|  |  |
| --- | --- |
|  |  |
|  | Microsoft Windows [Version 10.0.19042.1237]  (c) Microsoft Corporation. All rights reserved.  D:\one drive\OneDrive\Desktop\12190094\Assignment 1\bomb001>bash  //first input |
|  |  |
|  | tshering@TD:/mnt/d/one drive/OneDrive/Desktop/12190094/Assignment 1/bomb001$ gdb bomb |
|  | GNU gdb (Ubuntu 8.1.1-0ubuntu1) 8.1.1 |
|  | Copyright (C) 2021 Free Software Foundation, Inc. |
|  | License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html> |
|  | This is free software: you are free to change and redistribute it. |
|  | There is NO WARRANTY, to the extent permitted by law. |
|  | Type "show copying" and "show warranty" for details. |
|  | This GDB was configured as "x86\_64-linux-gnu". |
|  | Type "show configuration" for configuration details. |
|  | For bug reporting instructions, please see: |
|  | <https://www.gnu.org/software/gdb/bugs/>. |
|  | Find the GDB manual and other documentation resources online at: |
|  | <http://www.gnu.org/software/gdb/documentation/>. |
|  |  |
|  | For help, type "help". |
|  | Type "apropos word" to search for commands related to "word"... |
|  | Reading symbols from bomb... |
|  | (gdb) b phase\_4 |
|  | Breakpoint 1 at 0x40100e |
|  | (gdb) r answers.txt |
|  | Starting program: /mnt/d/one drive/OneDrive/Desktop/12190094/Assignment 1/bomb001/bomb answers.txt |
|  | [Thread debugging using libthread\_db enabled] |
|  | Using host libthread\_db library "/lib/x86\_64-linux-gnu/libthread\_db.so.1". |
|  | Welcome to my fiendish little bomb. You have 6 phases with |
|  | which to blow yourself up. Have a nice day! |
|  | Phase 1 defused. How about the next one? |
|  | That's number 2. Keep going! |
|  | Halfway there! |
|  |  |
|  |  |
|  | Breakpoint 1, 0x000000000040100e in phase\_4 () |
|  | (gdb) disas |
|  | Dump of assembler code for function phase\_4: |
|  | => 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | 0x0000000000401034 <+38>: cmp $0x2,%eax |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49> |
|  | 0x0000000000401039 <+43>: cmpl $0xe,(%rsp) |
|  | 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | 0x0000000000401056 <+72>: cmp $0x23,%eax |
|  | 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | (gdb) u\* 0x0000000000401034 |
|  | 0x0000000000401034 in phase\_4 () |
|  | (gdb) disas |
|  | Dump of assembler code for function phase\_4: |
|  | 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | => 0x0000000000401034 <+38>: cmp $0x2,%eax **;%eax is compared with value 2** |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49**>;if %eax and 2 is not equal then the bomb will explode.** |
|  | 0x0000000000401039 <+43>: cmpl $0xe,(%rsp) |
|  | 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | 0x0000000000401056 <+72>: cmp $0x23,%eax |
|  | 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | (gdb) ni |
|  | 0x0000000000401037 in phase\_4 () |
|  | **;Checking the value of %eax. %eax value is 2 so the bomb do not explode.** |
|  | (gdb) i r |
|  | rax 0x2 2**// %eax/%rax value is 2** |
|  | rbx 0x7fffffffdef8 140737488346872 |
|  | rcx 0x0 0 |
|  | rdx 0x17 23 |
|  | rsi 0x17 23 |
|  | rdi 0x7fffffffd760 140737488344928 |
|  | rbp 0x2 0x2 |
|  | rsp 0x7fffffffddb0 0x7fffffffddb0 |
|  | r8 0x0 0 |
|  | r9 0x0 0 |
|  | r10 0x7ffff7f3aac0 140737353329344 |
|  | r11 0x7ffff7f3b3c0 140737353331648 |
|  | r12 0x7fffffffdef8 140737488346872 |
|  | r13 0x400d56 4197718 |
|  | r14 0x0 0 |
|  | r15 0x7ffff7ffbc40 140737354120256 |
|  | rip 0x401037 0x401037 <phase\_4+41> |
|  | eflags 0x246 [ PF ZF IF ] |
|  | cs 0x33 51 |
|  | ss 0x2b 43 |
|  | ds 0x0 0 |
|  | es 0x0 0 |
|  | fs 0x0 0 |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | gs 0x0 0 |
|  | (gdb) ni |
|  | 0x0000000000401039 in phase\_4 () |
|  | (gdb) disas |
|  | Dump of assembler code for function phase\_4: |
|  | 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | 0x0000000000401034 <+38>: cmp $0x2,%eax |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49> |
|  | => 0x0000000000401039 <+43>: cmpl $0xe,(%rsp)//0xe = 14 |
|  | 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | 0x0000000000401056 <+72>: cmp $0x23,%eax |
|  | 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | (gdb) ni |
|  | 0x000000000040103d in phase\_4 () |
|  | **//First input should below/less than 0xe(14)** |
|  | (gdb) x/d $rsp |
|  | 0x7fffffffddb0: 2 |
|  | (gdb) disas |
|  | Dump of assembler code for function phase\_4: |
|  | 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | 0x0000000000401034 <+38>: cmp $0x2,%eax |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49> |
|  | 0x0000000000401039 <+43>: cmpl $0xe,(%rsp) |
|  | => 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | 0x0000000000401056 <+72>: cmp $0x23,%eax |
|  | 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | (gdb) ni |
|  | 0x0000000000401044 in phase\_4 () |
|  | (gdb) ni |
|  | 0x0000000000401049 in phase\_4 () |
|  | **//Since my first input is below 14, the bomb does not explod.** |
|  | (gdb) disas |
|  | Dump of assembler code for function phase\_4: |
|  | 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | 0x0000000000401034 <+38>: cmp $0x2,%eax |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49> |
|  | 0x0000000000401039 <+43>: cmpl $0xe,(%rsp) |
|  | 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | => 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | 0x0000000000401056 <+72>: cmp $0x23,%eax |
|  | 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | (gdb) ni |
|  | 0x000000000040104e in phase\_4 () |
|  | (gdb) disas |
|  | Dump of assembler code for function phase\_4: |
|  | 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | 0x0000000000401034 <+38>: cmp $0x2,%eax |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49> |
|  | 0x0000000000401039 <+43>: cmpl $0xe,(%rsp) |
|  | 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | => 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | 0x0000000000401056 <+72>: cmp $0x23,%eax |
|  | 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | (gdb) ni |
|  | 0x0000000000401051 in phase\_4 () |
|  | (gdb) ni |
|  | 0x0000000000401056 in phase\_4 () |
|  | (gdb) disas |
|  | Dump of assembler code for function phase\_4: |
|  | 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | 0x0000000000401034 <+38>: cmp $0x2,%eax |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49> |
|  | 0x0000000000401039 <+43>: cmpl $0xe,(%rsp) |
|  | 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | => 0x0000000000401056 <+72>: cmp $0x23,%eax//comparing eax value with 0x23. |
|  | 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | (gdb) ni |
|  | 0x0000000000401059 in phase\_4 () |
|  | (gdb) disas |
|  | Dump of assembler code for function phase\_4: |
|  | 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | 0x0000000000401034 <+38>: cmp $0x2,%eax |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49> |
|  | 0x0000000000401039 <+43>: cmpl $0xe,(%rsp) |
|  | 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | 0x0000000000401056 <+72>: cmp $0x23,%eax |
|  | => 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | **;Checking value of rax, rax is not equal to 0x23 and 35 (2\*16^1+3\*16^0=35 in decimal) as shown below so bomb will explode so therefore the input is wrong** |
|  | (gdb) i r |
|  | rax 0xd 13 **// rax value is 13** |
|  | rbx 0x7fffffffdef8 140737488346872 |
|  | rcx 0x0 0 |
|  | rdx 0x2 2 |
|  | rsi 0x2 2 |
|  | rdi 0x2 2 |
|  | rbp 0x2 0x2 |
|  | rsp 0x7fffffffddb0 0x7fffffffddb0 |
|  | r8 0x0 0 |
|  | r9 0x0 0 |
|  | r10 0x7ffff7f3aac0 140737353329344 |
|  | r11 0x7ffff7f3b3c0 140737353331648 |
|  | r12 0x7fffffffdef8 140737488346872 |
|  | r13 0x400d56 4197718 |
|  | r14 0x0 0 |
|  | r15 0x7ffff7ffbc40 140737354120256 |
|  | rip 0x401059 0x401059 <phase\_4+75> |
|  | eflags 0x283 [ CF SF IF ] |
|  | cs 0x33 51 |
|  | ss 0x2b 43 |
|  | ds 0x0 0 |
|  | es 0x0 0 |
|  | fs 0x0 0 |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | gs 0x0 0 |
|  | (gdb) disas |
|  | Dump of assembler code for function phase\_4: |
|  | 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | 0x0000000000401034 <+38>: cmp $0x2,%eax |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49> |
|  | 0x0000000000401039 <+43>: cmpl $0xe,(%rsp) |
|  | 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | 0x0000000000401056 <+72>: cmp $0x23,%eax |
|  | => 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | (gdb) |
|  |  |
|  | **;second input** |
|  |  |
|  | tshering@TD:/mnt/d/one drive/OneDrive/Desktop/12190094/Assignment 1/bomb001$ gdb bomb |
|  | GNU gdb (Ubuntu 8.1.1-0ubuntu1) 8.1.1 |
|  | Copyright (C) 2021 Free Software Foundation, Inc. |
|  | License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html> |
|  | This is free software: you are free to change and redistribute it. |
|  | There is NO WARRANTY, to the extent permitted by law. |
|  | Type "show copying" and "show warranty" for details. |
|  | This GDB was configured as "x86\_64-linux-gnu". |
|  | Type "show configuration" for configuration details. |
|  | For bug reporting instructions, please see: |
|  | <https://www.gnu.org/software/gdb/bugs/>. |
|  | Find the GDB manual and other documentation resources online at: |
|  | <http://www.gnu.org/software/gdb/documentation/>. |
|  |  |
|  | For help, type "help". |
|  | Type "apropos word" to search for commands related to "word"... |
|  | Reading symbols from bomb... |
|  | (gdb) b phase\_4 |
|  | Breakpoint 1 at 0x40100e |
|  | (gdb) r answers.txt |
|  | Starting program: /mnt/d/one drive/OneDrive/Desktop/12190094/Assignment 1/bomb001/bomb answers.txt |
|  | [Thread debugging using libthread\_db enabled] |
|  | Using host libthread\_db library "/lib/x86\_64-linux-gnu/libthread\_db.so.1". |
|  | Welcome to my fiendish little bomb. You have 6 phases with |
|  | which to blow yourself up. Have a nice day! |
|  | Phase 1 defused. How about the next one? |
|  | That's number 2. Keep going! |
|  | Halfway there! |
|  | 8 35//test input |
|  |  |
|  | Breakpoint 1, 0x000000000040100e in phase\_4 () |
|  | (gdb) disas |
|  | Dump of assembler code for function phase\_4: |
|  | => 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | 0x0000000000401034 <+38>: cmp $0x2,%eax |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49> |
|  | 0x0000000000401039 <+43>: cmpl $0xe,(%rsp) |
|  | 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | 0x0000000000401056 <+72>: cmp $0x23,%eax |
|  | 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | (gdb) u\* 0x0000000000401034 |
|  | 0x0000000000401034 in phase\_4 () |
|  | (gdb) i r |
|  | rax 0x2 2 **;eax value is 2** |
|  | rbx 0x7fffffffdef8 140737488346872 |
|  | rcx 0x0 0 |
|  | rdx 0x23 35 |
|  | rsi 0x23 35 |
|  | rdi 0x7fffffffd760 140737488344928 |
|  | rbp 0x2 0x2 |
|  | rsp 0x7fffffffddb0 0x7fffffffddb0 |
|  | r8 0x0 0 |
|  | r9 0x0 0 |
|  | r10 0x7ffff7f3aac0 140737353329344 |
|  | r11 0x7ffff7f3b3c0 140737353331648 |
|  | r12 0x7fffffffdef8 140737488346872 |
|  | r13 0x400d56 4197718 |
|  | r14 0x0 0 |
|  | r15 0x7ffff7ffbc40 140737354120256 |
|  | rip 0x401034 0x401034 <phase\_4+38> |
|  | eflags 0x206 [ PF IF ] |
|  | cs 0x33 51 |
|  | ss 0x2b 43 |
|  | ds 0x0 0 |
|  | es 0x0 0 |
|  | fs 0x0 0 |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | gs 0x0 0 |
|  | (gdb) disas |
|  | Dump of assembler code for function phase\_4: |
|  | 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | => 0x0000000000401034 <+38>: cmp $0x2,%eax **;eax or rax value is compared with 0x2 and we have already found out eax value is 2.** |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49> **;since it is equal, bomb does not explode** |
|  | 0x0000000000401039 <+43>: cmpl $0xe,(%rsp) |
|  | 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | 0x0000000000401056 <+72>: cmp $0x23,%eax |
|  | 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | (gdb) ni |
|  | 0x0000000000401037 in phase\_4 () |
|  | (gdb) ni |
|  | 0x0000000000401039 in phase\_4 () |
|  | (gdb) disas |
|  | Dump of assembler code for function phase\_4: |
|  | 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | 0x0000000000401034 <+38>: cmp $0x2,%eax |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49> |
|  | => 0x0000000000401039 <+43>: cmpl $0xe,(%rsp) |
|  | 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | 0x0000000000401056 <+72>: cmp $0x23,%eax |
|  | 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | **;Checking %rsp value, first input should be less than 0xe(14). The rsp value is 8, which is less than 14.** |
|  | (gdb) x/d $rsp |
|  | 0x7fffffffddb0: 8 |
|  | (gdb) ni |
|  | 0x000000000040103d in phase\_4 () |
|  | (gdb) disas |
|  | Dump of assembler code for function phase\_4: |
|  | 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | 0x0000000000401034 <+38>: cmp $0x2,%eax |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49> |
|  | 0x0000000000401039 <+43>: cmpl $0xe,(%rsp) |
|  | => 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | 0x0000000000401056 <+72>: cmp $0x23,%eax |
|  | 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | (gdb) ni |
|  | 0x0000000000401044 in phase\_4 () |
|  | **;Since %rsp is less than 0xe(14),the bomb doesn’t explode.** |
|  | (gdb) disas |
|  | Dump of assembler code for function phase\_4: |
|  | 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | 0x0000000000401034 <+38>: cmp $0x2,%eax |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49> |
|  | 0x0000000000401039 <+43>: cmpl $0xe,(%rsp) |
|  | 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | => 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | 0x0000000000401056 <+72>: cmp $0x23,%eax |
|  | 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | (gdb) ni |
|  | 0x0000000000401049 in phase\_4 () |
|  | (gdb) ni |
|  | 0x000000000040104e in phase\_4 () |
|  | (gdb) ni |
|  | 0x0000000000401051 in phase\_4 () |
|  | (gdb) disas |
|  | //function func4 is calling. |
|  | Dump of assembler code for function phase\_4: |
|  | 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | 0x0000000000401034 <+38>: cmp $0x2,%eax |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49> |
|  | 0x0000000000401039 <+43>: cmpl $0xe,(%rsp) |
|  | 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | => 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | 0x0000000000401056 <+72>: cmp $0x23,%eax |
|  | 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | (gdb) ni |
|  | 0x0000000000401056 in phase\_4 () |
|  | (gdb) ni |
|  |  |
|  | 0x0000000000401059 in phase\_4 () |
|  | (gdb) i r |
|  | rax 0x23 35 **;eax value is 0x23/35 so correct input** |
|  | rbx 0x7fffffffdef8 140737488346872 |
|  | rcx 0x0 0 |
|  | rdx 0x8 8 |
|  | rsi 0x8 8 |
|  | rdi 0x8 8 |
|  | rbp 0x2 0x2 |
|  | rsp 0x7fffffffddb0 0x7fffffffddb0 |
|  | r8 0x0 0 |
|  | r9 0x0 0 |
|  | r10 0x7ffff7f3aac0 140737353329344 |
|  | r11 0x7ffff7f3b3c0 140737353331648 |
|  | r12 0x7fffffffdef8 140737488346872 |
|  | r13 0x400d56 4197718 |
|  | r14 0x0 0 |
|  | r15 0x7ffff7ffbc40 140737354120256 |
|  | rip 0x401059 0x401059 <phase\_4+75> |
|  | eflags 0x246 [ PF ZF IF ] |
|  | cs 0x33 51 |
|  | ss 0x2b 43 |
|  | ds 0x0 0 |
|  | es 0x0 0 |
|  | fs 0x0 0 |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | gs 0x0 0 |
|  | (gdb) disas |
|  | Dump of assembler code for function phase\_4: |
|  | 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | 0x0000000000401034 <+38>: cmp $0x2,%eax |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49> |
|  | 0x0000000000401039 <+43>: cmpl $0xe,(%rsp) |
|  | 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | 0x0000000000401056 <+72>: cmp $0x23,%eax |
|  | => 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | (gdb) ni |
|  | 0x000000000040105b in phase\_4 () |
|  | **;Since the eax is equal to 0x23, the bomb doesn’t explode** |
|  | (gdb) disas |
|  | Dump of assembler code for function phase\_4: |
|  | 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | 0x0000000000401034 <+38>: cmp $0x2,%eax |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49> |
|  | 0x0000000000401039 <+43>: cmpl $0xe,(%rsp) |
|  | 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | 0x0000000000401056 <+72>: cmp $0x23,%eax |
|  | 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | => 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | **;Second input is 35** |
|  | (gdb) x/d 0x4+$rsp |
|  | 0x7fffffffddb4: 35 |
|  | (gdb) ni |
|  | 0x0000000000401060 in phase\_4 () |
|  | (gdb) ni |
|  | 0x0000000000401067 in phase\_4 () |
|  | **;Since the second input (35) is equal to 0x23. It bomb doesn’t explode.** |
|  | (gdb) disas |
|  | Dump of assembler code for function phase\_4: |
|  | 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | 0x0000000000401034 <+38>: cmp $0x2,%eax |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49> |
|  | 0x0000000000401039 <+43>: cmpl $0xe,(%rsp) |
|  | 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | 0x0000000000401056 <+72>: cmp $0x23,%eax |
|  | 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | => 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | (gdb) ni |
|  | 0x000000000040106c in phase\_4 () |
|  | (gdb) ni |
|  | 0x0000000000401075 in phase\_4 () |
|  | (gdb) ni |
|  | 0x000000000040107c in phase\_4 () |
|  | (gdb) disas |
|  | Dump of assembler code for function phase\_4: |
|  | 0x000000000040100e <+0>: sub $0x18,%rsp |
|  | 0x0000000000401012 <+4>: mov %fs:0x28,%rax |
|  | 0x000000000040101b <+13>: mov %rax,0x8(%rsp) |
|  | 0x0000000000401020 <+18>: xor %eax,%eax |
|  | 0x0000000000401022 <+20>: lea 0x4(%rsp),%rcx |
|  | 0x0000000000401027 <+25>: mov %rsp,%rdx |
|  | 0x000000000040102a <+28>: mov $0x4025cf,%esi |
|  | 0x000000000040102f <+33>: call 0x400bb0 <\_\_isoc99\_sscanf@plt> |
|  | 0x0000000000401034 <+38>: cmp $0x2,%eax |
|  | 0x0000000000401037 <+41>: jne 0x40103f <phase\_4+49> |
|  | 0x0000000000401039 <+43>: cmpl $0xe,(%rsp) |
|  | 0x000000000040103d <+47>: jbe 0x401044 <phase\_4+54> |
|  | 0x000000000040103f <+49>: call 0x40143d <explode\_bomb> |
|  | 0x0000000000401044 <+54>: mov $0xe,%edx |
|  | 0x0000000000401049 <+59>: mov $0x0,%esi |
|  | 0x000000000040104e <+64>: mov (%rsp),%edi |
|  | 0x0000000000401051 <+67>: call 0x400fdb <func4> |
|  | 0x0000000000401056 <+72>: cmp $0x23,%eax |
|  | 0x0000000000401059 <+75>: jne 0x401062 <phase\_4+84> |
|  | 0x000000000040105b <+77>: cmpl $0x23,0x4(%rsp) |
|  | 0x0000000000401060 <+82>: je 0x401067 <phase\_4+89> |
|  | 0x0000000000401062 <+84>: call 0x40143d <explode\_bomb> |
|  | --Type <RET> for more, q to quit, c to continue without paging-- |
|  | 0x0000000000401067 <+89>: mov 0x8(%rsp),%rax |
|  | 0x000000000040106c <+94>: xor %fs:0x28,%rax |
|  | 0x0000000000401075 <+103>: je 0x40107c <phase\_4+110> |
|  | 0x0000000000401077 <+105>: call 0x400b00 <\_\_stack\_chk\_fail@plt> |
|  | => 0x000000000040107c <+110>: add $0x18,%rsp |
|  | 0x0000000000401080 <+114>: ret |
|  | End of assembler dump. |
|  | (gdb) ni |
|  | 0x0000000000401080 in phase\_4 () |
|  | (gdb) ni |
|  | main (argc=<optimized out>, argv=<optimized out>) at bomb.c:96 |
|  | 96 phase\_defused(); |
|  | **;running the program again after removing the breakpoints, phase 4 gets defused.** |
|  | (gdb) r answers.txt |
|  | The program being debugged has been started already. |
|  | Start it from the beginning? (y or n) y |
|  | Starting program: /mnt/d/one drive/OneDrive/Desktop/12190094/Assignment 1/bomb001/bomb answers.txt |
|  | [Thread debugging using libthread\_db enabled] |
|  | Using host libthread\_db library "/lib/x86\_64-linux-gnu/libthread\_db.so.1". |
|  | Welcome to my fiendish little bomb. You have 6 phases with |
|  | which to blow yourself up. Have a nice day! |
|  | Phase 1 defused. How about the next one? |
|  | That's number 2. Keep going! |
|  | Halfway there! |
|  | 8 35 **; solution for phase 4 is 8 35** |
|  | So you got that one. Try this one. |
|  |  |
|  |  |
|  |  |