TINTORERA

Attack Surface Intelligence of Source Code

#HITB2014AMS

De Beurs van Berlage

The 5th Annual
Hack In The Box
Security Conference
in The Netherlands

VULNEX

ME & VULNEX

Simon Roses Femerling

- Founder & CEO, VULNEX www.vulnex.com
-  @simonroses
- Former Microsoft, PwC, @Stake
- Black Hat, RSA, OWASP, SOURCE, AppSec, DeepSec, TECHNET

VULNEX

- CyberSecurity Startup
-  @vulnexsl
- Services & Training
- Products: BinSecSweeper (Binary Analysis)

VULNEX

TALK OBJECTIVES

- GCC & Python, hand to hand
- Transformations: source code to useful data
- Practical code understanding

WORK IN PROGRESS

VULNEX

AGENDA

1. The need of Attack Surface Intelligence of Source Code
2. GCC Overview
3. GCC-Python-Plugin
4. Source Code Intelligence
5. Tintorera Overview
6. Tintorera Analysis Demos
7. Conclusions
8. Q&A

1. The need of Attack Surface Intelligence of Source Code

VULNEX

1. CODE IS GETTING COMPLEX!

Software	SLOC
Firefox	14 Million
Windows Server 2003	50 Million
Debian 7.0	419 Million
Mac OS X 10.4	86 Million
Linux Kernel 2.6.25	13.5 Million
Linux Kernel 3.6	15.9 Million

1. DOCUMENTATION

1. TYPICAL CODE REVIEW

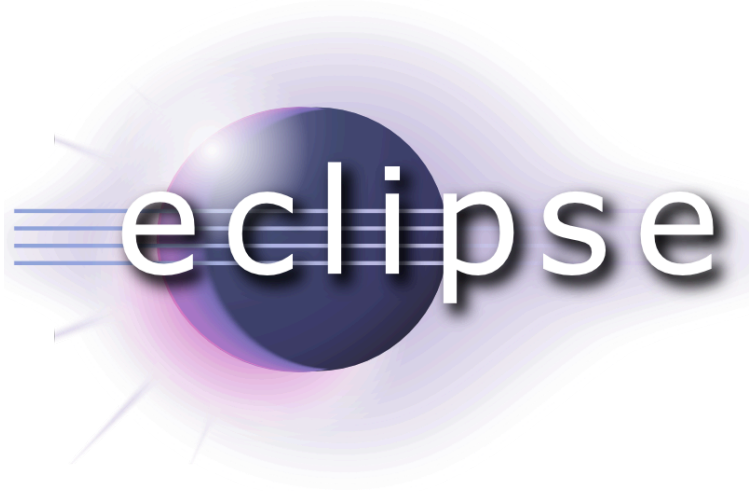
VULNEX

1. WHERE TO START?

- File operations
- Networking
- Processes
- Crypto
- Authentication
- ??

VULNEX

1. TOOLS?



VULNEX

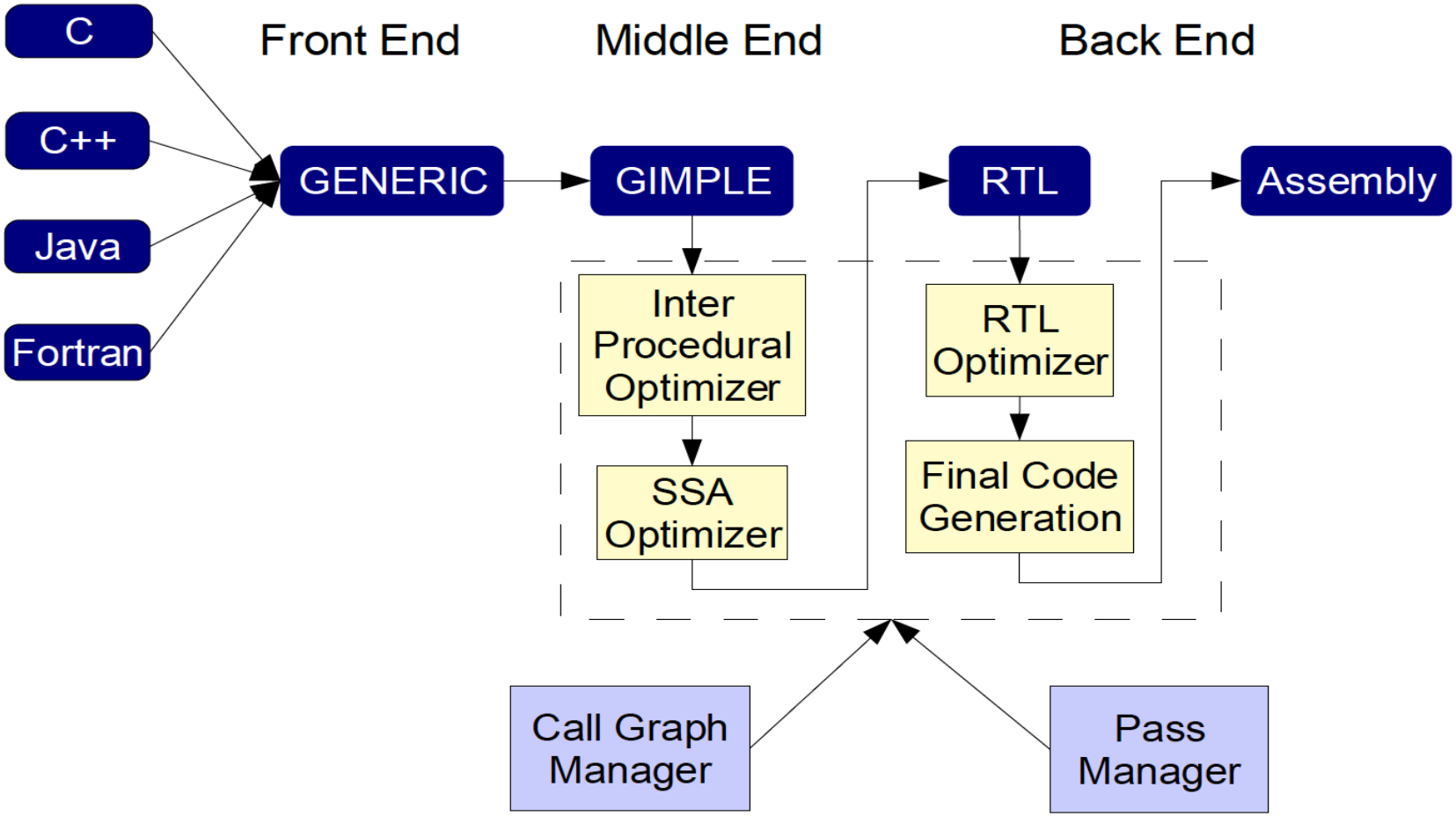
2. GCC Overview

VULNEX

2. GCC

- Compiler system that supports various programming languages
- Popular UNIX variants
- Supports all major languages: C, C++, Java, Objective-C, etc.
- PLUGINS!!
- FREE

2. GCC INTERNALS



http://www.airs.com/dnovillo/Papers/cgo2007-gcc-internals.pdf

2. GCC TERMINOLOGY

- GENERIC is common representation shared by all front ends
  - Each parser must emit GENERIC
- GIMPLE is a simplified version of GENERIC
  - 3 address representation
  - Simplified control flow
- RTL (Register Transfer Language), assembler for an abstract machine

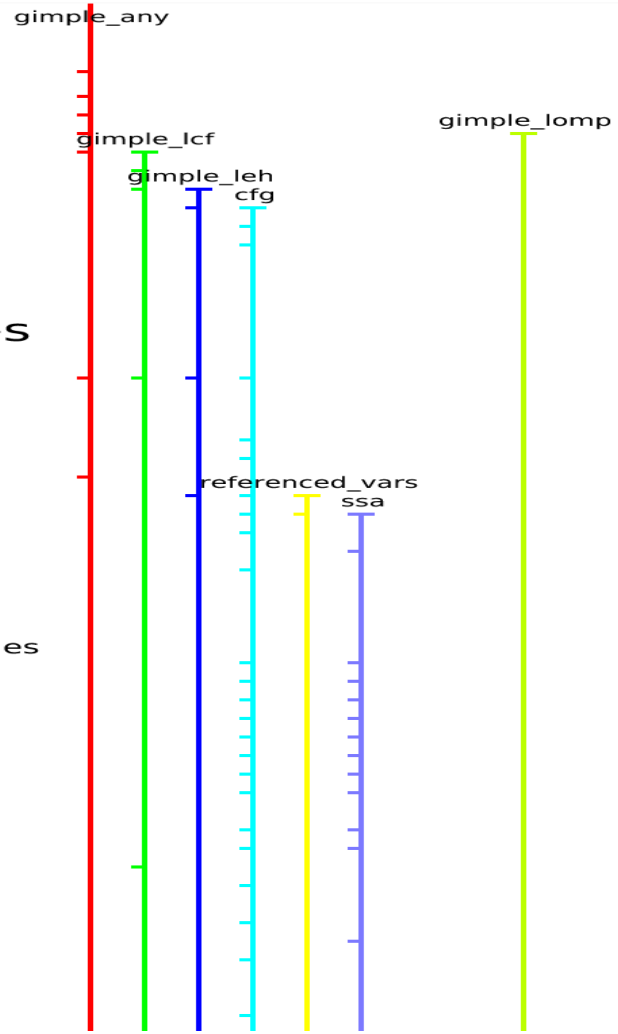
2. GCC PASSES

The lowering passes

*warn_unused_result
*diagnose_omp_blocks
mudflap1
omplower
lower
ehopt
eh
cfg
*warn_function_return
*build_cgraph_edges

The "small IPA" passes

*free_lang_data
visibility
early_local_cleanups
    *free_cfg_annotations
    *init_datastructures
    ompexp
    *referenced_vars
    ssa
    veclower
    *early_warn_uninitialized
    *rebuild_cgraph_edges
    inline_param
    einline
    early_optimizations
        *remove_cgraph_callee_edges
        copyrename
        ccp
        forwprop
        ealias
        esra
        copyprop
        mergephi
        cddce
        eipa_sra
        tailr
        switchconv
        ehcleanup
        profile
        local-pure-const
        fnsplit
    release_ssa
    *rebuild_cgraph_edges
    inline_param
tree_profile_ipa
    feedback_fnsplit



http://gcc-python-plugin.readthedocs.org/en/latest/tables-of-passes.html

VULNEX

3. GCC-Python-Plugin

3. GCC-PYTHON-PLUGIN

- GCC plugin that embeds Python in GCC ☺
- Now your Python script can access GCC passes and perform analysis
- Developed by David Malcolm (Fedora)

http://gcc-python-plugin.readthedocs.org/en/latest/

3. GCC-PYTHON-PLUGIN EXAMPLE

```
1  #   Copyright 2011 David Malcolm <dmalcolm@redhat.com>
2  #   Copyright 2011 Red Hat, Inc.
3  #
4  #   This is free software: you can redistribute it and/or modify it
5  #   under the terms of the GNU General Public License as published by
6  #   the Free Software Foundation, either version 3 of the License, or
7  #   (at your option) any later version.
8  #
9  #   This program is distributed in the hope that it will be useful, but
10 #   WITHOUT ANY WARRANTY; without even the implied warranty of
11 #   MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.  See the GNU
12 #   General Public License for more details.
13 #
14 #   You should have received a copy of the GNU General Public License
15 #   along with this program.  If not, see
16 #   <http://www.gnu.org/licenses/>.
17
18 # Sample python script, to be run by our gcc plugin
19 # Show all the passes that get executed
20 import gcc
21
22 def my_pass_execution_callback(*args, **kwargs):
23     (optpass, fun) = args
24     print(args)
25
26 gcc.register_callback(gcc.PLUGIN_PASS_EXECUTION,
27                       my_pass_execution_callback)
```

VULNEX

3. GCC-PYTHON-PLUGIN DEMO

3. GCC-PYTHON-PLUGIN IDEAS

- Write scripts for:
  - malloc/free usage
  - Array boundary checks
  - Code visualizations
  - You name it!

4. Source Code Intelligence

VULNEX

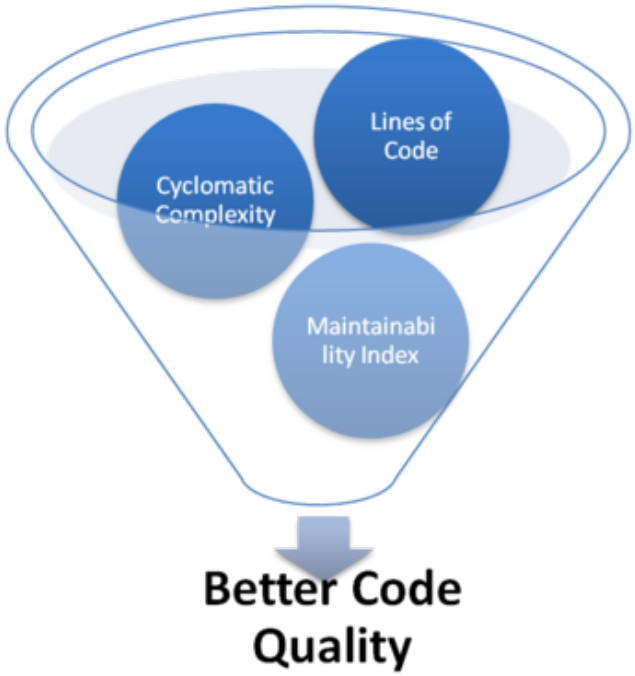
4. CODE UNDERSTATING

- What API are being used?
- Number of functions?
- Inputs / Outputs of functions?
- Function relationship
- What comments said?
- Code complexity

4. CODE METRICS

- Controversial topic but needed

- Metrics:
  - Function complexity (Cyclomatic)
  - Number of:
    - Lines
    - Code
    - Blanks
    - Comments
  - Line Length
  - Number: Bugs per Line
  - You name it….



4. CODE COMPLEXITY

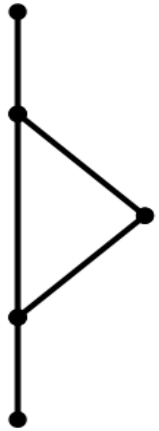
- Counts the number of linearly independent paths through the source code
- Basically we can have an idea of the complexity of functions
- Complexity is security enemy!
- Created by Thomas McCabe
  http://www.literateprogramming.com/mccabe.pdf

4. CODE COMPLEXITY THRESHOLD

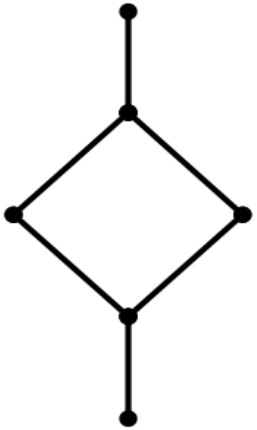
Cyclomatic Complexity	Risk Evaluation
1-10	a simple program, without much risk
11-20	more complex, moderate risk
21-50	complex, high risk program
greater than 50	untestable program (very high risk)

http://www.sei.cmu.edu/reports/97hb001.pdf

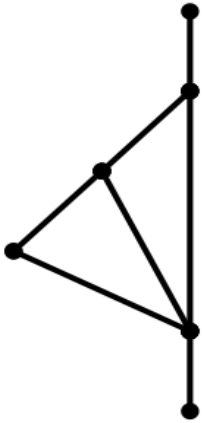
4. SOURCE CODE ANALYSIS FLOWGRAPH NOTATION



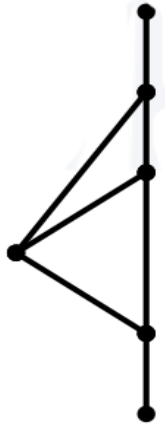
If .. then



If .. then .. else

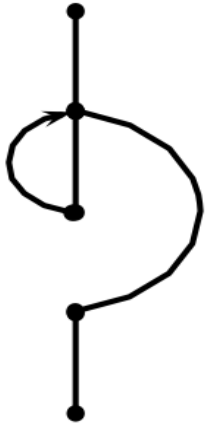


If .. and .. then

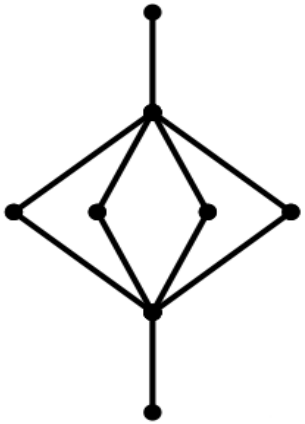


If .. or .. then

Do .. While

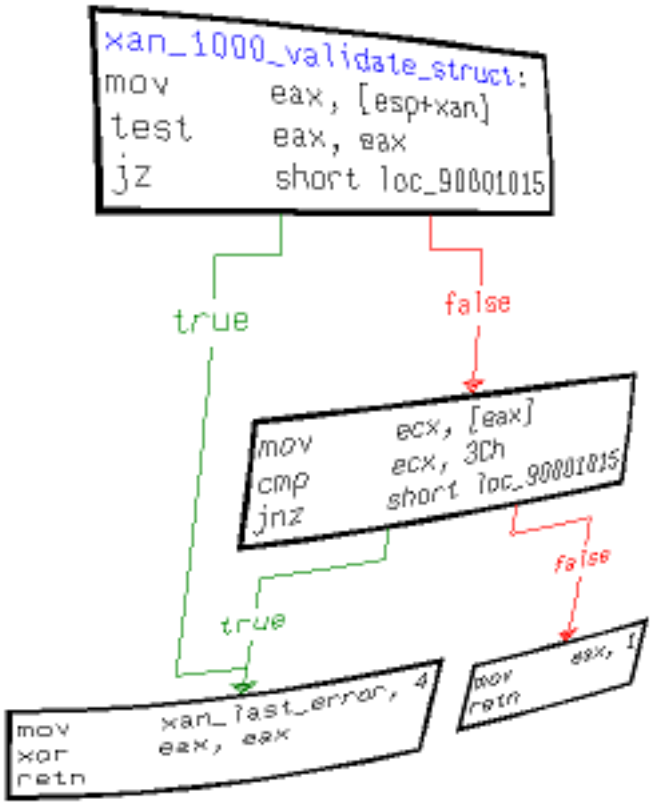


While .. Do

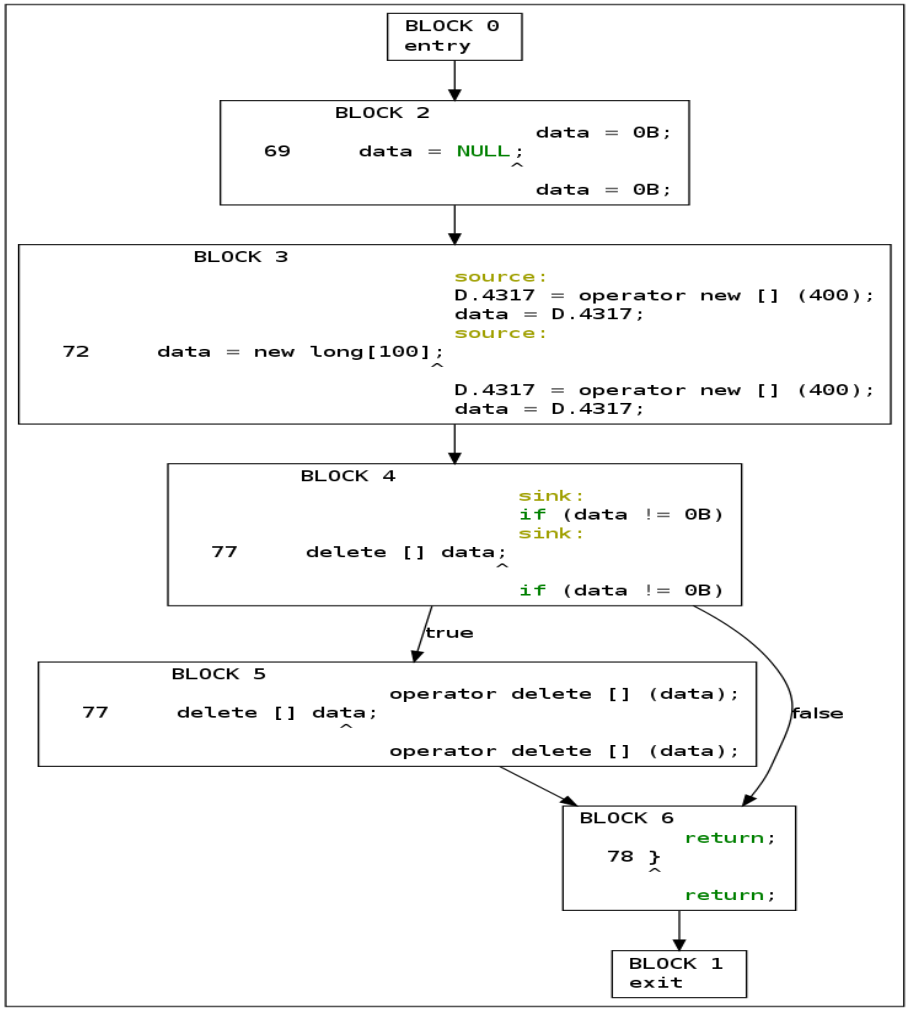


Switch

4. SOURCE CODE VISUALS TOO



BINARY



SOURCE CODE

VULNEX

5. Tintorera Overview

5. TINTORERA – BLUE SHARK

- "Put source code into context"
- Objective: Get a feeling of the code while compiling!!
- Intelligence of source code:
  - Code visualizations
  - Comments analysis
  - API identification
  - Metrics
  - HTML Reports
- C code transformed to JSON files, now you can query and perform analysis on data

5. TINTORERA INTERNALS

- Two files:
  - analyzer.py: To be used while compiling a project
  - do_report_tintorera.py: Use after project has been compiled to generate report
- Composed of:
  - Python code
  - JSON data files
  - HTML / CSS / Javascript

5. TINTORERA STRUCTURE

- Python files
- Folders:
  - data/ : API JSON file
  - templates/ : HTML templates
  - js/ : Javascript code
  - images/
  - Tintorera_lib/ : python code

5. TINTORERA INSTALL & USAGE

1. GCC version 4.7 or later
2. Install gcc-python-plugin (See web doc)
3. Set path:
    1. Export LD_LIBRARY_PATH=/gcc-python-plugin/gcc-c-api
4. Add line to Makefile (CC= tag)
    1. gcc –fplugin=/gcc-python-plugin/python.so –fplugin-arg-python-script=/tintorera/analyzer.py
5. Run make
6. After compile use:
    1. Python do_report_tintorera.py –c tinan.cfg

5. TINTORERA CONFIG FILE

- Edit tinan.cfg to suit your needs

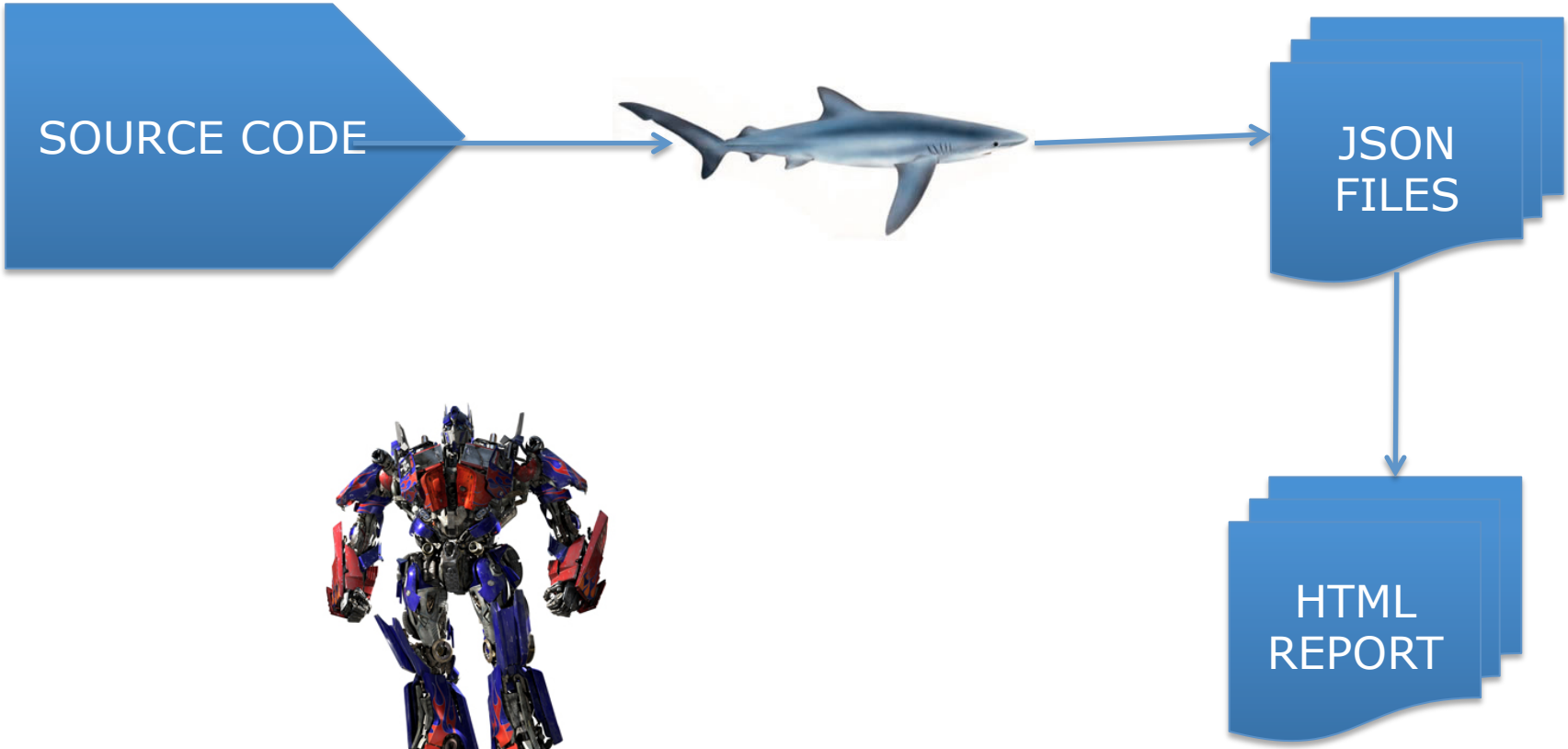
- Set parameters such as:
  - Folder to save analysis report
  - Enable / disable analysis
    - Basic blocks
    - Callgraphs
    - Comments
    - Gimples
    - Etc.
  - Cyclomatic Thresholds

5. TINTORERA DATA FILES

- Folder: /data
- File: tinto_api.json

- JSON file to define APIs

5. CODE TRANSFORMATION



VULNEX

5. TRANSFORMED JSON FILES

- 3 files:

1. tintorera_bb_file.json: code basic blocks
2. tintorera_meta_info.json: general information, file size and code & comments not inside functions
3. tintorera_temp_file.json: functions information

5. TINTORERA_BB_FILE.JSON


```
[
    {
        "file_name": "loop_tester.c",
        "func_gimples": [
            {
                "bb_data": "gcc.BasicBlock(index=0)",
                "bb_0": 0,
                "desc": "entry"
            },
            {
                "bb_1": 1,
                "bb_data": "gcc.BasicBlock(index=1)",
                "desc": "exit"
            },
            {
                "bb_2": 2,
                "bb_data": [
                    {
                        "loc": "loop_tester.c:132",
                        "code": "    no_loop();",
                        "repr(stmt)": "gcc.GimpleCall()",
                        "exprcode": "<type 'gcc.CallExpr'>",
                        "str(stmt)": "no_loop ();",
                        "rhs": "[<gcc.AddrExpr object at 0xaf024e8>, None]",
                        "lhs": "None",
                        "exprtype": "<gcc.VoidType object at 0xaf02c98>"
                    },
                    {
                        "loc": "loop_tester.c:133",
                        "code": "    if_stmt();",
                        "repr(stmt)": "gcc.GimpleCall()",
                        "exprcode": "<type 'gcc.CallExpr'>",
                        "str(stmt)": "if_stmt ();",
                        "rhs": "[<gcc.AddrExpr object at 0xaf02518>, None]",
                        "lhs": "None",
                        "exprtype": "<gcc.VoidType object at 0xaf02c98>"
                    },
                    {
                        "loc": "loop_tester.c:134"
```


VULNEX

5. TINTORERA_META_FILE.JSON


```
{
    "file_name": "loop_tester.c",
    "file_size": 1541,
    "file_comments": [],
    "file_comments_len": 0,
    "file_lines_len": [
        {
            "line": 0,
            "len": 19
        },
        {
            "line": 1,
            "len": 1
        },
        {
            "line": 2,
            "len": 19
        }
    ],
    "file_ploc": 3,
    "file_loc": 2,
    "file_blank": 1,
    "file_metrics": {
        "total_code": 127,
        "line_max": "38",
        "line_min": "1",
        "total_bb": 74,
        "total_cc": 27,
        "cc_avg": 2,
        "total_funcs": 13,
        "total_lines": 136,
        "line_avg": "11.4666666667",
        "total_blanks": 8,
        "total_comments": 0
    },
    "file_sha256": "",
    "file_code": [
        "#include <stdio.h>\n",
        "\n",
        "void no_loop(void)\n"
```


VULNEX

5. TINTORERA_TEMP_FILE.JSON


```
{
    "func_inline_asm": "",
    "file_name": "loop_tester.c",
    "func_api_res": [],
    "func_loc": 16,
    "func_code": [
        "int main(int *argc, char *argv[]) \n",
        "{\n",
        "  \n",
        "     no_loop();\n",
        "     if_stmt();\n",
        "     if_else_stmt();\n",
        "     if_and_stmt();\n",
        "     if_or_stmt();\n",
        "     if_and_else_stmt();\n",
        "     if_or_else_stmt();\n",
        "     while_stmt();\n",
        "     for_stmt(); \n",
        "     do_while_stmt();\n",
        "     switch_stmt();\t\n",
        "     goto_stmt();\n",
        "\n",
        "     return 0;\n",
        "}\n"
    ],
    "func_comments_len": 0,
    "func_ploc": 18,
    "func_count_gimples": {
        "gimplecall": 12,
        "gimpleasm": 0,
        "gimplereturn": 1,
        "gimplenop": 0,
        "gimplephi": 0,
        "gimplecond": 0,
        "gimpleswitch": 0,
        "gimpleassign": 1,
        "gimplelabel": 1
    },
    "func_end_line": 147,
    "func_decl": {
```


VULNEX

5. TINTORERA SOURCE CODE METRICS

- Current metrics:

1. Number of:
    1. Lines
    2. Code
    3. Blanks
    4. Comments
    5. Colons
2. Average line length
3. Minimum line
4. Maximum line
5. Total Basic Blocks
6. Total Cyclomatic Complexity
7. Average Cyclomatic Complexity

5. SOURCE CODE COMMENT ANALYSIS

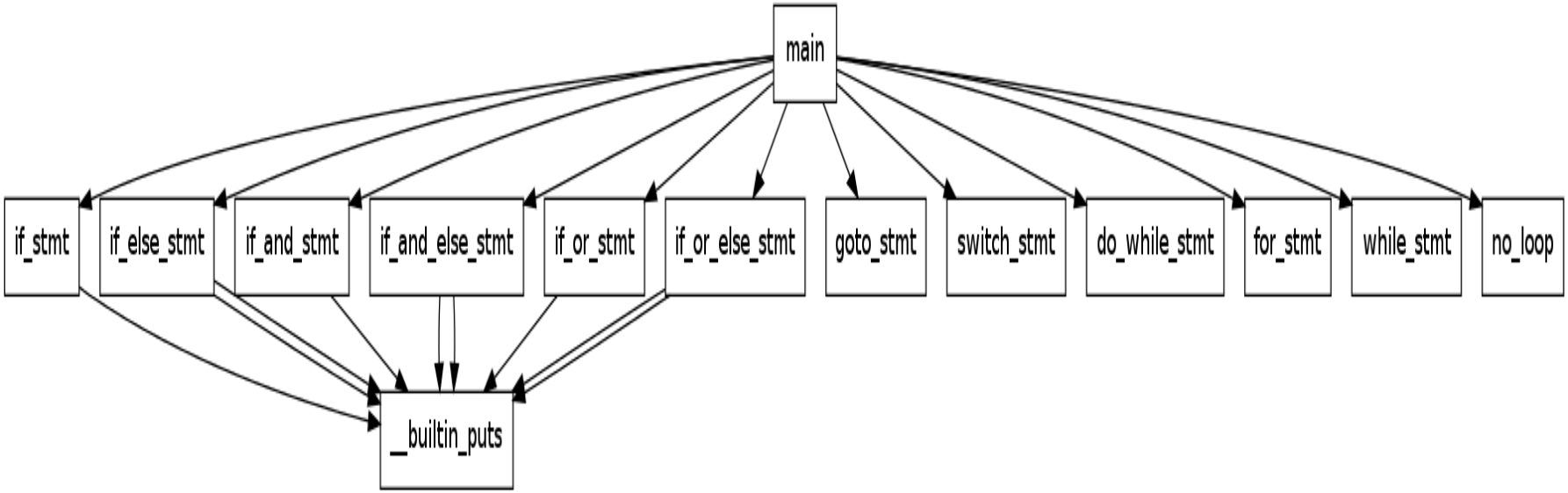
VULNEX

6. Tintorera Analysis Demos

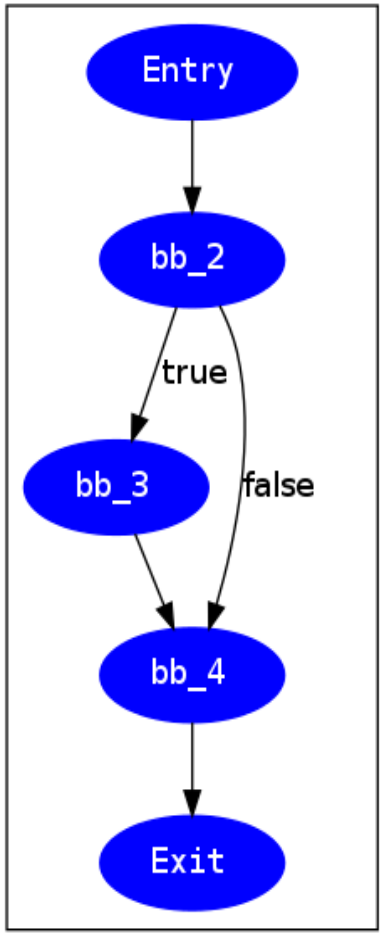
VULNEX

6. DEMO I: LOOP TESTER

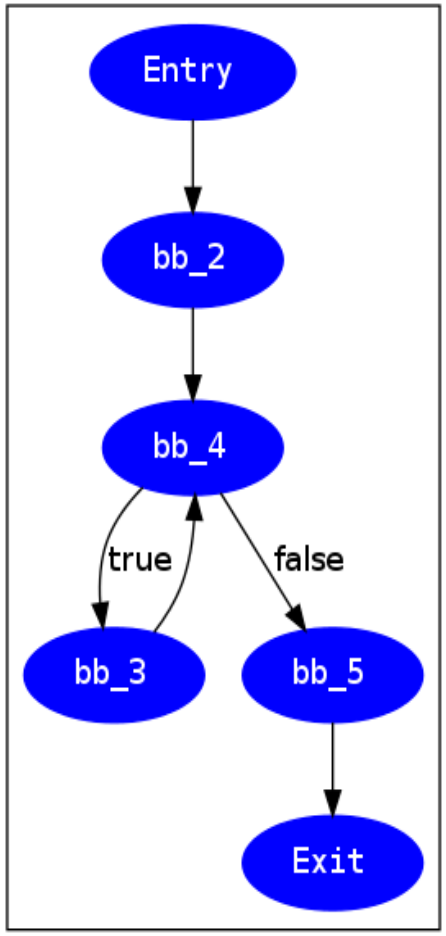
6. DEMO I: LOOP TESTER



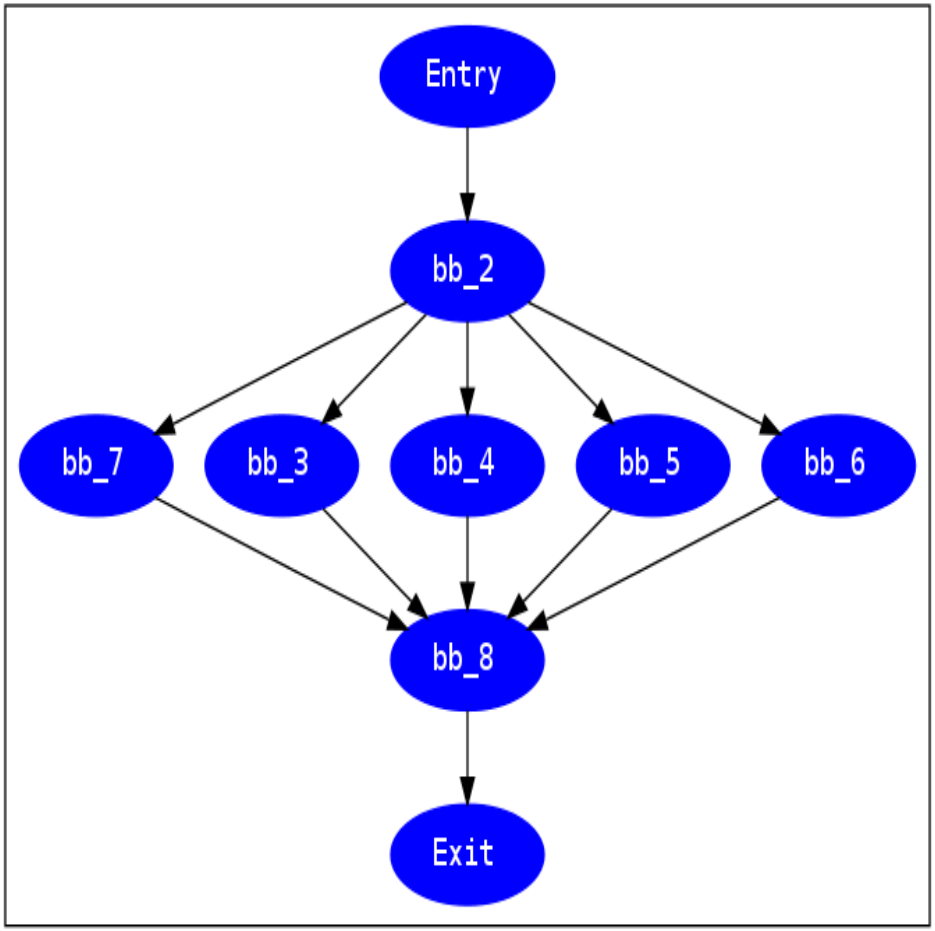
6. DEMO I: LOOP TESTER



IF ELSE



WHILE



SWITCH

VULNEX

6. DEMO II: SENDMAIL CRACKADDR (CVE2002-1337)

Pure Complexity….

6. DEMO II: SENDMAIL CRACKADDR (CVE2002-1337) FUNCTION COMPLEXITY

Function = sendmail_crackaddr_cve2002_1337.c -> crackaddr

<<

Function Details

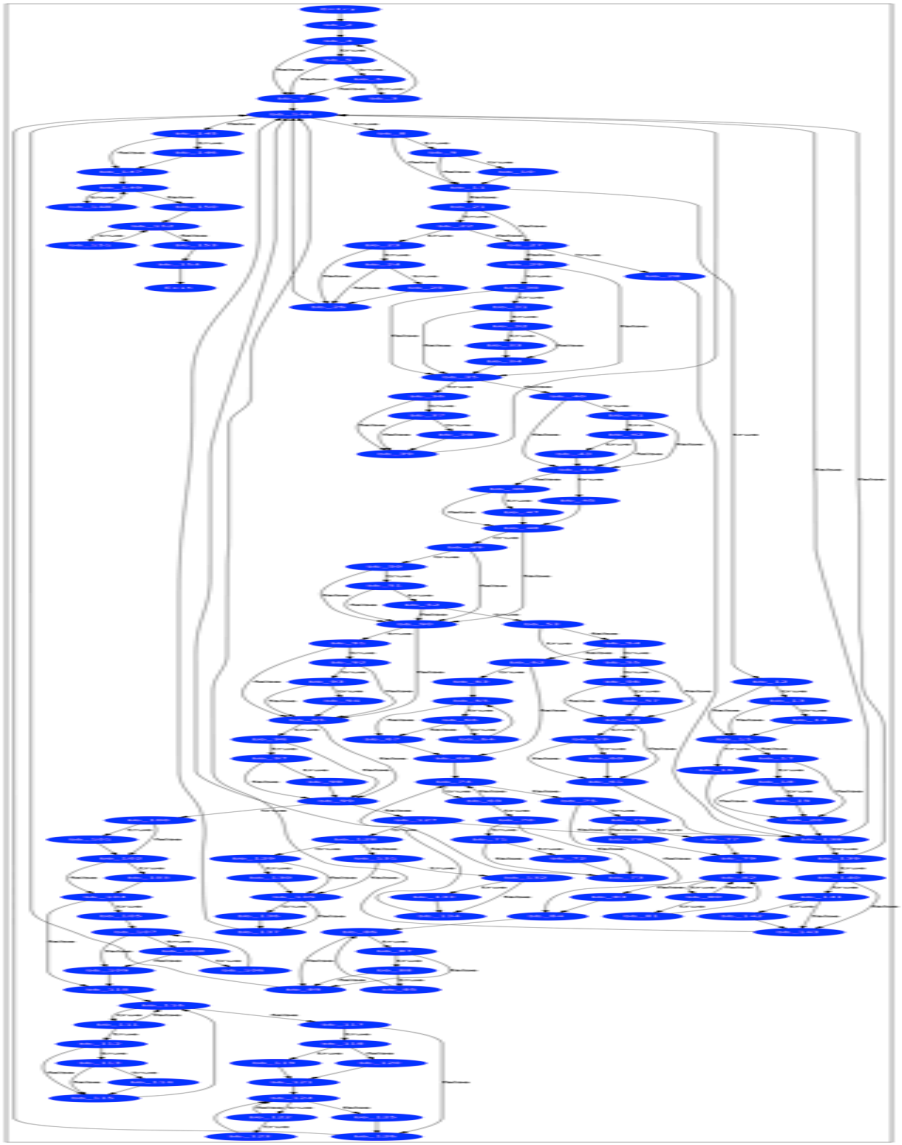
Function Name	crackaddr
Arguments (1)	[u'char *']
Return Type	char *
Function LOC	247
Function Physical LOC	334
Function Start Line	56
Function End Line	390
Comments	54
Blank Lines	33

Code Details & Metrics

Basic Blocks	155
Code Complexity	89
Metrics-> Count Colons	125

VULNEX

6. DEMO II: SENDMAIL CRACKADDR (CVE2002-1337) FUNCTION COMPLEXITY



VULNEX

6. DEMO III: MONGOOSE WEB SERVER ANALYSIS

- Mongoose is the most easy to use web server on the planet. A web server of choice for Web developers (PHP, Ruby, Python, etc) and Web designers.

6. DEMO III: MONGOOSE WEB SERVER ANALYSIS

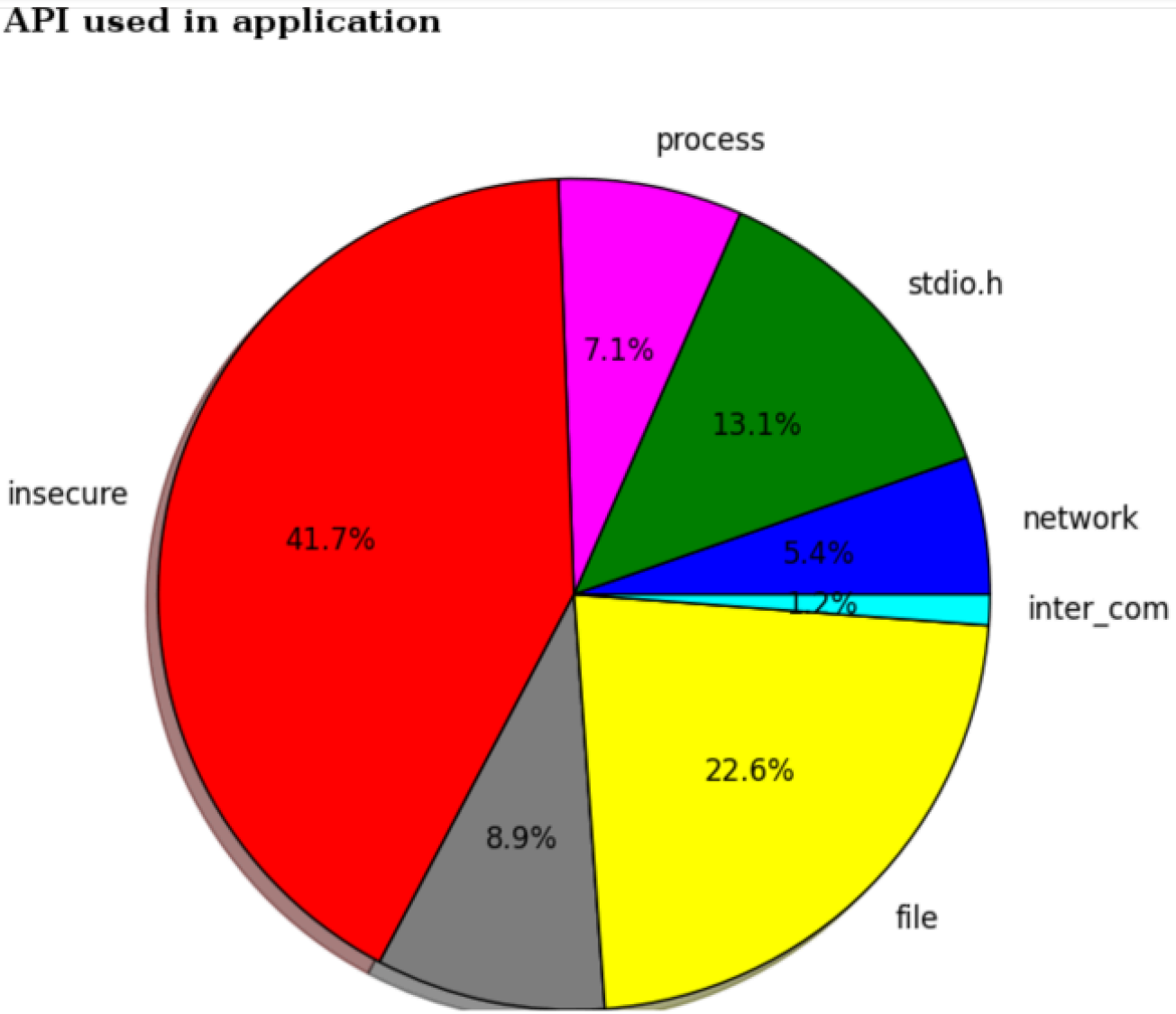
Application Summary

<<

Application Details

Total Files	2
Total Functions	156
Total Basic Blocks	1930
Total LOC	3507
Total Physical LOC	4306
Total Comments	337
Total Blanks	460

6. DEMO III: MONGOOSE WEB SERVER ANALYSIS



VULNEX

6. DEMO III: MONGOOSE WEB SERVER ANALYSIS

Files Results

<<

# All Funcs	# File Funcs	File Name	Function Name	Basic Blocks	Cyclomatic Complexity	API Calls	Inline ASM
1	1	main.c	main	7	2	YES	
2	2	main.c	start_mongoose	17	9	YES	
3	3	main.c	mongoose_callback	6	2		
4	4	main.c	init_server_name	3	1	YES	
5	5	main.c	process_command_line_arguments	31	15	YES	
6	6	main.c	set_option	13	6		
7	7	main.c	sdup	6	2	YES	
8	8	main.c	verify_document_root	9	5	YES	
9	9	main.c	show_usage_and_exit	9	3	YES	
10	10	main.c	die	3	1	YES	
11	11	main.c	signal_handler	3	1		
12	1	mongoose.c	mg_start	34	17	YES	
13	2	mongoose.c	mg_stop	6	2		
14	3	mongoose.c	free_context	14	6		
15	4	mongoose.c	master_thread	19	8		
16	5	mongoose.c	accept_new_connection	7	3	YES	

6. DEMO III: MONGOOSE WEB SERVER ANALYSIS

Function = main.c -> main

<<

Function Details

Function Name	main
Arguments (2)	[u'int', u'char * *']
Return Type	int
Function LOC	16
Function Physical LOC	17
Function Start Line	882
Function End Line	899
Comments	0
Blank Lines	1

Code Details & Metrics

Basic Blocks	7
Code Complexity	2
Metrics-> Count Colons	9

VULNEX

6. DEMO IV: BOA WEB SERVER

Boa, a high performance web server for Unix-alike computers

6. DEMO IV: BOA WEB SERVER

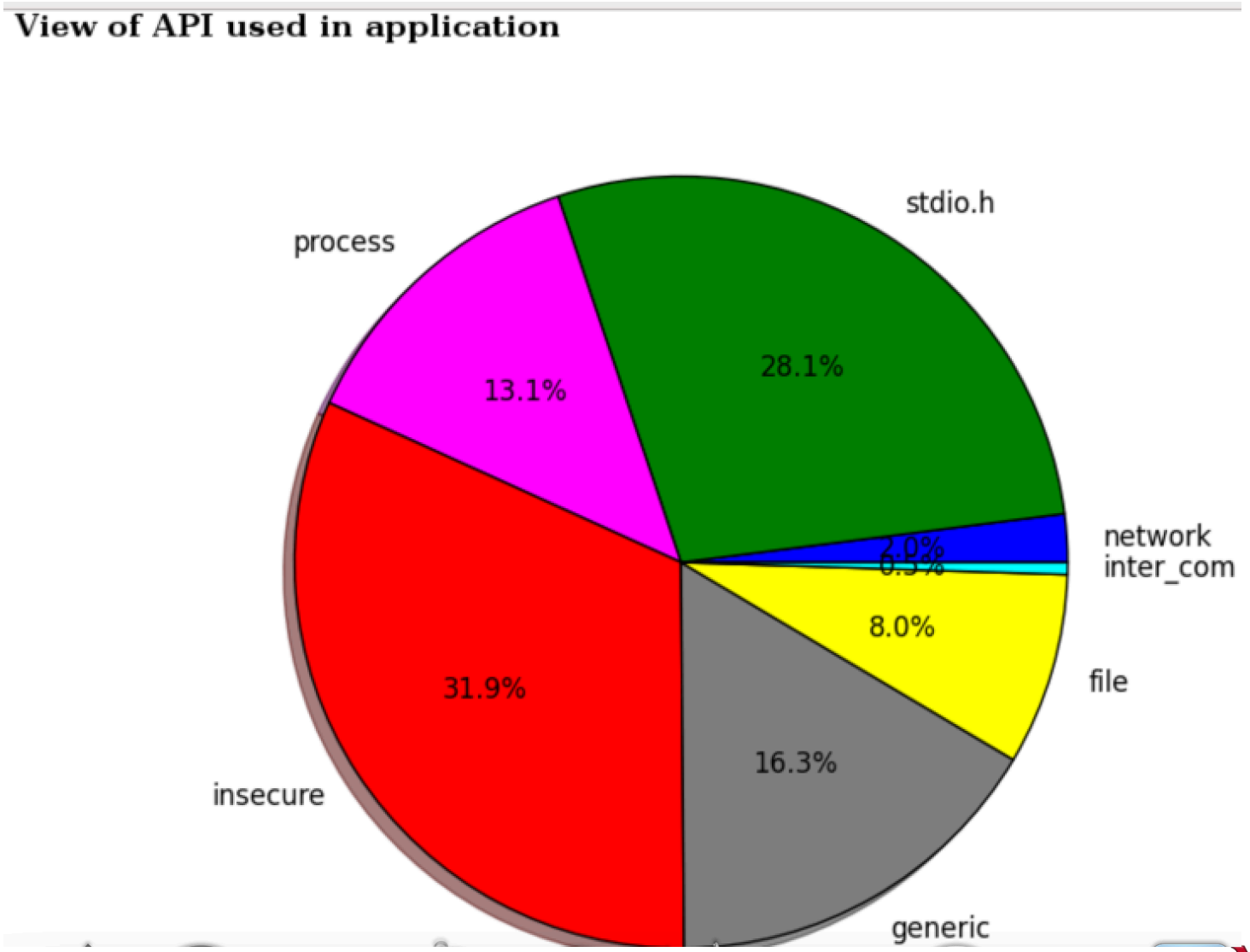
Application Summary

<<

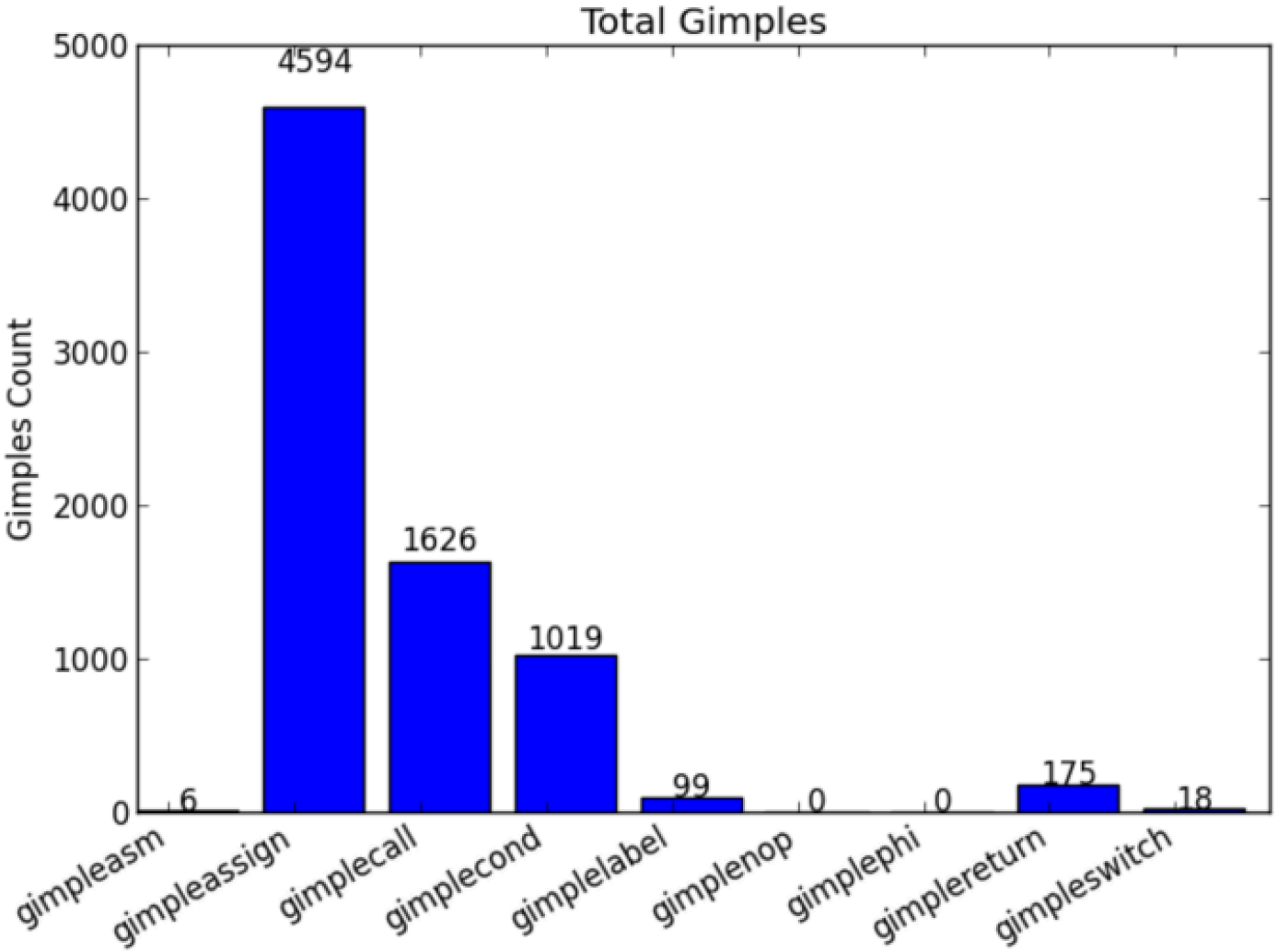
Application Details

Total Files	24
Total Functions	182
Total Basic Blocks	2714
Total LOC	6237
Total Physical LOC	8596
Total Comments	1470
Total Blanks	865

6. DEMO IV: BOA WEB SERVER



6. DEMO IV: BOA WEB SERVER



VULNEX

6. DEMO IV: BOA WEB SERVER

Files Results

<<

# All Funcs	# File Funcs	File Name	Function Name	Basic Blocks	Cyclomatic Complexity	API Calls	Inline ASM
1	1	index_dir.c	main	11	4	YES	
2	2	index_dir.c	index_directory	58	26	YES	
3	3	index_dir.c	stat	3	1		
4	4	index_dir.c	select_files	6	2		
5	5	index_dir.c	send_error	11	2	YES	
6	6	index_dir.c	http_escape_string	24	11		
7	7	index_dir.c	html_escape_string	16	5		
8	1	timestamp.c	timestamp	3	1	YES	
9	1	select.c	loop	42	23	YES	YES
10	2	select.c	fdset_update	40	21		
11	1	sublog.c	open_gen_fd	8	3	YES	
12	2	sublog.c	open_net_fd	15	6	YES	YES
13	3	sublog.c	open_pipe_fd	12	5	YES	
14	1	util.c	parse_debug	19	9	YES	
15	2	util.c	print_debug_usage	6	2	YES	
16	3	util.c	strlower	6	2		

6. DEMO IV: BOA WEB SERVER

Function Start Line	46
Function End Line	135
Comments	21
Blank Lines	10

Code Details & Metrics

Basic Blocks	42
Code Complexity	23
Metrics-> Count Colons	24

API Details

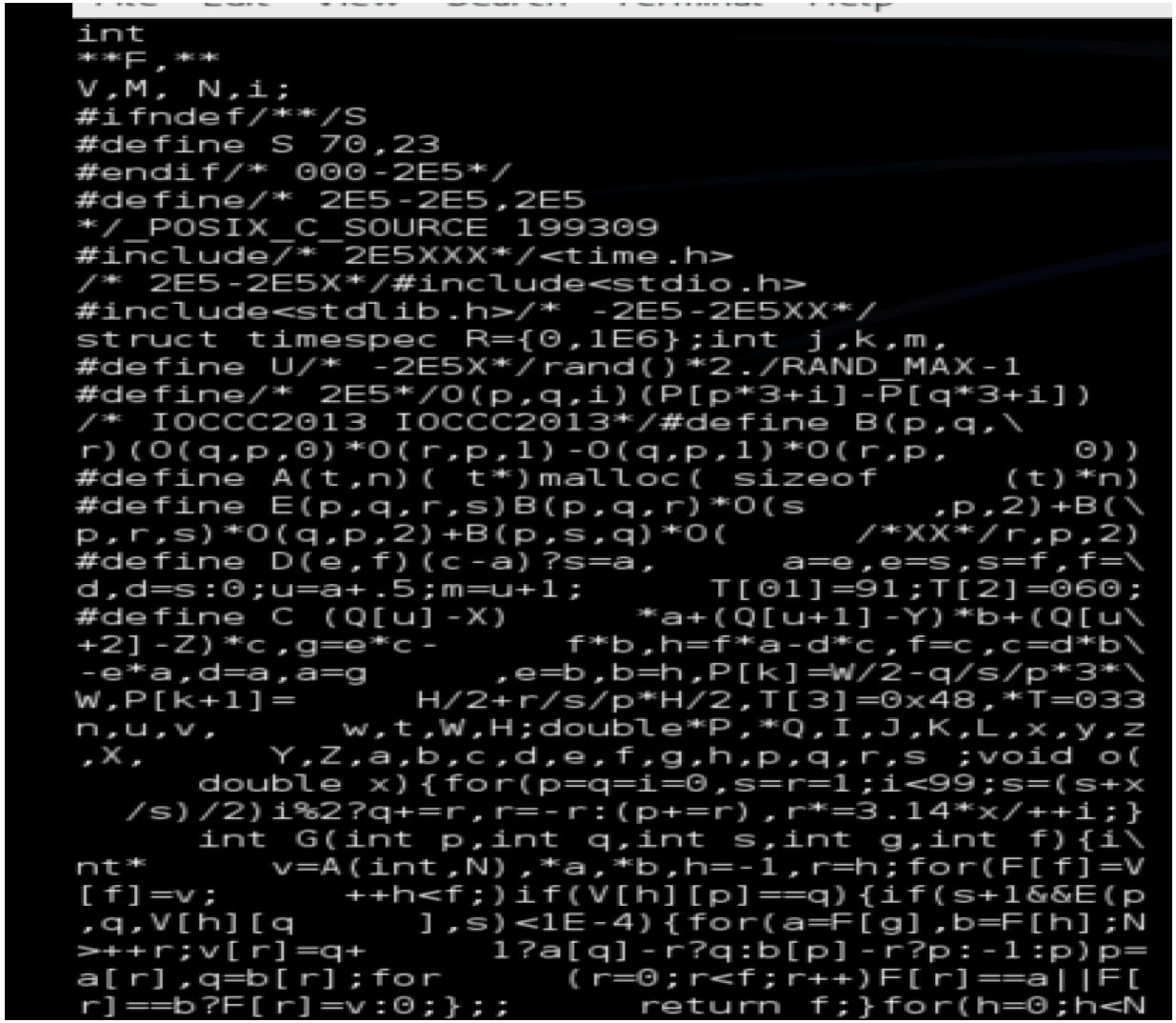
API Calls		
Category	API	LOC
generic	close	select.c:70

Inline ASM Details

Inline ASM	__asm__ __volatile__("cld; rep; stosl" : "=c" __d0, "=D" __d1 : "a" 0, "0" 32, "1" &block_read_fdset.__fds_bits[0] : "memory");

VULNEX

6. DEMO V: OBFUSCATED C CODE ANALYSIS, ENDOH4.C



```
int
**F,**
V,M, N,i;
#ifndef/**/S
#define S 70,23
#endif/* 000-2E5*/
#define/* 2E5-2E5,2E5
*/_POSIX_C_SOURCE 199309
#include/* 2E5XXX*/<time.h>
/* 2E5-2E5X*/#include<stdio.h>
#include<stdlib.h>/* -2E5-2E5XX*/
struct timespec R={0,1E6};int j,k,m,
#define U/* -2E5X*/rand()*2./RAND_MAX-1
#define/* 2E5*/O(p,q,i)(P[p*3+i]-P[q*3+i])
/* IOCCC2013 IOCCC2013*/#define B(p,q,\
r)(O(q,p,0)*O(r,p,1)-O(q,p,1)*O(r,p,      0))
#define A(t,n)( t*)malloc( sizeof      (t)*n)
#define E(p,q,r,s)B(p,q,r)*O(s      ,p,2)+B(\
p,r,s)*O(q,p,2)+B(p,s,q)*O(      /*XX*/r,p,2)
#define D(e,f)(c-a)?s=a,      a=e,e=s,s=f,f=\
d,d=s:0;u=a+.5;m=u+1;      T[01]=91;T[2]=060;
#define C (Q[u]-X)      *a+(Q[u+1]-Y)*b+(Q[u\
+2]-Z)*c,g=e*c-      f*b,h=f*a-d*c,f=c,c=d*b\
-e*a,d=a,a=g      ,e=b,b=h,P[k]=W/2-q/s/p*3*\
W,P[k+1]=      H/2+r/s/p*H/2,T[3]=0x48,*T=033
n,u,v,      w,t,W,H;double*P,*Q,I,J,K,L,x,y,z
,X,      Y,Z,a,b,c,d,e,f,g,h,p,q,r,s ;void o(
      double x){for(p=q=i=0,s=r=1;i<99;s=(s+x
   /s)/2)i%2?q+=r,r=-r:(p+=r),r*=3.14*x/++i;}
      int G(int p,int q,int s,int g,int f){i\
nt*      v=A(int,N),*a,*b,h=-1,r=h;for(F[f]=V
[f]=v;      ++h<f;)if(V[h][p]==q){if(s+1&&E(p
,q,V[h][q      ],s)<1E-4){for(a=F[g],b=F[h];N
>++r;v[r]=q+      1?a[q]-r?q:b[p]-r?p:-1:p)p=
a[r],q=b[r];for      (r=0;r<f;r++)F[r]==a||F[
r]==b?F[r]=v:0;};;      return f;}for(h=0;h<N
```

6. DEMO V: OBFUSCATED C CODE ANALYSIS, ENDOH4.C

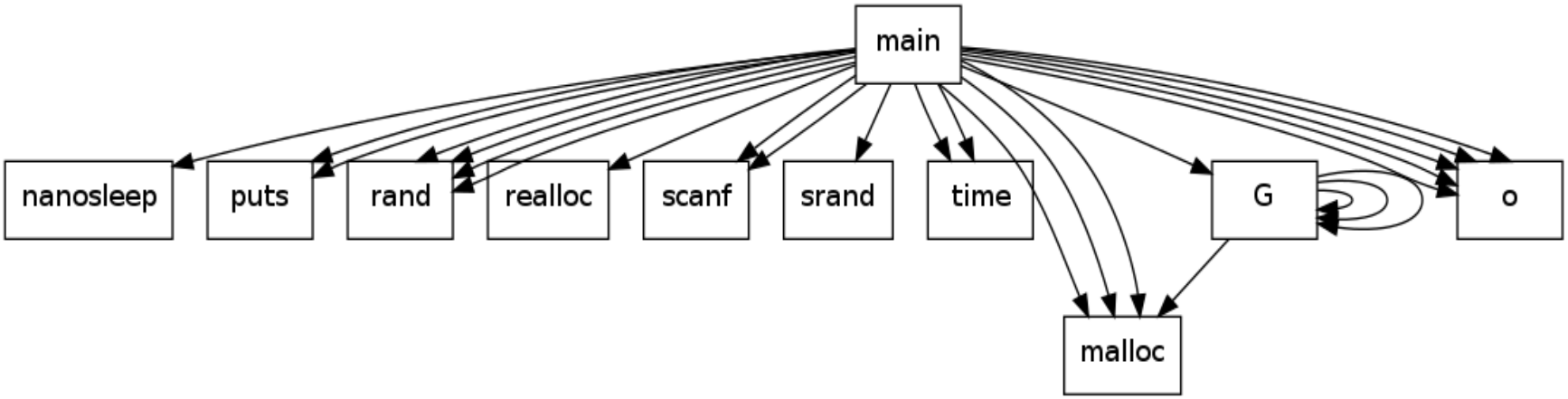
Files Results

<<

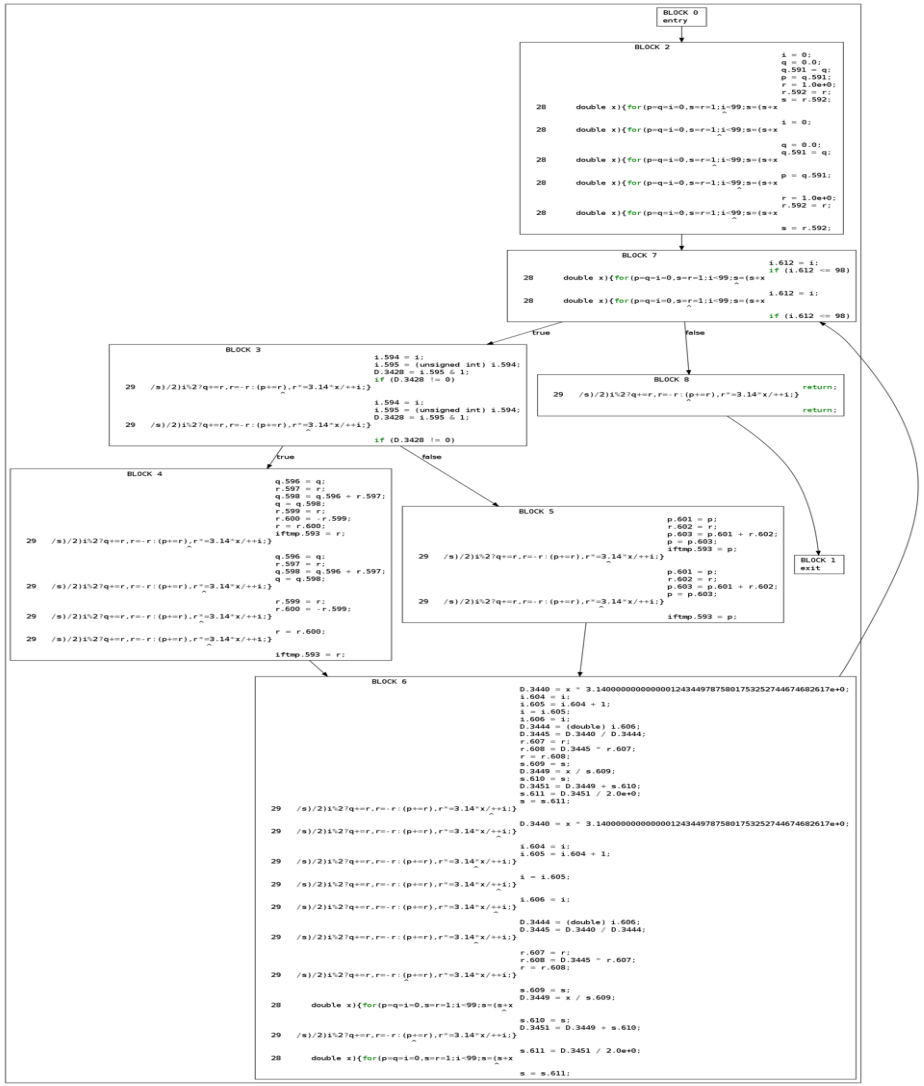
# All Funcs	# File Funcs	File Name	Function Name	Basic Blocks	Cyclomatic Complexity	API Calls	Inline ASM
1	1	endoh4.c	main	93	37	YES	
2	2	endoh4.c	G	38	17		
3	3	endoh4.c	o	9	3		

<<

6. DEMO V: OBFUSCATED C CODE ANALYSIS, ENDOH4.C

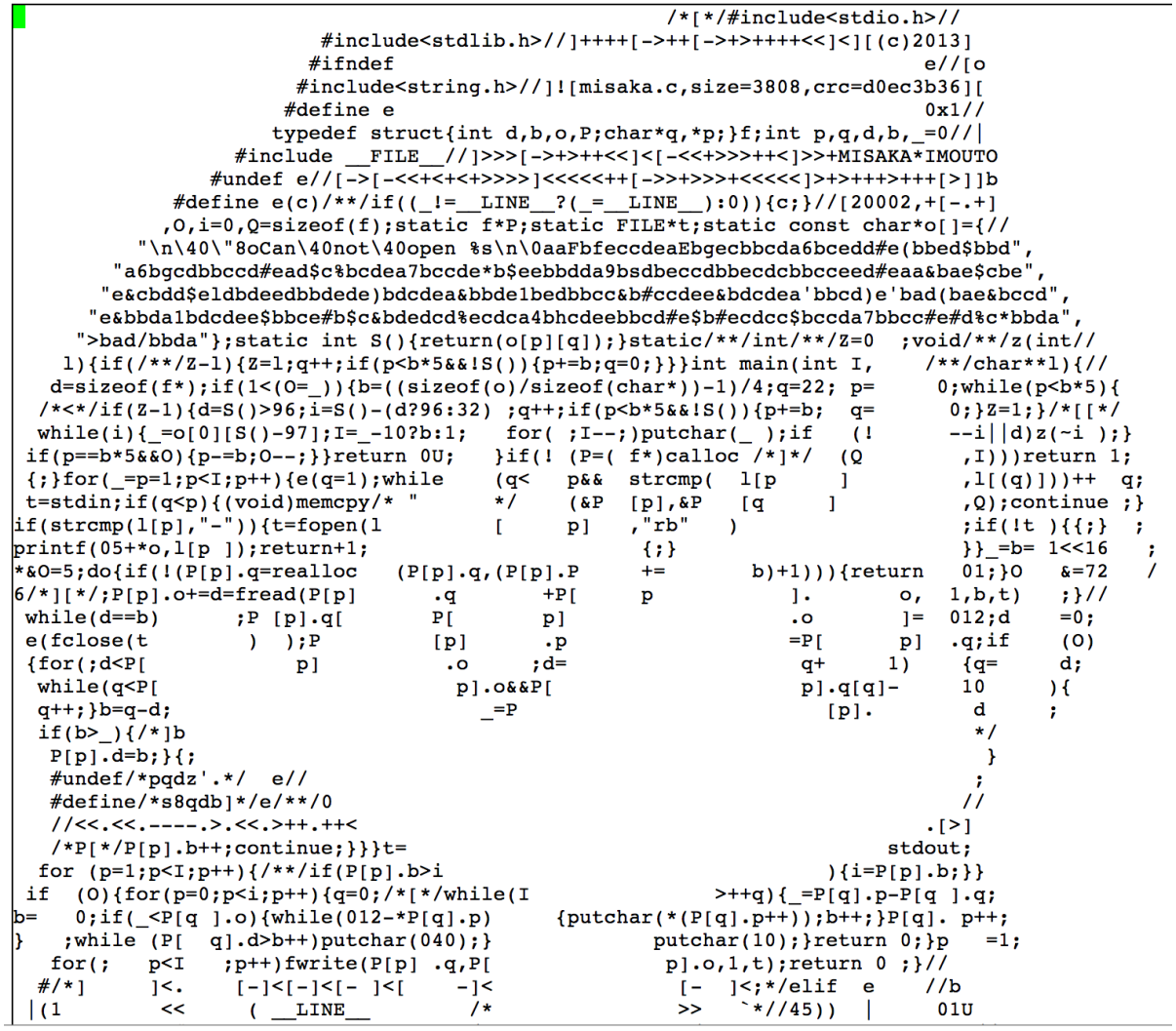


6. DEMO V: OBFUSCATED C CODE ANALYSIS, ENDOH4.C



O function

6. DEMO VI: OBFUSCATED C CODE ANALYSIS, MISAKA



```
                                                  /*[*/#include<stdio.h>//
                        #include<stdlib.h>//]++++[->++[->+>++++<<]<][(c)2013]
                       #ifndef                                          e//[o
                      #include<string.h>//]![misaka.c,size=3808,crc=d0ec3b36][
                     #define e                                          0x1//
                    typedef struct{int d,b,o,P;char*q,*p;}f;int p,q,d,b,_=0//|
                  #include __FILE__//]>>>[->+>+++<<]<[-<<+>>>++<]>>+MISAKA*IMOUTO
                 #undef e//[->[-<<+<+<+>>>>]<<<<<++[->>+>>>+<<<<<]>+>+++>+++[>]]b
               #define e(c)/**/if((_!=__LINE__?(_=__LINE__):0)){c;}//[20002,+[-.+]
              ,O,i=0,Q=sizeof(f);static f*P;static FILE*t;static const char*o[]={//
            "\n\40\"8oCan\40not\40open %s\n\0aaFbfeccdeaEbgecbbcda6bcedd#e(bbed$bbd",
          "a6bgcdbbccd#ead$c%bcdea7bccde*b$eebbdda9bsdbeccdbbecdcbbcceed#eaa&bae$cbe",
         "e&cbdd$eldbdeedbbdede)bdcdea&bbde1bedbbcc&b#ccdee&bdcdea'bbcd)e'bad(bae&bccd",
       "e&bbda1bdcdee$bbce#b$c&bdedcd%ecdca4bhcdeebbcd#e$b#ecdcc$bccda7bbcc#e#d%c*bbda",
      ">bad/bbda"};static int S(){return(o[p][q]);}static/**/int/**/Z=0  ;void/**/z(int//
    l){if(/**/Z-l){Z=l;q++;if(p<b*5&&!S()){p+=b;q=0;}}}int main(int I,    /**/char**l){//
   d=sizeof(f*);if(1<(O=_)){b=((sizeof(o)/sizeof(char*))-1)/4;q=22; p=     0;while(p<b*5){
  /*<*/if(Z-1){d=S()>96;i=S()-(d?96:32) ;q++;if(p<b*5&&!S()){p+=b;  q=      0;}Z=1;}/*[[*/
  while(i){_=o[0][S()-97];I=_-10?b:1;   for( ;I--;)putchar(_ );if   (!      --i||d)z(~i );}
 if(p==b*5&&O){p-=b;O--;}}return 0U;   }if(! (P=( f*)calloc /*]*/  (Q       ,I)))return 1;
 {;}for(_=p=1;p<I;p++){e(q=1);while    (q<   p&&  strcmp(  l[p     ]         ,l[(q)]))++  q;
 t=stdin;if(q<p){(void)memcpy/* "      */    (&P  [p],&P   [q     ]         ,Q);continue ;}
if(strcmp(l[p],"-")){t=fopen(l         [     p]   ,"rb"   )                  ;if(!t ){{;}  ;
printf(05+*o,l[p ]);return+1;                          {;}                   }}_=b= 1<<16   ;
*&O=5;do{if(!(P[p].q=realloc   (P[p].q,(P[p].P     +=       b)+1))){return   01;}O   &=72   /
6/*][*/;P[p].o+=d=fread(P[p]      .q       +P[     p           ].       o,  1,b,t)   ;}//
 while(d==b)      ;P [p].q[       P[       p]                  .o       ]=  012;d    =0;
 e(fclose(t        )  );P         [p]      .p                  =P[      p]  .q;if    (O)
 {for(;d<P[            p]         .o     ;d=                   q+     1)    {q=     d;
  while(q<P[                       p].o&&P[                   p].q[q]-     10     ){
  q++;}b=q-d;                        _=P                        [p].        d     ;
  if(b>_){/*]b                                                               */
   P[p].d=b;}{;                                                               }
   #undef/*pqdz'.*/  e//                                                     ;
   #define/*s8qdb]*/e/**/0                                                 //
   //<<.<<.----.>.<<.>++.++<                                              .[>]
   /*P[*/P[p].b++;continue;}}}t=                                        stdout;
  for (p=1;p<I;p++){/**/if(P[p].b>i                                    ){i=P[p].b;}}
 if  (O){for(p=0;p<i;p++){q=0;/*[*/while(I              >++q){_=P[q].p-P[q ].q;
b=   0;if(_<P[q ].o){while(012-*P[q].p)     {putchar(*(P[q].p++));b++;}P[q]. p++;
}   ;while (P[  q].d>b++)putchar(040);}            putchar(10);}return 0;}p   =1;
   for(;   p<I   ;p++)fwrite(P[p] .q,P[            p].o,1,t);return 0 ;}//
  #/*]     ]<.    [-]<[-]<[- ]<[    -]<           [-  ]<;*/elif  e    //b
 |(1        <<     ( __LINE__        /*           >>   `*//45))  |     01U
```

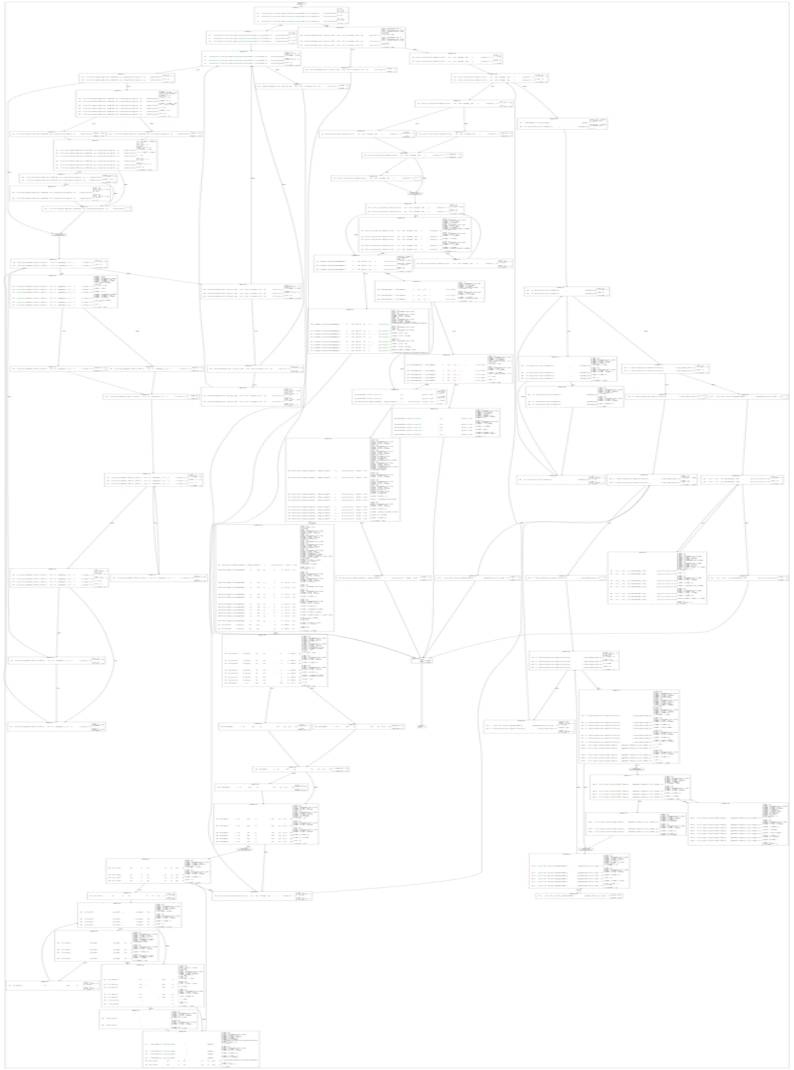
The International Obfuscated C Code Contest - http://www.ioccc.org/

VULNEX

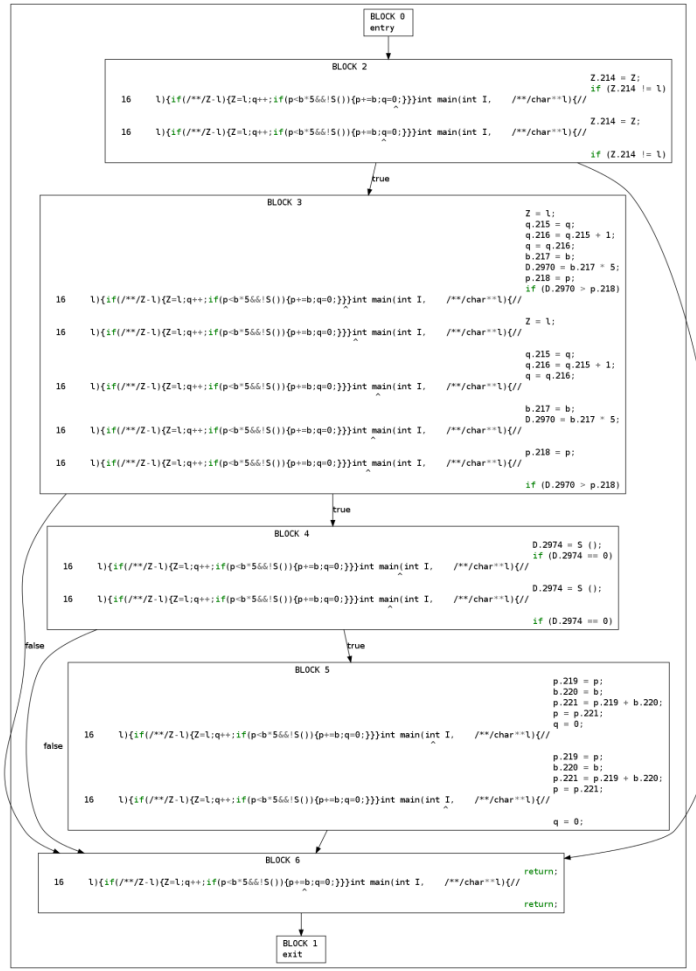
6. DEMO VI: OBFUSCATED C CODE ANALYSIS, MISAKA

```
root@kali:~/Desktop/hitb_demos/code/obfuscated_c/2013/misaka# gcc -fplugin=/vulnex/gcc_scanner/gcc-python-plugin/python.so -fplugin-arg-pyth
on-script=/vulnex/gcc_scanner/scripts/tintorera/analyzer.py -o misaka misaka.c
Warning: nobold #B00040 is not a known color.
Warning: nobold #B00040 is not a known color.
cc1: gcc-python-tree.c:549: PyObject* PyGccFunction_TypeObj_get_argument_types(PyGccTree*, void*): Assertion `size>0' failed.
*** WARNING *** there are active plugins, do not report this as a bug unless you can reproduce it without enabling any plugins.
Event                          | Plugins
PLUGIN_FINISH                  | python python
PLUGIN_PASS_EXECUTION          | python
misaka.c: In function 'S':
misaka.c:15:33: internal compiler error: Aborted
Please submit a full bug report,
with preprocessed source if appropriate.
See <file:///usr/share/doc/gcc-4.7/README.Bugs> for instructions.
root@kali:~/Desktop/hitb_demos/code/obfuscated_c/2013/misaka#
```

6. DEMO VI: OBFUSCATED C CODE ANALYSIS, MISAKA



MAIN



Z

7. Conclusions

7. DRAWBACKS

- gcc-python-plugin needs more work, fails many times
- So do Tintorera…
- Only C / C++ code

7. CONCLUSIONS

- Tintorera helps to analyze C code faster & better

- Practical code understanding for:
  - Saving time
  - Security reviews
  - Fuzzing: what and where to fuzz

7. NEXT STEPS

- Better & focused analysis (security, etc.)
- Vulnerabilities Detection
- More metrics
- Code Diff
- Cooler reports!
- Other languages ¿?

8. Q&A

- Thanks!
- @simonroses / @vulnexsl
- www.vulnex.com

VULNEX