	0.43301	0.41%
	str008	185.00000	183.00000	183.16667	0.30556	0.55277	0.30%
	str009	170.00000	169.00000	169.91667	0.07639	0.27639	0.16%
Dexter	str001	4.00000	4.00000	4.00000	0.00000	0.00000	0.00%
	str002	13.00000	13.00000	13.00000	0.00000	0.00000	0.00%
	str003	7.00000	7.00000	7.00000	0.00000	0.00000	0.00%
	str004	31.00000	31.00000	31.00000	0.00000	0.00000	0.00%
	str005	14.00000	13.00000	13.91667	0.07639	0.27639	1.99%
	str006	7.00000	7.00000	7.00000	0.00000	0.00000	0.00%
	str007	2.00000	1.00000	1.91667	0.07639	0.27639	14.42%
	str008	13.00000	12.00000	12.91667	0.07639	0.27639	2.14%
	str009	4.00000	4.00000	4.00000	0.00000	0.00000	0.00%
Winkey	str001	0.00000	0.00000	0.00000	0.00000	0.00000	
	str002	0.00000	0.00000	0.00000	0.00000	0.00000	
	str003	1.00000	1.00000	1.00000	0.00000	0.00000	0.00%
	str004	6.00000	5.00000	5.83333	0.13889	0.37268	6.39%
	str005	2.00000	2.00000	2.00000	0.00000	0.00000	0.00%
	str006	0.00000	0.00000	0.00000	0.00000	0.00000	
	str007	2.00000	2.00000	2.00000	0.00000	0.00000	0.00%
	str008	0.00000	0.00000	0.00000	0.00000	0.00000	
	str009	2.00000	1.00000	1.91667	0.07639	0.27639	14.42%

File geometry data more stable than graph attributes

Counting of items superior to size measures

15% estimated to be reasonable abstraction

Cost: parsing time and setup, cost to deploy, cost to maintain

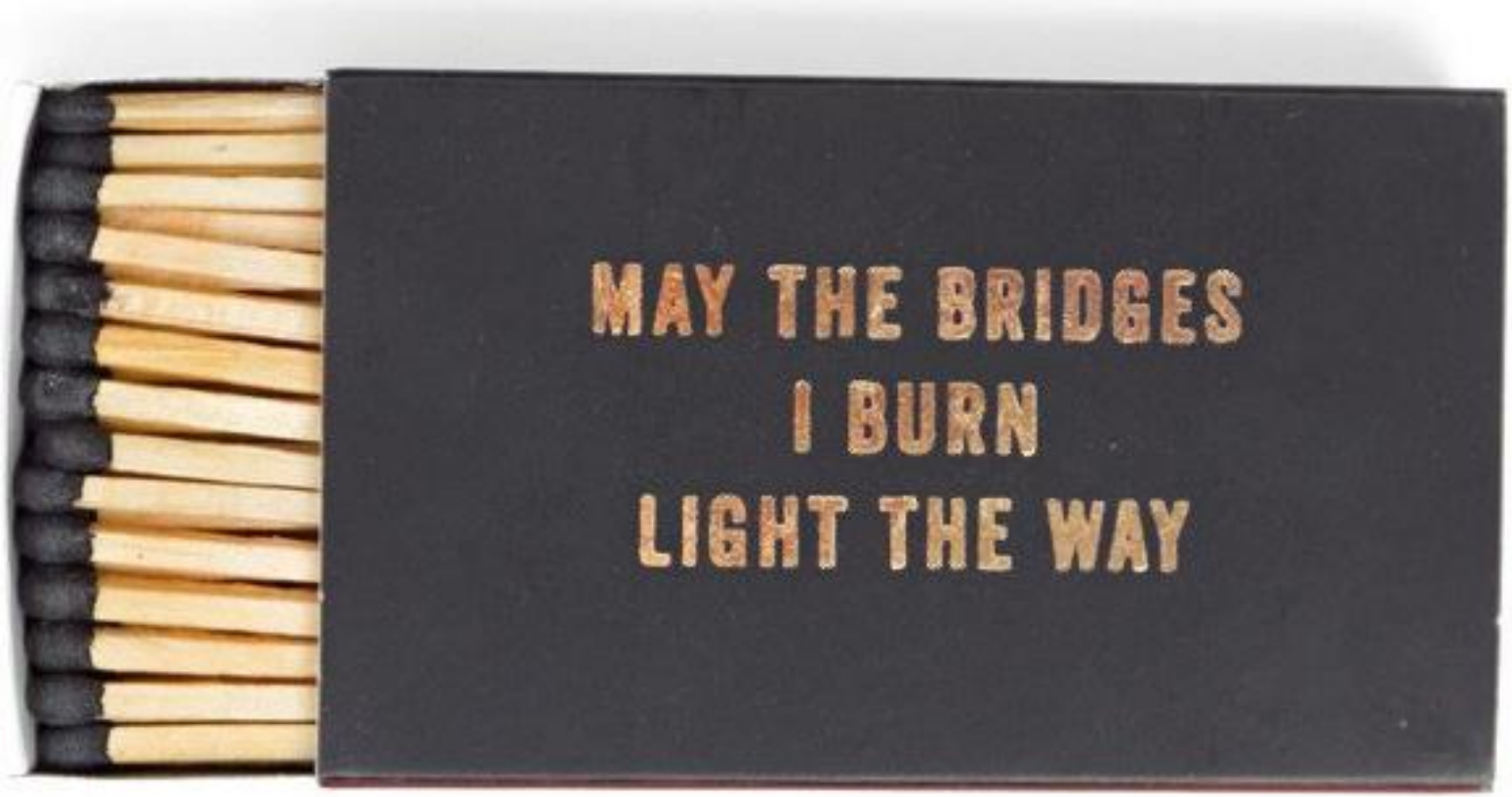
Scalability: ease of feature extraction and adaption

Resilience: robustness against changes in binaries/infrastructure

Reliability: correctness of data

Because f* Threat Intel

MISP
Threat Sharing



Thank you!

@pinkflawd
marion@0x1338.at

http://karmadecay.com/