Static and Dynamic Analysis

MST - 2/2

This will be you after today

Why reverse engineer?

- Develop competing products?
- Achieve interoperability
- Analyze malware
- Crack digital rights management
- Locate vulnerabilities

Static vs Dynamic Analysis

- **Static analysis** is the examination of program code without running it
  - Reading assembly, analyzing code patterns
  - Can be difficult for a human or computationally expensive for a computer
- **Dynamic analysis** is the examination of program state as it is running
  - Also known as debugging
  - Involves pausing a program at time t=x and examining the system state
  - Code paths that aren’t run won’t be easily examined

Static analysis

Static analysis tools

- **xxd**
  - Hex editor
- **strings**
  - View ASCII data in a file
- **objdump**
  - Basic disassembler
- **IDA**
  - Industry standard disassembler

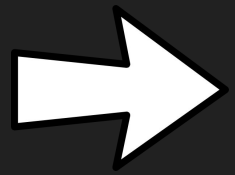
xxd

- Hex editors provide us with the raw bytes in a given file
- No interpretation of file data

```
$ cat hello.txt
Hello MST! Arbitrary data here
$ xxd hello.txt
0000000: 4865 6c6c 6f20 4d53  Hello MS
0000008: 5421 2041 7262 6974  T! Arbit
0000010: 7261 7279 2064 6174  rary dat
0000018: 6120 6865 7265 0a    a here.
```

xxd

```
.data
message:
.string "Hello World!"
.text
.globl main
main:
mov ecx, offset message
push ecx
call puts
pop ecx
mov eax, 0
ret
```

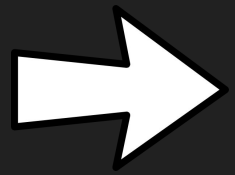


```
$ xxd hello
0000000: 7f45 4c46 0101 0100  .ELF....
0000008: 0000 0000 0000 0000  ........
0000010: 0200 0300 0100 0000  ........
0000018: 2083 0408 3400 0000   ...4...
. . . . .
000041d: b920 a004 0851 e8c8  . ...Q..
0000425: feff ff59 b800 0000  ...Y....
000042d: 00c3 9055 5731 ff56  ...UW1.V
. . . . .
```

Which part is the program code?

xxd

```
.data
message:
.string "Hello World!"
.text
.globl main
main:
mov ecx, offset message
push ecx
call puts
pop ecx
mov eax, 0
ret
```



```
$ xxd hello
0000000: 7f45 4c46 0101 0100  .ELF....
0000008: 0000 0000 0000 0000  ........
0000010: 0200 0300 0100 0000  ........
0000018: 2083 0408 3400 0000   ...4...
. . . . .
000041d: b920 a004 0851 e8c8  . ...Q..
0000425: feff ff59 b800 0000  ...Y....
000042d: 00c3 9055 5731 ff56  ...UW1.V
. . . . .
```

Which part is the program code?

strings

- Prints out any ASCII strings from the file that are 4+ characters in length
- Useful for getting an idea of what data your file contains

```
$ strings hello
/lib/ld-linux.so.2
libc.so.6
_IO_stdin_used
puts
__libc_start_main
__gmon_start__
GLIBC_2.0
PTRh
[^_]
;*2$"8
Hello World!
GCC: (Ubuntu 4.8.4-2ubuntu1~14.04.3)
4.8.4
. . . . .
```

objdump

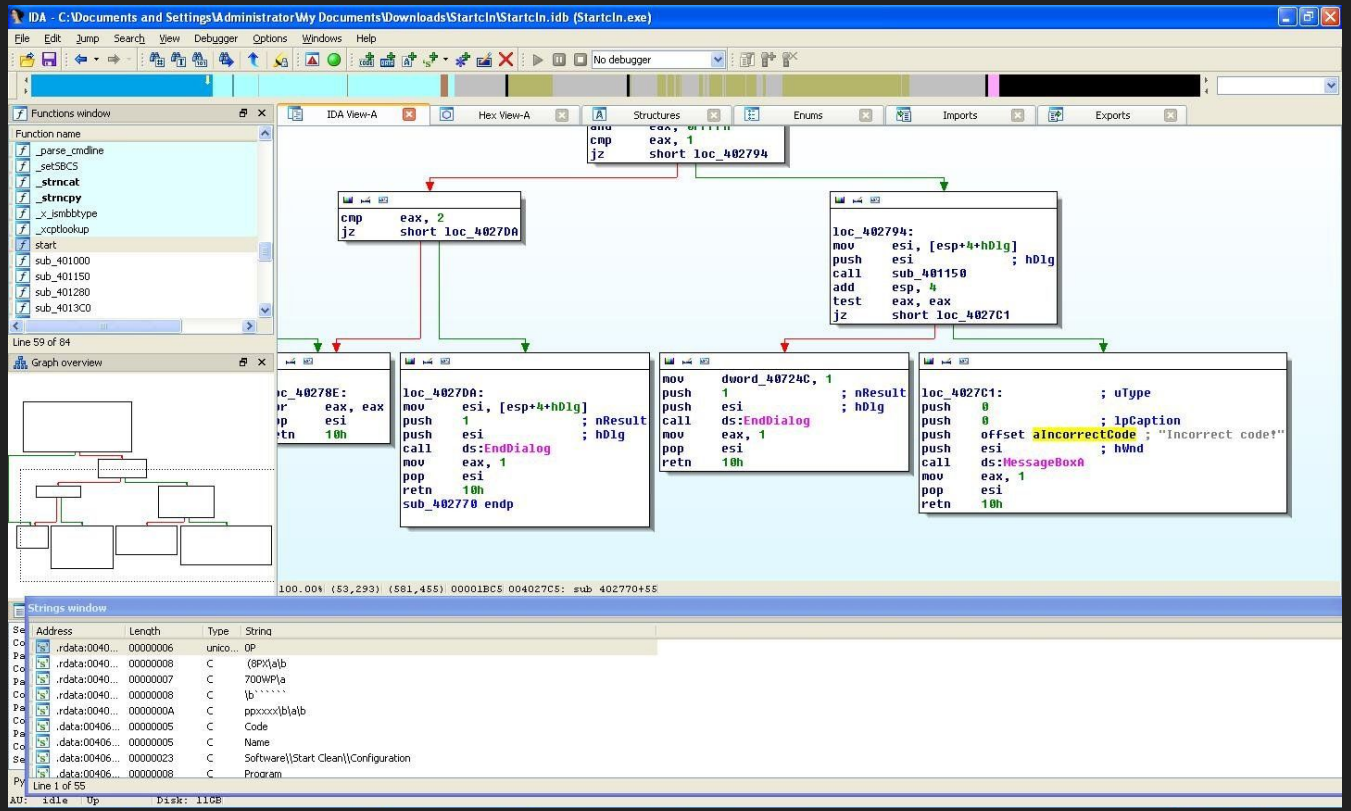
- Displays tons of useful information about an executable file
- Can display code for each function in the file

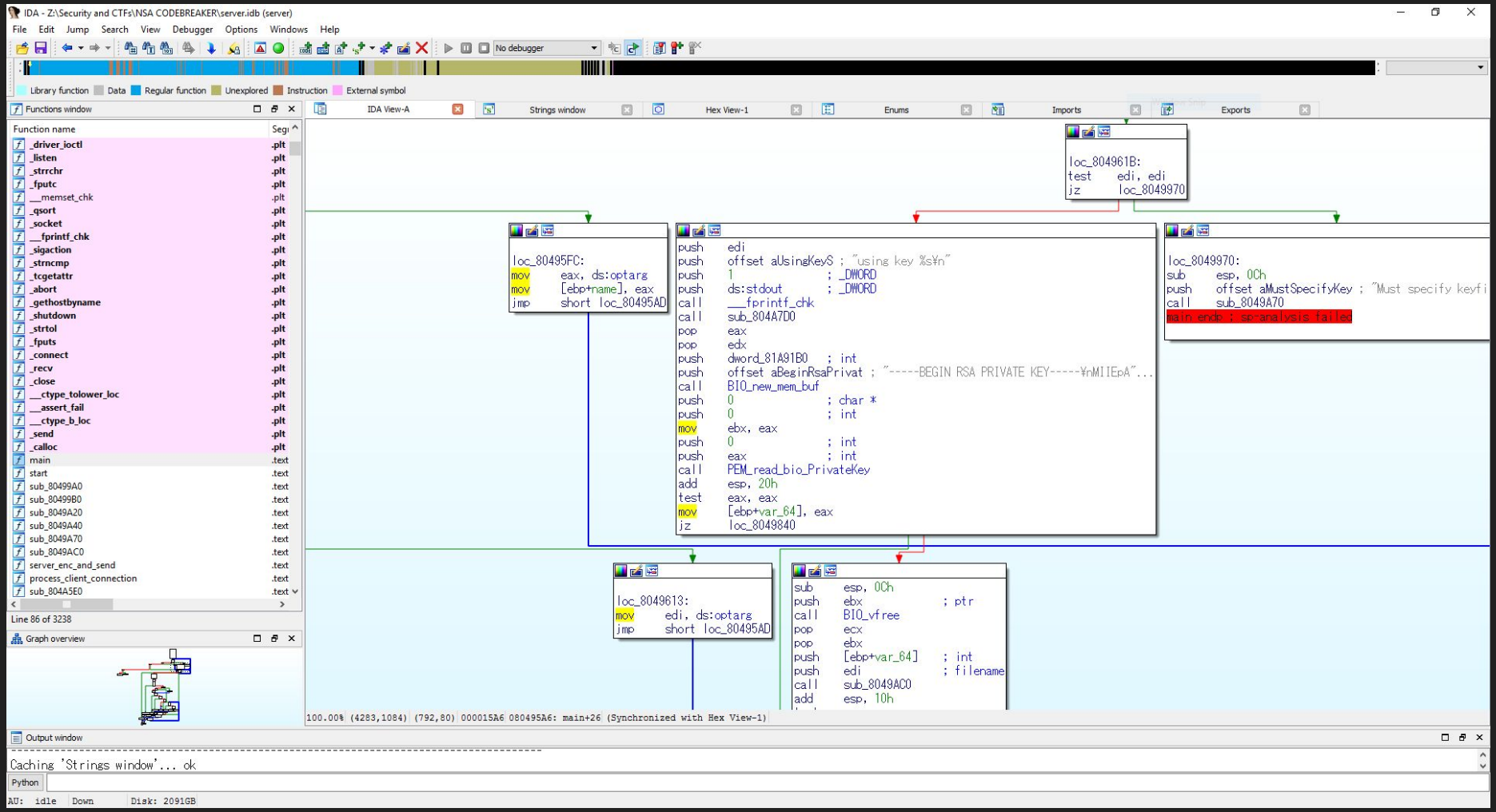
```
$ objdump -d hello
0804841d <main>:
 804841d:       b9 20 a0 04 08          mov    $0x804a020,%ecx
 8048422:       51                      push   %ecx
 8048423:       e8 c8 fe ff ff          call   80482f0 <puts@plt>
 8048428:       59                      pop    %ecx
 8048429:       b8 00 00 00 00          mov    $0x0,%eax
 804842e:       c3                      ret
```

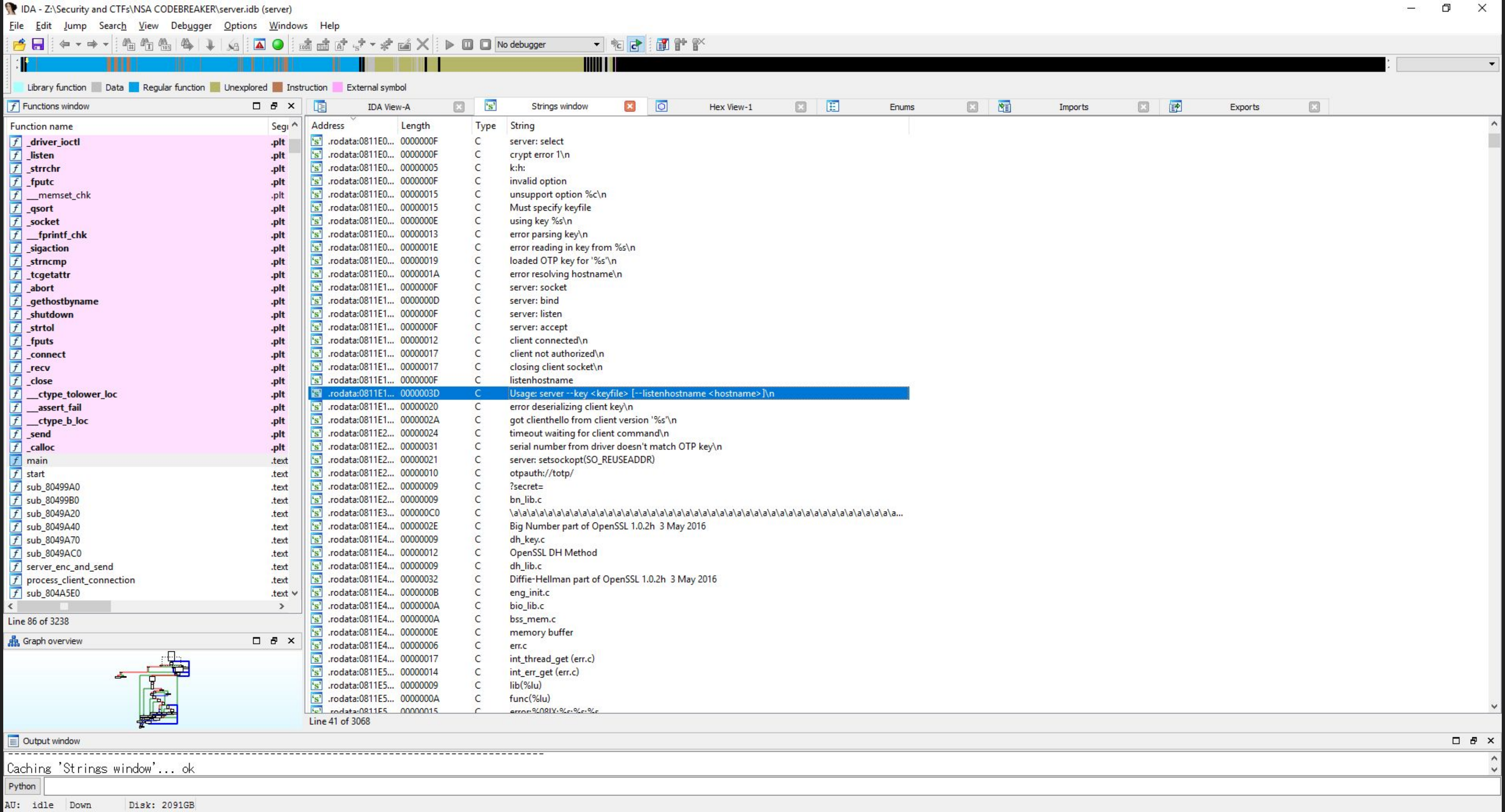
IDA (Interactive Disassembler)

- Gold standard disassembler that everyone uses
- Graphical display, runs on Windows/OSX/Linux (but mainly Windows)
- Provides all the features in tools we've seen, and more
- Extensive code analysis features
- Very expensive

IDA (Interactive Disassembler)




IDA - Z:\Security and CTFs\NSA CODEBREAKER\server.idb (server)
File Edit Jump Search View Debugger Options Windows Help
No debugger
Library function Data Regular function Unexplored Instruction External symbol
Functions window
Function name
_driver_ioctl .plt
_listen .plt
_strrchr .plt
_fputc .plt
__memset_chk .plt
_qsort .plt
_socket .plt
__fprintf_chk .plt
_sigaction .plt
_strncmp .plt
_tcgetattr .plt
_abort .plt
_gethostbyname .plt
_shutdown .plt
_strtol .plt
_fputs .plt
_connect .plt
_recv .plt
_close .plt
__ctype_tolower_loc .plt
__assert_fail .plt
__ctype_b_loc .plt
_send .plt
_calloc .plt
main .text
start .text
sub_80499A0 .text
sub_80499B0 .text
sub_8049A20 .text
sub_8049A40 .text
sub_8049A70 .text
sub_8049AC0 .text
server_enc_and_send .text
process_client_connection .text
sub_804A5E0 .text
Line 86 of 3238
Graph overview
IDA View-A
Strings window
Hex View-1
Enums
Imports
Exports
loc_804961B:
test edi, edi
jz loc_8049970
loc_80495FC:
mov eax, ds:optarg
mov [ebp+name], eax
jmp short loc_80495AD
push edi
push offset aUsingKeyS ; "using key %s¥n"
push 1 ; _DWORD
push ds:stdout ; _DWORD
call ___fprintf_chk
call sub_804A7D0
pop eax
pop edx
push dword_81A91B0 ; int
push offset aBeginRsaPrivat ; "-----BEGIN RSA PRIVATE KEY-----¥nMIIEpA"...
call BIO_new_mem_buf
push 0 ; char *
push 0 ; int
mov ebx, eax
push 0 ; int
push eax ; int
call PEM_read_bio_PrivateKey
add esp, 20h
test eax, eax
mov [ebp+var_64], eax
jz loc_8049840
loc_8049970:
sub esp, 0Ch
push offset aMustSpecifyKey ; "Must specify keyfi
call sub_8049A70
main endp ; sp-analysis failed
loc_8049613:
mov edi, ds:optarg
jmp short loc_80495AD
sub esp, 0Ch
push ebx ; ptr
call BIO_vfree
pop ecx
pop ebx
push [ebp+var_64] ; int
push edi ; filename
call sub_8049AC0
add esp, 10h
100.00% (4283,1084) (792,80) 000015A6 080495A6: main+26 (Synchronized with Hex View-1)
Output window
Caching 'Strings window'... ok
Python
AU: idle Down Disk: 2091GB


IDA - Z:\Security and CTFs\NSA CODEBREAKER\server.idb (server)
File Edit Jump Search View Debugger Options Windows Help
No debugger
Library function Data Regular function Unexplored Instruction External symbol
Functions window
Function name Segi
_driver_ioctl .plt
_listen .plt
_strrchr .plt
_fputc .plt
__memset_chk .plt
_qsort .plt
_socket .plt
__fprintf_chk .plt
_sigaction .plt
_strncmp .plt
_tcgetattr .plt
_abort .plt
_gethostbyname .plt
_shutdown .plt
_strtol .plt
_fputs .plt
_connect .plt
_recv .plt
_close .plt
__ctype_tolower_loc .plt
__assert_fail .plt
__ctype_b_loc .plt
_send .plt
_calloc .plt
main .text
start .text
sub_80499A0 .text
sub_80499B0 .text
sub_8049A20 .text
sub_8049A40 .text
sub_8049A70 .text
sub_8049AC0 .text
server_enc_and_send .text
process_client_connection .text
sub_804A5E0 .text
Line 86 of 3238
Graph overview
IDA View-A
Strings window
Hex View-1
Enums
Imports
Exports
Address Length Type String
.rodata:0811E0... 0000000F C server: select
.rodata:0811E0... 0000000F C crypt error 1\n
.rodata:0811E0... 00000005 C k:h:
.rodata:0811E0... 0000000F C invalid option
.rodata:0811E0... 00000015 C unsupport option %c\n
.rodata:0811E0... 00000015 C Must specify keyfile
.rodata:0811E0... 0000000E C using key %s\n
.rodata:0811E0... 00000013 C error parsing key\n
.rodata:0811E0... 0000001E C error reading in key from %s\n
.rodata:0811E0... 00000019 C loaded OTP key for '%s'\n
.rodata:0811E0... 0000001A C error resolving hostname\n
.rodata:0811E1... 0000000F C server: socket
.rodata:0811E1... 0000000D C server: bind
.rodata:0811E1... 0000000F C server: listen
.rodata:0811E1... 0000000F C server: accept
.rodata:0811E1... 00000012 C client connected\n
.rodata:0811E1... 00000017 C client not authorized\n
.rodata:0811E1... 00000017 C closing client socket\n
.rodata:0811E1... 0000000F C listenhostname
.rodata:0811E1... 0000003D C Usage: server --key <keyfile> [--listenhostname <hostname>]\n
.rodata:0811E1... 00000020 C error deserializing client key\n
.rodata:0811E1... 0000002A C got clienthello from client version '%s'\n
.rodata:0811E2... 00000024 C timeout waiting for client command\n
.rodata:0811E2... 00000031 C serial number from driver doesn't match OTP key\n
.rodata:0811E2... 00000021 C server: setsockopt(SO_REUSEADDR)
.rodata:0811E2... 00000010 C otpauth://totp/
.rodata:0811E2... 00000009 C ?secret=
.rodata:0811E2... 00000009 C bn_lib.c
.rodata:0811E3... 000000C0 C \a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a\a...
.rodata:0811E4... 0000002E C Big Number part of OpenSSL 1.0.2h 3 May 2016
.rodata:0811E4... 00000009 C dh_key.c
.rodata:0811E4... 00000012 C OpenSSL DH Method
.rodata:0811E4... 00000009 C dh_lib.c
.rodata:0811E4... 00000032 C Diffie-Hellman part of OpenSSL 1.0.2h 3 May 2016
.rodata:0811E4... 0000000B C eng_init.c
.rodata:0811E4... 0000000A C bio_lib.c
.rodata:0811E4... 0000000A C bss_mem.c
.rodata:0811E4... 0000000E C memory buffer
.rodata:0811E4... 00000006 C err.c
.rodata:0811E4... 00000017 C int_thread_get (err.c)
.rodata:0811E5... 00000014 C int_err_get (err.c)
.rodata:0811E5... 00000009 C lib(%lu)
.rodata:0811E5... 0000000A C func(%lu)
Line 41 of 3068
Output window
Caching 'Strings window'... ok
Python
AU: idle Down Disk: 2091GB

IDA - Z:\Security and CTFs\NSA CODEBREAKER\server.idb (server)

File Edit Jump Search View Debugger Options Windows Help

No debugger

Library function | Data | Regular function | Unexplored | Instruction | External symbol

Functions window

Function name	Segment
_driver_ioctl	.plt
_listen	.plt
_strrchr	.plt
_fputc	.plt
__memset_chk	.plt
_qsort	.plt
_socket	.plt
__fprintf_chk	.plt
_sigaction	.plt
_strncmp	.plt
_tcgetattr	.plt
_abort	.plt
_gethostbyname	.plt
_shutdown	.plt
_strtol	.plt
_fputs	.plt
_connect	.plt
_recv	.plt
_close	.plt
__ctype_tolower_loc	.plt
__assert_fail	.plt
__ctype_b_loc	.plt
_send	.plt
_calloc	.plt
main	.text
start	.text
sub_80499A0	.text
sub_80499B0	.text
sub_8049A20	.text
sub_8049A40	.text
sub_8049A70	.text
sub_8049AC0	.text
server_enc_and_send	.text
process_client_connection	.text
sub_804A5E0	.text

Line 86 of 3238

Graph overview

IDA View-A | Pseudocode-A | Strings window | Hex View-1 | Enums | Imports | Exports

```
37  v5 = a2;
38  v30 = *MK_FP(__GS__, 20);
39  while ( 1 )
40  {
41    v6 = getopt_long(v4, v5, "k:h:", &longopts, 0);
42    if ( v6 == -1 )
43      break;
44    if ( v6 != 104 )
45    {
46      if ( v6 != 107 )
47      {
48        if ( v6 != 63 )
49        {
50          __fprintf_chk(stdout, 1, "unsupport option %c¥n");
51          exit(1);
52        }
53        sub_8049A70("invalid option");
54      }
55      v3 = (char *)optarg;
56    }
57  }
58  if ( !v3 )
59LABEL_45:
60    sub_8049A70("Must specify keyfile");
61  __fprintf_chk(stdout, 1, "using key %s¥n");
62  sub_804A7D0();
63  v7 = (void *)BIO_new_mem_buf(aBeginRsaPrivat, dword_81A91B0);
64  v19 = (void *)PEM_read_bio_PrivateKey((int)v7, 0, 0, 0);
65  if ( !v19 )
66  {
67    __fprintf_chk(stderr, 1, "error parsing key¥n");
68    goto LABEL_32;
69  }
70  BIO_vfree(v7);
71  ptr = sub_8049AC0(v3, (int)v19);
72  if ( !ptr )
73  {
74    __fprintf_chk(stderr, 1, "error reading in key from %s¥n");
75    goto LABEL_36;
76  }
77  v8 = 0;
78  __printf_chk(1, "loaded OTP key for '%s'¥n", ptr[2], v8);
```

000018DD main:74

Output window

```
804A7E0: using guessed type int __cdecl sub_804A7E0(_DWORD, _DWORD);
81AA9C0: using guessed type int optarg;
```

Python

AU: idle | Down | Disk: 2091GB

What’s important about this

- objdump & IDA will be very useful for this week’s homework
- Learning IDA is essential if you’re interested in the field
  - IDA Pro Book
  - IDA Free version

Dynamic analysis

Dynamic Analysis (debugging)

- Analyze a program during runtime to see system and memory state
- Step through code instructions to follow what happens

Itrace

strace

gdb

Homework

Homework

- Crack our program!
- You must find the correct password that the program accepts
- Submit both the correct password and an explanation of how you found it
- You can use any tool we've talked about, or similar tools, to help solve it

Tips

- gdb is useful for stepping through the program and following execution
- IDA can show you an overarching program structure that is easier to follow than straight assembly
- The binary is x64, so parameters are passed in RDI, RSI, RDX, RCX, R8, R9

```
$ ./crackme
What is the secret password?
testing123
Nope, that is not correct. Sorry!
$ ./crackme
What is the password?
(HIDDEN)
Well done! You found the secret password!
```

Homework grading

- Submit your assembly code file to cm7bv@virginia.edu with the subject “MST Assignment 2 - <YOUR_UVA_ID>”
  - eg: “MST Assignment 2 - cm7bv”
- Also, include a brief (1-paragraph) description of what you did and how it went

Useful Resources

- IDA Free version
- IDA Pro book
- CS 2150 gdb tutorial
- IDA Pro tutorial