baby-re[EN]

Unknown macro: 'html'

List

Information

Description

Get to reversing.

baby-re

File

- baby-re

Source Code

- https://github.com/legitbs/quals-2016/tree/master/baby-re

Writeup

File information

```
lazenca0x0@ubuntu:~/CTF/DEFCON2016/baby's/baby-re$ file baby-re
baby-re: ELF 64-bit LSB  executable, x86-64, version 1 (SYSV), dynamically linked (uses shared libs), for GNU
/Linux 2.6.32, BuildID[sha1]=5d5783d23d78bf70b80d658bccbce365f7448693, not stripped
lazenca0x0@ubuntu:~/CTF/DEFCON2016/baby's/baby-re$ checksec --file baby-re
RELRO           STACK CANARY      NX            PIE             RPATH      RUNPATH        FORTIFY
Fortified Fortifiable  FILE
Partial RELRO   Canary found      NX enabled    No PIE          No RPATH   No RUNPATH   Yes
0               2        baby-re
```

Binary analysis

Main

- The question is input a total of 13 values by user.
- The input values are passed to CheckSolution() function.
    - The function outputs Flag if the values input by user are satisfied with condition.

main()

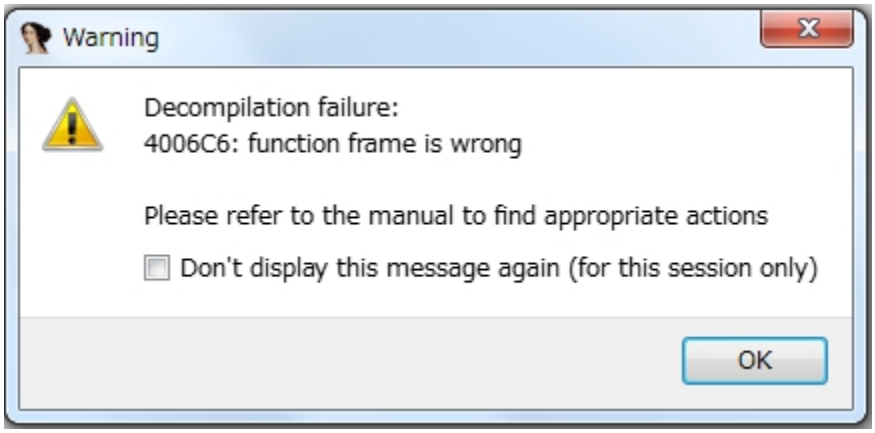
```
int __cdecl main(int argc, const char **argv, const char **envp)
{
  int result; // eax@4
  __int64 v4; // rbx@4
  unsigned int v5[13]; // [rsp+0h] [rbp-60h]@1
  __int64 v6; // [rsp+38h] [rbp-28h]@1

  v6 = *MK_FP(__FS__, 40LL);
  printf("Var[0]: ", argv, envp);
  fflush(_bss_start);
  __isoc99_scanf("%d", v5);
  printf("Var[1]: ", v5);
  fflush(_bss_start);
  __isoc99_scanf("%d", &v5[1]);
  printf("Var[2]: ", &v5[1]);
  fflush(_bss_start);
  __isoc99_scanf("%d", &v5[2]);
  printf("Var[3]: ", &v5[2]);
  fflush(_bss_start);
  __isoc99_scanf("%d", &v5[3]);
  printf("Var[4]: ", &v5[3]);
  fflush(_bss_start);
  __isoc99_scanf("%d", &v5[4]);
  printf("Var[5]: ", &v5[4]);
  fflush(_bss_start);
  __isoc99_scanf("%d", &v5[5]);
  printf("Var[6]: ", &v5[5]);
  fflush(_bss_start);
  __isoc99_scanf("%d", &v5[6]);
  printf("Var[7]: ", &v5[6]);
  fflush(_bss_start);
  __isoc99_scanf("%d", &v5[7]);
  printf("Var[8]: ", &v5[7]);
  fflush(_bss_start);
  __isoc99_scanf("%d", &v5[8]);
  printf("Var[9]: ", &v5[8]);
  fflush(_bss_start);
  __isoc99_scanf("%d", &v5[9]);
  printf("Var[10]: ", &v5[9]);
  fflush(_bss_start);
  __isoc99_scanf("%d", &v5[10]);
  printf("Var[11]: ", &v5[10]);
  fflush(_bss_start);
  __isoc99_scanf("%d", &v5[11]);
  printf("Var[12]: ", &v5[11]);
  fflush(_bss_start);
  __isoc99_scanf("%d", &v5[12]);
  if ( (unsigned __int8)CheckSolution(v5) )
    printf(
      "The flag is: %c%c%c%c%c%c%c%c%c%c%c%c%c\n", v5[0], v5[1], v5[2], v5[3], v5[4], v5[5], v5[6],        v5
[7], v5[8], v5[9], v5[10], v5[11], v5[12]);
  else
    puts("Wrong");
  result = 0;
  v4 = *MK_FP(__FS__, 40LL) ^ v6;
  return result;
}
```

CheckSolution

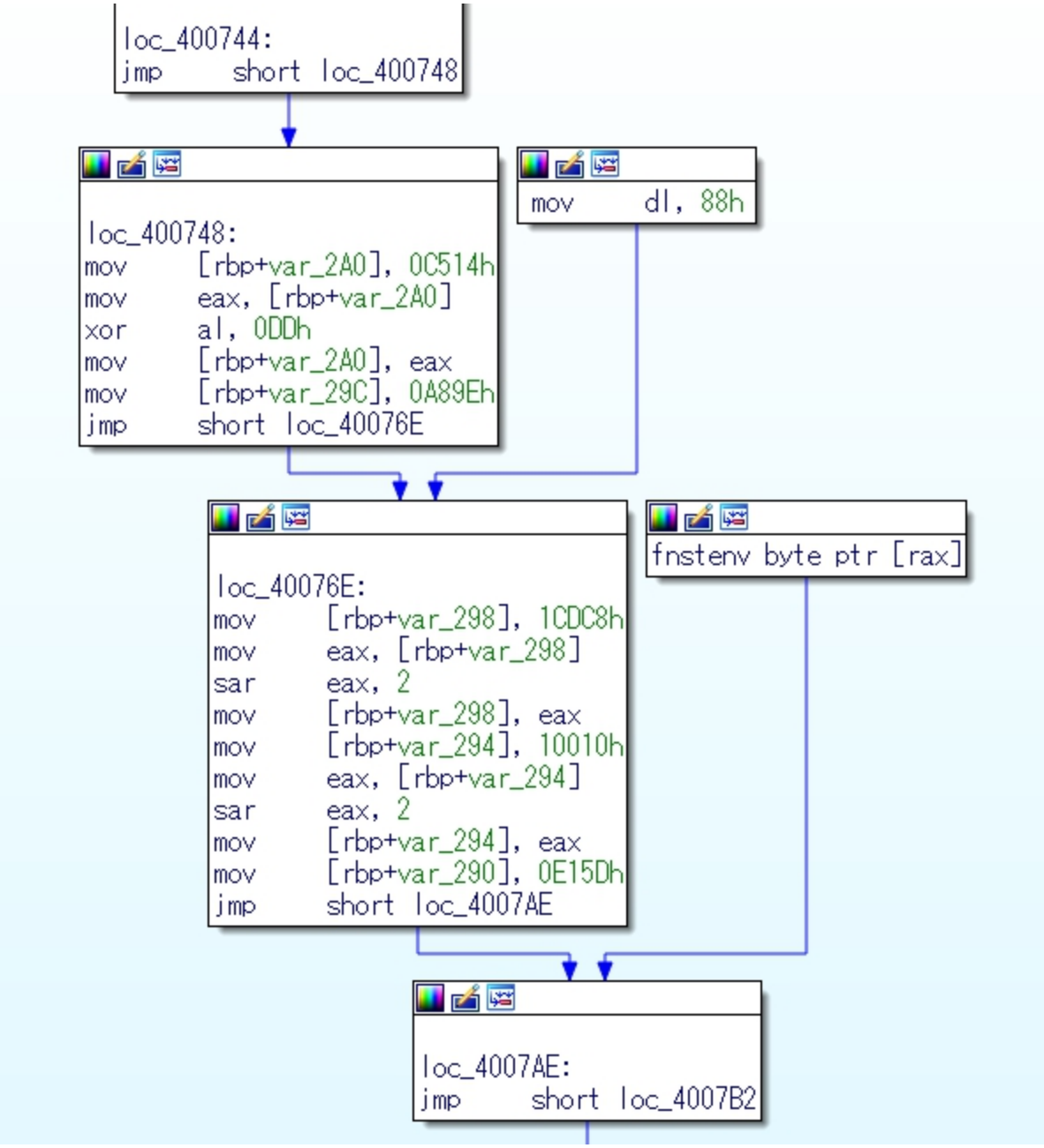
- The error occurs as below if open the function by Hex-rays.

Warning



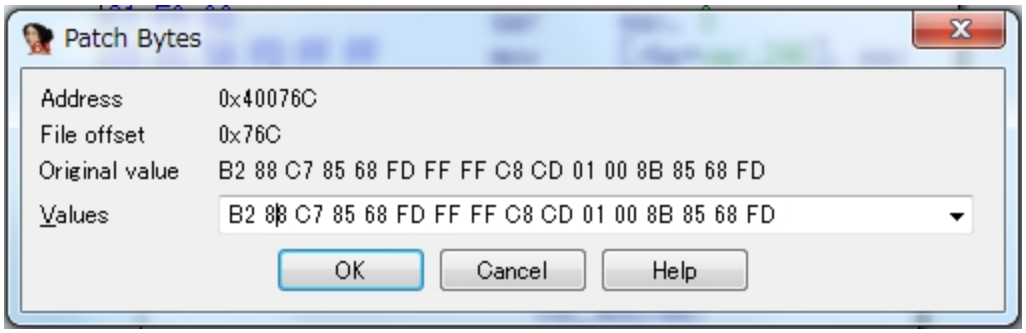
- The error occurs because Junk code is inserted in the code.

JUNK Code

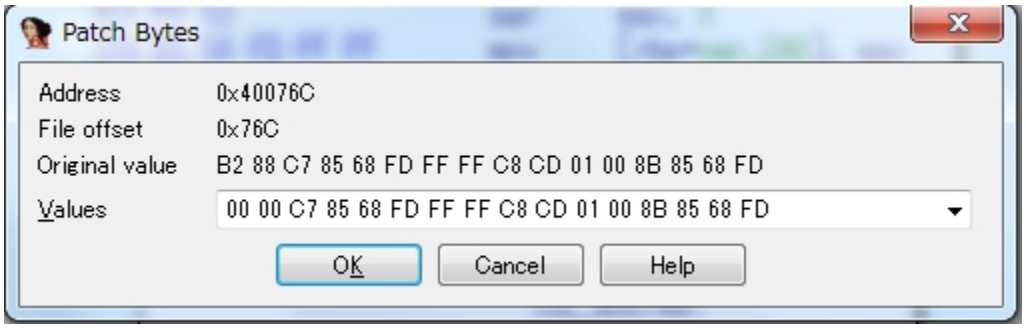


- After selecting the code to be removed, change byte value of JUNK Code to "00" using "Edit" "Patch program" "Change byte...".

JUNK Code



JUNK Code



- You can see the code below using Hex-Rays after patching.
- The function performs as below.
- It manages given operation using passed argument value.
- It checks if operation value and condition value are same using if().
    - It repeats the process 13times.
- Which means, Flag string can be known if 13 conditional statements are satisfied.

CheckSolution(Remove JUNK Code)

```
__int64 __fastcall CheckSolution(_DWORD *a1)
{
  __int64 result; // rax@2
  __int64 v2; // rsi@26

  if ( 39342 * a1[11]
     + 21090 * a1[10]
     + 14626 * a1[9]
     + 57693 * a1[8]
     + 16388 * a1[7]
     + 29554 * a1[6]
     + 43166 * a1[5]
     + 50633 * a1[4]
     + 37485 * *a1
     - 21621 * a1[1]
     - 1874 * a1[2]
     - 46273 * a1[3]
     + 54757 * a1[12] == 21399379 )
  {
    if ( 22599 * a1[5]
       + 14794 * a1[4]
       + 38962 * a1[3]
       + 50936 * *a1
       + 4809 * a1[1]
       - 6019 * a1[2]
       - 837 * a1[6]
       - 36727 * a1[7]
```

```
   - 50592 * a1[8]
   - 11829 * a1[9]
   - 20046 * a1[10]
   - 9256 * a1[11]
   + 53228 * a1[12] == 1453872 )
{
  if ( 5371 * a1[11]
     + 42654 * a1[10]
     + 17702 * a1[8]
     + 26907 * a1[3]
     + -38730 * *a1
     + 52943 * a1[1]
     - 16882 * a1[2]
     - 44446 * a1[4]
     - 18601 * a1[5]
     - 65221 * a1[6]
     - 47543 * a1[7]
     - 33910 * a1[9]
     + 11469 * a1[12] == -5074020 )
  {
    if ( 8621 * a1[10]
       + 34805 * a1[7]
       + 10649 * a1[6]
       + 54317 * a1[4]
       + 57747 * *a1
       - 23889 * a1[1]
       - 26016 * a1[2]
       - 25170 * a1[3]
       - 32337 * a1[5]
       - 9171 * a1[8]
       - 22855 * a1[9]
       - 634 * a1[11]
       - 11864 * a1[12] == -5467933 )
    {
      if ( 15578 * a1[11]
         + 43186 * a1[9]
         + 28134 * a1[8]
         + 54889 * a1[4]
         + 34670 * a1[3]
         + 43964 * a1[2]
         + -14005 * *a1
         + 16323 * a1[1]
         - 6141 * a1[5]
         - 35427 * a1[6]
         - 61977 * a1[7]
         - 59676 * a1[10]
         + 50082 * a1[12] == 7787144 )
      {
        if ( 10305 * a1[11]
           + 29341 * a1[10]
           + 13602 * a1[7]
           + 39603 * a1[6]
           + 13608 * a1[2]
           + -40760 * *a1
           - 22014 * a1[1]
           - 4946 * a1[3]
           - 26750 * a1[4]
           - 31708 * a1[5]
           - 59055 * a1[8]
           - 32738 * a1[9]
           - 15650 * a1[12] == -8863847 )
        {
          if ( 16047 * a1[9]
             + 55241 * a1[7]
             + 13477 * a1[2]
             + -47499 * *a1
             + 57856 * a1[1]
             - 10219 * a1[3]
             - 5032 * a1[4]
             - 21039 * a1[5]
             - 29607 * a1[6]
```

```
   - 6065 * a1[8]
   - 4554 * a1[10]
   - 2262 * a1[11]
   + 18903 * a1[12] == -747805 )
{
  if ( 41178 * a1[11]
     + 47909 * a1[7]
     + 53309 * a1[6]
     + -65419 * *a1
     + 17175 * a1[1]
     - 9410 * a1[2]
     - 22514 * a1[3]
     - 52377 * a1[4]
     - 9235 * a1[5]
     - 59111 * a1[8]
     - 41289 * a1[9]
     - 24422 * a1[10]
     - 23447 * a1[12] == -11379056 )
  {
    if ( 15699 * a1[10]
       + 58551 * a1[5]
       + 46767 * a1[4]
       + 33381 * a1[3]
       + 1805 * *a1
       + 4135 * a1[1]
       - 16900 * a1[2]
       - 34118 * a1[6]
       - 44920 * a1[7]
       - 11933 * a1[8]
       - 20530 * a1[9]
       - 36597 * a1[11]
       + 18231 * a1[12] == -166140 )
    {
      if ( 10788 * a1[10]
         + 18975 * a1[9]
         + 15033 * a1[8]
         + 42363 * a1[7]
         + 47052 * a1[6]
         + 41284 * a1[3]
         + -42941 * *a1
         + 61056 * a1[1]
         - 45169 * a1[2]
         - 1722 * a1[4]
         - 26423 * a1[5]
         - 33319 * a1[11]
         + 63680 * a1[12] == 9010363 )
      {
        if ( 30753 * a1[10]
           + 22613 * a1[9]
           + 58786 * a1[7]
           + 12587 * a1[6]
           + 12746 * a1[5]
           + -37085 * *a1
           - 51590 * a1[1]
           - 17798 * a1[2]
           - 10127 * a1[3]
           - 52388 * a1[4]
           - 8269 * a1[8]
           - 20853 * a1[11]
           + 32216 * a1[12] == -4169825 )
        {
          if ( 57612 * a1[11]
             + 47348 * a1[9]
             + 48719 * a1[8]
             + 9228 * a1[5]
             + 65196 * a1[4]
             + 36650 * *a1
             + 47566 * a1[1]
             - 33282 * a1[2]
             - 59180 * a1[3]
             - 59599 * a1[6]
```

```
                        - 62888 * a1[7]
                        - 37592 * a1[10]
                        + 40510 * a1[12] == 4081505 )
                       result = 25633 * a1[11]
                              + 25252 * a1[9]
                              + 28153 * a1[8]
                              + 26517 * a1[7]
                              + 59511 * a1[4]
                              + 4102 * a1[3]
                              + 51735 * *a1
                              + 35879 * a1[1]
                              - 63890 * a1[2]
                              - 21386 * a1[5]
                              - 20769 * a1[6]
                              - 43789 * a1[10]
                              + 7314 * a1[12] == 1788229;
                     else
                       result = 0LL;
                   }
                   else
                   {
                     result = 0LL;
                   }
                 }
                 else
                 {
                   result = 0LL;
                 }
               }
               else
               {
                 result = 0LL;
               }
             }
             else
             {
               result = 0LL;
             }
           }
           else
           {
             result = 0LL;
           }
         }
         else
         {
           result = 0LL;
         }
       }
       else
       {
         result = 0LL;
       }
     }
     else
     {
       result = 0LL;
     }
   }
   else
   {
     result = 0LL;
   }
 }
 else
 {
```

```
    result = 0LL;
  }
  v2 = *MK_FP(__FS__, 40LL) ^ *MK_FP(__FS__, 40LL);
  return result;
}
```

structure of Exploit code

1. Input 13 values satisfying "==" operation.
    a. Find 13 values using the simultaneous linear equations.

Information for attack

Find 13 values satisfying "==" operation("z3" Module(Python))

- You can use "z3" Module (Python) developed by Microsoft to solve the operation. You need to install "z3" module first.

z3 install

```
$ git clone https://github.com/Z3Prover/z3.git
$ cd z3
$ python scripts/mk_make.py
$ cd build
$ make
$ sudo make install
```

Github

https://github.com/Z3Prover/z3

- Let's solve the simultaneous linear equations using "z3" as below.

z3 Example

```
from z3 import *

x = Int('x')
y = Int('y')
solve(x > 2, y < 10, x + 2*y == 7)
```

- You can get the following result after running the code.

python test2.py

```
lazenca0x0@ubuntu:~/CTF/DEFCON2016/baby's/baby-re$ python test2.py
[y = 0, x = 7]
```

Z3 API in Python

http://www.cs.tau.ac.il/~msagiv/courses/asv/z3py/guide-examples.htm

- You can solve the simultaneous linear equations used in this program as below.

test.py

```
from z3 import *
```

```
a1 = [Int("a1[%d]"%(i)) for i in range(0,13)]
s = Solver()

s.add(39342 * a1[11]
     + 21090 * a1[10]
     + 14626 * a1[9]
     + 57693 * a1[8]
     + 16388 * a1[7]
     + 29554 * a1[6]
     + 43166 * a1[5]
     + 50633 * a1[4]
     + 37485 * a1[0]
     - 21621 * a1[1]
     - 1874 * a1[2]
     - 46273 * a1[3]
     + 54757 * a1[12] == 21399379)

s.add(22599 * a1[5]
     + 14794 * a1[4]
     + 38962 * a1[3]
     + 50936 * a1[0]
     + 4809 * a1[1]
     - 6019 * a1[2]
     - 837 * a1[6]
     - 36727 * a1[7]
     - 50592 * a1[8]
     - 11829 * a1[9]
     - 20046 * a1[10]
     - 9256 * a1[11]
     + 53228 * a1[12] == 1453872)

s.add(5371 * a1[11]
     + 42654 * a1[10]
     + 17702 * a1[8]
     + 26907 * a1[3]
     + -38730 * a1[0]
     + 52943 * a1[1]
     - 16882 * a1[2]
     - 44446 * a1[4]
     - 18601 * a1[5]
     - 65221 * a1[6]
     - 47543 * a1[7]
     - 33910 * a1[9]
     + 11469 * a1[12] == -5074020)

s.add(8621 * a1[10]
     + 34805 * a1[7]
     + 10649 * a1[6]
     + 54317 * a1[4]
     + 57747 * a1[0]
     - 23889 * a1[1]
     - 26016 * a1[2]
     - 25170 * a1[3]
     - 32337 * a1[5]
     - 9171 * a1[8]
     - 22855 * a1[9]
     - 634 * a1[11]
     - 11864 * a1[12] == -5467933)

s.add(15578 * a1[11]
       + 43186 * a1[9]
       + 28134 * a1[8]
       + 54889 * a1[4]
       + 34670 * a1[3]
       + 43964 * a1[2]
       + -14005 * a1[0]
       + 16323 * a1[1]
       - 6141 * a1[5]
       - 35427 * a1[6]
       - 61977 * a1[7]
       - 59676 * a1[10]
```

```
       + 50082 * a1[12] == 7787144)

s.add(10305 * a1[11]
       + 29341 * a1[10]
       + 13602 * a1[7]
       + 39603 * a1[6]
       + 13608 * a1[2]
       + -40760 * a1[0]
       - 22014 * a1[1]
       - 4946 * a1[3]
       - 26750 * a1[4]
       - 31708 * a1[5]
       - 59055 * a1[8]
       - 32738 * a1[9]
       - 15650 * a1[12] == -8863847)

s.add(16047 * a1[9]
       + 55241 * a1[7]
       + 13477 * a1[2]
       + -47499 * a1[0]
       + 57856 * a1[1]
       - 10219 * a1[3]
       - 5032 * a1[4]
       - 21039 * a1[5]
       - 29607 * a1[6]
       - 6065 * a1[8]
       - 4554 * a1[10]
       - 2262 * a1[11]
       + 18903 * a1[12] == -747805 )

s.add(41178 * a1[11]
       + 47909 * a1[7]
       + 53309 * a1[6]
       + -65419 * a1[0]
       + 17175 * a1[1]
       - 9410 * a1[2]
       - 22514 * a1[3]
       - 52377 * a1[4]
       - 9235 * a1[5]
       - 59111 * a1[8]
       - 41289 * a1[9]
       - 24422 * a1[10]
       - 23447 * a1[12] == -11379056)

s.add(15699 * a1[10]
          + 58551 * a1[5]
      + 46767 * a1[4]
      + 33381 * a1[3]
      + 1805 * a1[0]
      + 4135 * a1[1]
      - 16900 * a1[2]
      - 34118 * a1[6]
      - 44920 * a1[7]
      - 11933 * a1[8]
      - 20530 * a1[9]
      - 36597 * a1[11]
      + 18231 * a1[12] == -166140)

s.add(10788 * a1[10]
       + 18975 * a1[9]
       + 15033 * a1[8]
       + 42363 * a1[7]
       + 47052 * a1[6]
       + 41284 * a1[3]
       + -42941 * a1[0]
       + 61056 * a1[1]
       - 45169 * a1[2]
       - 1722 * a1[4]
       - 26423 * a1[5]
       - 33319 * a1[11]
       + 63680 * a1[12] == 9010363)
```

```
s.add(30753 * a1[10]
        + 22613 * a1[9]
    + 58786 * a1[7]
    + 12587 * a1[6]
    + 12746 * a1[5]
    + -37085 * a1[0]
    - 51590 * a1[1]
    - 17798 * a1[2]
    - 10127 * a1[3]
    - 52388 * a1[4]
    - 8269 * a1[8]
    - 20853 * a1[11]
    + 32216 * a1[12] == -4169825)

s.add(57612 * a1[11]
    + 47348 * a1[9]
    + 48719 * a1[8]
    + 9228 * a1[5]
    + 65196 * a1[4]
    + 36650 * a1[0]
    + 47566 * a1[1]
    - 33282 * a1[2]
    - 59180 * a1[3]
    - 59599 * a1[6]
    - 62888 * a1[7]
        - 37592 * a1[10]
        + 40510 * a1[12] == 4081505)

s.add(25633 * a1[11]
        + 25252 * a1[9]
    + 28153 * a1[8]
    + 26517 * a1[7]
        + 59511 * a1[4]
        + 4102 * a1[3]
    + 51735 * a1[0]
    + 35879 * a1[1]
    - 63890 * a1[2]
    - 21386 * a1[5]
    - 20769 * a1[6]
    - 43789 * a1[10]
    + 7314 * a1[12] == 1788229)

print s.check()
m = s.model()

for d in m.decls():
  print "%s = %s" %(d.name(),m[d])
```

- The following result is output if run the code.

python test.py

```
lazenca0x0@ubuntu:~/CTF/DEFCON2016/baby's/baby-re$ python test.py
sat
a1[3] = 104
a1[7] = 32
a1[9] = 97
a1[6] = 115
a1[0] = 77
a1[10] = 114
a1[12] = 33
a1[11] = 100
a1[8] = 104
a1[1] = 97
a1[4] = 32
a1[5] = 105
a1[2] = 116
```

- The output result is as below.

Flag string

a1[0]	77	**a1[7]**	32
a1[1]	97	**a1[8]**	104
a1[2]	116	**a1[9]**	97
a1[3]	104	**a1[10]**	114
a1[4]	32	**a1[11]**	100
a1[5]	105	**a1[12]**	33
a1[6]	115		

Find 13 values satisfying "==" operation(angr).

- This question is also solved using (angr)
- The source code is as below
    - The code is published on https://docs.angr.io/docs/examples.html.

solve.py

```
#!/usr/bin/env python2

"""
Author: David Manouchehri <manouchehri@protonmail.com>
DEFCON CTF Qualifier 2016
Challenge: baby-re
Team: hack.carleton
Write-up: http://hack.carleton.team/2016/05/21/defcon-ctf-qualifier-2016-baby-re/
Runtime: ~8 minutes (single threaded E5-2650L v3 @ 1.80GHz on DigitalOcean)

DigitalOcean is horrible for single threaded applications, I would highly suggest using something else.
"""
import angr

def main():
        proj = angr.Project('./baby-re',  load_options={'auto_load_libs': False})
        path_group = proj.factory.path_group(threads=4) # Doesn't really help to have more threads, but
whatever.
        path_group.explore(find=0x40294b, avoid=0x402941)
        return path_group.found[0].state.posix.dumps(1) # The flag is at the end.

def test():
        assert 'Math is hard!' in main()

if __name__ == '__main__':
        print(repr(main()))
```

anrg Example

- http://docs.angr.io/docs/examples.html
- https://github.com/angr/angr-doc/blob/master/examples/defcon2016quals_baby-re_0/solve.py

- The code is mainly about "path_group.explore()"
    - If every conditions of 13 if () are met, the program moves to "0x40294b".
        - Save the address in "find". (find=0x40294b)
    - If conditions are not met, the program moves to "0x402941"
        - Save the address in "avoid". (avoid=0x402941)

```
.text:0000000000402924 45 89 F1                          mov     r9d, r14d
.text:0000000000402927 45 89 E8                          mov     r8d, r13d
.text:000000000040292A 89 C6                             mov     esi, eax
.text:000000000040292C BF 88 2A 40 00                    mov     edi, offset aTheFlagIsCCCCC ; "The flag is: %c%c%c%c%c%c%c%c%c%c"...
.text:0000000000402931 B8 00 00 00 00                    mov     eax, 0
.text:0000000000402936 E8 45 DC FF FF                    call    _printf
.text:000000000040293B 48 83 C4 40                       add     rsp, 40h
.text:000000000040293F EB 0A                             jmp     short loc_40294B
.text:0000000000402941                          ; ---------------------------------------------------------------------------
.text:0000000000402941
.text:0000000000402941                          loc_402941:                             ; CODE XREF: main+300↑j
.text:0000000000402941 BF B1 2A 40 00                    mov     edi, offset s   ; "Wrong"
.text:0000000000402946 E8 15 DC FF FF                    call    _puts
.text:000000000040294B
.text:000000000040294B                          loc_40294B:                             ; CODE XREF: main+358↑j
.text:000000000040294B B8 00 00 00 00                    mov     eax, 0
.text:0000000000402950 48 8B 5D D8                       mov     rbx, [rbp+var_28]
.text:0000000000402954 64 48 33 1C 25 28 00 00+          xor     rbx, fs:28h
.text:000000000040295D 74 05                             jz      short loc_402964
.text:000000000040295F E8 0C DC FF FF                    call    ___stack_chk_fail
.text:0000000000402964                          ; ---------------------------------------------------------------------------
.text:0000000000402964
```

- You can get the following result after running the code.

python test3.py

```
$ mkvirtualenv angr
New python executable in angr/bin/python
Installing setuptools, pip...done.
(angr)$ python test3.
WARNING | 2016-08-08 00:37:13,869 | simuvex.plugins.symbolic_memory | Concretizing symbolic length. Much sad;
think about implementing.
WARNING | 2016-08-08 00:37:15,772 | simuvex.plugins.symbolic_memory | Concretizing symbolic length. Much sad;
think about implementing.
WARNING | 2016-08-08 00:37:18,286 | simuvex.plugins.symbolic_memory | Concretizing symbolic length. Much sad;
think about implementing.
WARNING | 2016-08-08 00:37:22,451 | simuvex.plugins.symbolic_memory | Concretizing symbolic length. Much sad;
think about implementing.
WARNING | 2016-08-08 00:37:28,217 | simuvex.plugins.symbolic_memory | Concretizing symbolic length. Much sad;
think about implementing.
WARNING | 2016-08-08 00:37:36,537 | simuvex.plugins.symbolic_memory | Concretizing symbolic length. Much sad;
think about implementing.
WARNING | 2016-08-08 00:37:49,255 | simuvex.plugins.symbolic_memory | Concretizing symbolic length. Much sad;
think about implementing.
WARNING | 2016-08-08 00:38:06,851 | simuvex.plugins.symbolic_memory | Concretizing symbolic length. Much sad;
think about implementing.
WARNING | 2016-08-08 00:38:28,900 | simuvex.plugins.symbolic_memory | Concretizing symbolic length. Much sad;
think about implementing.
WARNING | 2016-08-08 00:38:57,506 | simuvex.plugins.symbolic_memory | Concretizing symbolic length. Much sad;
think about implementing.
WARNING | 2016-08-08 00:39:29,892 | simuvex.plugins.symbolic_memory | Concretizing symbolic length. Much sad;
think about implementing.
WARNING | 2016-08-08 00:40:20,124 | simuvex.plugins.symbolic_memory | Concretizing symbolic length. Much sad;
think about implementing.
WARNING | 2016-08-08 00:41:10,223 | simuvex.plugins.symbolic_memory | Concretizing symbolic length. Much sad;
think about implementing.
'Var[0]: Var[1]: Var[2]: Var[3]: Var[4]: Var[5]: Var[6]: Var[7]: Var[8]: Var[9]: Var[10]: Var[11]: Var[12]: The
flag is: Math is hard!\n'
(angr)$
```

Exploit Code

Exploit code

```
from pwn import *

p = process("./baby-re")

def CharInput(ch):
        p.recvuntil(':')
        p.sendline(str(ch))

CharInput(77)
CharInput(97)
CharInput(116)
CharInput(104)
CharInput(32)
CharInput(105)
CharInput(115)
CharInput(32)
CharInput(104)
CharInput(97)
CharInput(114)
CharInput(100)
CharInput(33)

print p.recvuntil("The flag is:") + p.recv()
```

Flag

Flag	

Unknown macro: 'html'

Related Site

- http://docs.angr.io/docs/examples.html
- https://github.com/angr/angr-doc/blob/master/examples/defcon2016quals_baby-re_0/solve.py
- https://github.com/angr/angr-doc/blob/master/examples/defcon2016quals_baby-re_1/solve.py