(gdb) b phase\_5

Breakpoint 1 at 0x401081

(gdb) disas phase\_5

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp // 18 bytes of stack frame is created

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax // eax = eax ^ eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx //rcx = rsp - 0x4

0x000000000040109a <+25>: mov %rsp,%rdx // move rsp in rdx : rdx = rsp

0x000000000040109d <+28>: mov $0x4025cf,%esi // esi = "%d %d"

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax // eax = rsp

0x00000000004010b4 <+51>: and $0xf,%eax // eax = 0xf & eax

0x00000000004010b7 <+54>: mov %eax,(%rsp) // rsp = eax

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx // move 0 to ecx

0x00000000004010c4 <+67>: mov $0x0,%edx // move 0 to edx

0x00000000004010c9 <+72>: add $0x1,%edx // edx = edx + 0x1

0x00000000004010cc <+75>: cltq // sign extend eax to quadword (rax)

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

--Type <RET> for more, q to quit, c to continue without paging--

0x00000000004010d7 <+86>: cmp $0xf,%eax // f = 15 is compare with eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx // if (edx != 15) explode\_bomb()

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114> if (ecx == \*(rsp + 4)) goto phase\_5+114

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp // rsp = rsp + 0x18

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) r answers.txt

Starting program: /home/ngawang/Desktop/Ngawang/CS I/Assignment 1/bomb001/bomb answers.txt

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!

So you got that one. Try this one.

5 15

Breakpoint 1, 0x0000000000401081 in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

=> 0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109>

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) u\* 0x00000000004010a7

0x00000000004010a7 in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

=> 0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx // move 0 to ecx

0x00000000004010c4 <+67>: mov $0x0,%edx // move 0 to edx

0x00000000004010c9 <+72>: add $0x1,%edx // edx = edx + 0x1

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) i r

rax 0x2 2

rbx 0x7fffffffdf48 140737488346952

rcx 0x0 0

rdx 0x7fffffffde34 140737488346676

rsi 0x0 0

rdi 0x7fffffffd7e0 140737488345056

rbp 0x0 0x0

rsp 0x7fffffffde30 0x7fffffffde30

r8 0xffffffff 4294967295

r9 0x0 0

r10 0x7ffff7f61ac0 140737353489088

r11 0x0 0

r12 0x400c60 4197472

r13 0x7fffffffdf40 140737488346944

r14 0x0 0

r15 0x0 0

rip 0x4010a7 0x4010a7 <phase\_5+38>

eflags 0x202 [ IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

k0 0x0 0

k1 0x0 0

k2 0x0 0

k3 0x0 0

k4 0x0 0

k5 0x0 0

k6 0x0 0

k7 0x0 0

(gdb) ni

0x00000000004010aa in phase\_5 ()

(gdb) ni

0x00000000004010b1 in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

=> 0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) ni

0x00000000004010b4 in phase\_5 ()

(gdb) ni

0x00000000004010b7 in phase\_5 ()

(gdb) ni

0x00000000004010ba in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

=> 0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) i r

rax 0x5 5

rbx 0x7fffffffdf48 140737488346952

rcx 0x0 0

rdx 0x7fffffffde34 140737488346676

rsi 0x0 0

rdi 0x7fffffffd7e0 140737488345056

rbp 0x0 0x0

rsp 0x7fffffffde30 0x7fffffffde30

r8 0xffffffff 4294967295

r9 0x0 0

r10 0x7ffff7f61ac0 140737353489088

r11 0x0 0

r12 0x400c60 4197472

r13 0x7fffffffdf40 140737488346944

r14 0x0 0

r15 0x0 0

rip 0x4010ba 0x4010ba <phase\_5+57>

eflags 0x206 [ PF IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

k0 0x0 0

k1 0x0 0

k2 0x0 0

k3 0x0 0

k4 0x0 0

k5 0x0 0

k6 0x0 0

k7 0x0 0

(gdb) ni

0x00000000004010bd in phase\_5 ()

(gdb) ni

0x00000000004010bf in phase\_5 ()

(gdb) ni

0x00000000004010c4 in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

=> 0x00000000004010c4 <+67>: mov $0x0,%edx

0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) ni

0x00000000004010c9 in phase\_5 ()

(gdb) ni

0x00000000004010cc in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

0x00000000004010c9 <+72>: add $0x1,%edx

=> 0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) ni

0x00000000004010ce in phase\_5 ()

(gdb) ni

0x00000000004010d5 in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

=> 0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) ni

0x00000000004010d7 in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

=> 0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) i r

rax 0xc 12

rbx 0x7fffffffdf48 140737488346952

rcx 0xc 12

rdx 0x1 1

rsi 0x0 0

rdi 0x7fffffffd7e0 140737488345056

rbp 0x0 0x0

rsp 0x7fffffffde30 0x7fffffffde30

r8 0xffffffff 4294967295

r9 0x0 0

r10 0x7ffff7f61ac0 140737353489088

r11 0x0 0

r12 0x400c60 4197472

r13 0x7fffffffdf40 140737488346944

r14 0x0 0

r15 0x0 0

rip 0x4010d7 0x4010d7 <phase\_5+86>

eflags 0x206 [ PF IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

k0 0x0 0

k1 0x0 0

k2 0x0 0

k3 0x0 0

k4 0x0 0

k5 0x0 0

k6 0x0 0

k7 0x0 0

(gdb) ni

0x00000000004010da in phase\_5 ()

(gdb) ni

0x00000000004010c9 in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

=> 0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) u\*0x00000000004010c9

0x00000000004010c9 in phase\_5 ()

(gdb) u\*0x00000000004010c9

0x00000000004010c9 in phase\_5 ()

(gdb) u\*0x00000000004010c9

0x00000000004010c9 in phase\_5 ()

(gdb) u\*0x00000000004010c9

0x00000000004010c9 in phase\_5 ()

(gdb) u\*0x00000000004010c9

0x00000000004010c9 in phase\_5 ()

(gdb) u\*0x00000000004010c9

0x00000000004010c9 in phase\_5 ()

(gdb) u\*0x00000000004010c9

0x00000000004010c9 in phase\_5 ()

(gdb) u\*0x00000000004010c9

0x00000000004010c9 in phase\_5 ()

(gdb) u\*0x00000000004010c9

0x00000000004010c9 in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

=> 0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) ni

0x00000000004010cc in phase\_5 ()

(gdb) ni

0x00000000004010ce in phase\_5 ()

(gdb) ni

0x00000000004010d5 in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

=> 0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) ni

0x00000000004010d7 in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

=> 0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) i r

rax 0x1 1

rbx 0x7fffffffdf48 140737488346952

rcx 0x4e 78

rdx 0xb 11

rsi 0x0 0

rdi 0x7fffffffd7e0 140737488345056

rbp 0x0 0x0

rsp 0x7fffffffde30 0x7fffffffde30

r8 0xffffffff 4294967295

r9 0x0 0

r10 0x7ffff7f61ac0 140737353489088

r11 0x0 0

r12 0x400c60 4197472

r13 0x7fffffffdf40 140737488346944

r14 0x0 0

r15 0x0 0

rip 0x4010d7 0x4010d7 <phase\_5+86>

eflags 0x206 [ PF IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

k0 0x0 0

k1 0x0 0

k2 0x0 0

k3 0x0 0

k4 0x0 0

k5 0x0 0

k6 0x0 0

k7 0x0 0

(gdb) u\*0x00000000004010c9

0x00000000004010c9 in phase\_5 ()

(gdb) u\*0x00000000004010c9

0x00000000004010c9 in phase\_5 ()

(gdb) u\*0x00000000004010c9

0x00000000004010c9 in phase\_5 ()

(gdb) u\*0x00000000004010c9

0x00000000004010c9 in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

=> 0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) ni

0x00000000004010cc in phase\_5 ()

(gdb) ni

0x00000000004010ce in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

=> 0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) ni

0x00000000004010d5 in phase\_5 ()

(gdb) ni

0x00000000004010d7 in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109>

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

=> 0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) i r

rax 0xf 15

rbx 0x7fffffffdf48 140737488346952

rcx 0x73 115

rdx 0xf 15

rsi 0x0 0

rdi 0x7fffffffd7e0 140737488345056

rbp 0x0 0x0

rsp 0x7fffffffde30 0x7fffffffde30

r8 0xffffffff 4294967295

r9 0x0 0

r10 0x7ffff7f61ac0 140737353489088

r11 0x0 0

r12 0x400c60 4197472

r13 0x7fffffffdf40 140737488346944

r14 0x0 0

r15 0x0 0

rip 0x4010d7 0x4010d7 <phase\_5+86>

eflags 0x212 [ AF IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

k0 0x0 0

k1 0x0 0

k2 0x0 0

k3 0x0 0

k4 0x0 0

k5 0x0 0

k6 0x0 0

k7 0x0 0

(gdb) ni

0x00000000004010da in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

=> 0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) ni

0x00000000004010dc in phase\_5 ()

(gdb) ni

0x00000000004010e3 in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

=> 0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) i r

rax 0xf 15

rbx 0x7fffffffdf48 140737488346952

rcx 0x73 115

rdx 0xf 15

rsi 0x0 0

rdi 0x7fffffffd7e0 140737488345056

rbp 0x0 0x0

rsp 0x7fffffffde30 0x7fffffffde30

r8 0xffffffff 4294967295

r9 0x0 0

r10 0x7ffff7f61ac0 140737353489088

r11 0x0 0

r12 0x400c60 4197472

r13 0x7fffffffdf40 140737488346944

r14 0x0 0

r15 0x0 0

rip 0x4010e3 0x4010e3 <phase\_5+98>

eflags 0x246 [ PF ZF IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

k0 0x0 0

k1 0x0 0

k2 0x0 0

k3 0x0 0

k4 0x0 0

k5 0x0 0

k6 0x0 0

k7 0x0 0

(gdb) ni

0x00000000004010e6 in phase\_5 ()

(gdb) ni

0x00000000004010e8 in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

=> 0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) p/x \* (int \*) ($rsp+0x4)

$1 = 0xf

(gdb) i r

rax 0xf 15

rbx 0x7fffffffdf48 140737488346952

rcx 0x73 115

rdx 0xf 15

rsi 0x0 0

rdi 0x7fffffffd7e0 140737488345056

rbp 0x0 0x0

rsp 0x7fffffffde30 0x7fffffffde30

r8 0xffffffff 4294967295

r9 0x0 0

r10 0x7ffff7f61ac0 140737353489088

r11 0x0 0

r12 0x400c60 4197472

r13 0x7fffffffdf40 140737488346944

r14 0x0 0

r15 0x0 0

rip 0x4010e8 0x4010e8 <phase\_5+103>

eflags 0x246 [ PF ZF IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

k0 0x0 0

k1 0x0 0

k2 0x0 0

k3 0x0 0

k4 0x0 0

k5 0x0 0

k6 0x0 0

k7 0x0 0

(gdb) r answers.txt

The program being debugged has been started already.

Start it from the beginning? (y or n) y

Starting program: /home/ngawang/Desktop/Ngawang/CS I/Assignment 1/bomb001/bomb answers.txt

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!

So you got that one. Try this one.

5 115

Breakpoint 1, 0x0000000000401081 in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

=> 0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) u\*0x00000000004010e8

0x00000000004010e8 in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

=> 0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) p/x \* (int \*) ($rsp+0x4)

$2 = 0x73

(gdb) i r

rax 0xf 15

rbx 0x7fffffffdf48 140737488346952

rcx 0x73 115

rdx 0xf 15

rsi 0x0 0

rdi 0x7fffffffd7e0 140737488345056

rbp 0x0 0x0

rsp 0x7fffffffde30 0x7fffffffde30

r8 0xffffffff 4294967295

r9 0x0 0

r10 0x7ffff7f61ac0 140737353489088

r11 0x0 0

r12 0x400c60 4197472

r13 0x7fffffffdf40 140737488346944

r14 0x0 0

r15 0x0 0

rip 0x4010e8 0x4010e8 <phase\_5+103>

eflags 0x246 [ PF ZF IF ]

cs 0x33 51

ss 0x2b 43

ds 0x0 0

es 0x0 0

fs 0x0 0

gs 0x0 0

k0 0x0 0

k1 0x0 0

k2 0x0 0

k3 0x0 0

k4 0x0 0

k5 0x0 0

k6 0x0 0

k7 0x0 0

(gdb) ni

0x00000000004010ec in phase\_5 ()

(gdb) ni

0x00000000004010f3 in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

=> 0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) ni

0x00000000004010f8 in phase\_5 ()

(gdb) ni

0x0000000000401101 in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx

0x00000000004010c4 <+67>: mov $0x0,%edx

0x00000000004010c9 <+72>: add $0x1,%edx

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

=> 0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) ni

0x0000000000401108 in phase\_5 ()

(gdb) ni

0x000000000040110c in phase\_5 ()

(gdb) disas

Dump of assembler code for function phase\_5:

0x0000000000401081 <+0>: sub $0x18,%rsp // create stack frame

0x0000000000401085 <+4>: mov %fs:0x28,%rax

0x000000000040108e <+13>: mov %rax,0x8(%rsp)

0x0000000000401093 <+18>: xor %eax,%eax

0x0000000000401095 <+20>: lea 0x4(%rsp),%rcx

0x000000000040109a <+25>: mov %rsp,%rdx

0x000000000040109d <+28>: mov $0x4025cf,%esi

0x00000000004010a2 <+33>: callq 0x400bb0 <\_\_isoc99\_sscanf@plt>

0x00000000004010a7 <+38>: cmp $0x1,%eax

0x00000000004010aa <+41>: jg 0x4010b1 <phase\_5+48>

0x00000000004010ac <+43>: callq 0x40143d <explode\_bomb>

0x00000000004010b1 <+48>: mov (%rsp),%eax

0x00000000004010b4 <+51>: and $0xf,%eax

0x00000000004010b7 <+54>: mov %eax,(%rsp)

0x00000000004010ba <+57>: cmp $0xf,%eax

0x00000000004010bd <+60>: je 0x4010ee <phase\_5+109> // if(0xf != eax) then bomb explode

0x00000000004010bf <+62>: mov $0x0,%ecx // ecx = 0

0x00000000004010c4 <+67>: mov $0x0,%edx //edx = 0

0x00000000004010c9 <+72>: add $0x1,%edx //edx = edx + 0x1

0x00000000004010cc <+75>: cltq

0x00000000004010ce <+77>: mov 0x402480(,%rax,4),%eax

0x00000000004010d5 <+84>: add %eax,%ecx

0x00000000004010d7 <+86>: cmp $0xf,%eax

0x00000000004010da <+89>: jne 0x4010c9 <phase\_5+72> // if (eax != 0xf) goto phase\_5+72

0x00000000004010dc <+91>: movl $0xf,(%rsp)

0x00000000004010e3 <+98>: cmp $0xf,%edx

0x00000000004010e6 <+101>: jne 0x4010ee <phase\_5+109>

0x00000000004010e8 <+103>: cmp 0x4(%rsp),%ecx

0x00000000004010ec <+107>: je 0x4010f3 <phase\_5+114>

0x00000000004010ee <+109>: callq 0x40143d <explode\_bomb>

0x00000000004010f3 <+114>: mov 0x8(%rsp),%rax

0x00000000004010f8 <+119>: xor %fs:0x28,%rax

0x0000000000401101 <+128>: je 0x401108 <phase\_5+135>

0x0000000000401103 <+130>: callq 0x400b00 <\_\_stack\_chk\_fail@plt>

0x0000000000401108 <+135>: add $0x18,%rsp

=> 0x000000000040110c <+139>: retq

End of assembler dump.

(gdb) ni

main (argc=<optimized out>, argv=<optimized out>) at bomb.c:102

102 phase\_defused();

(gdb)