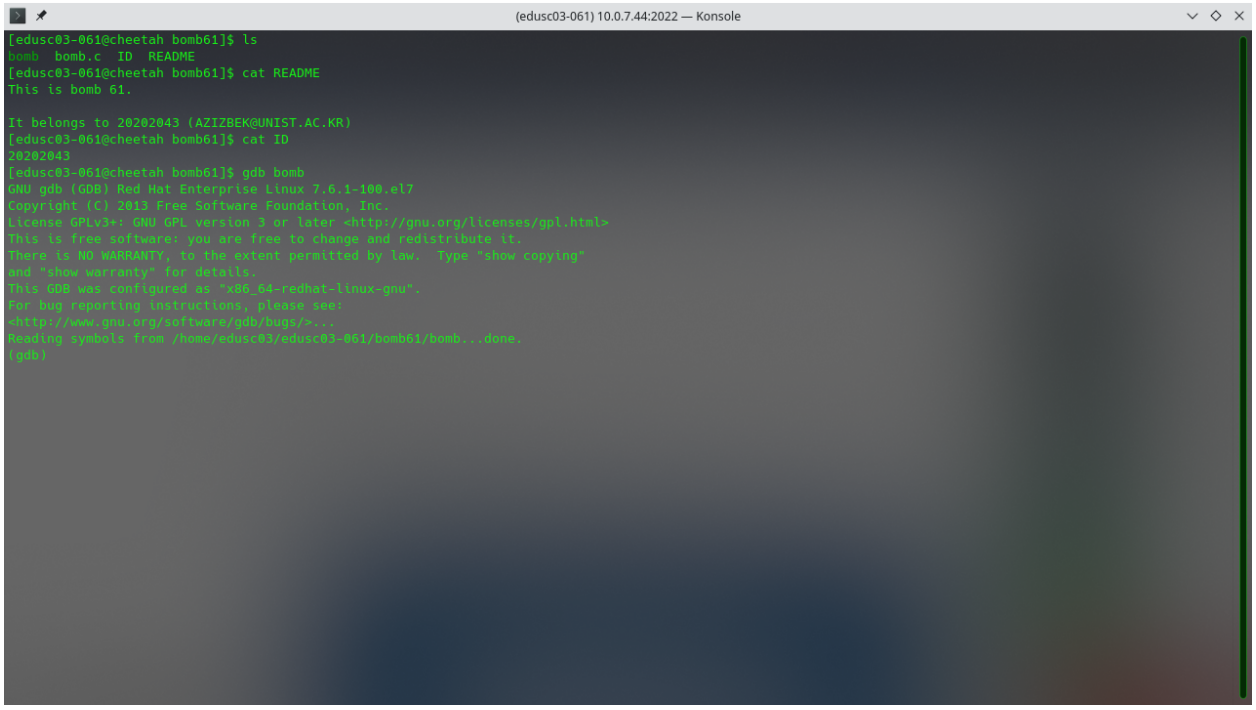
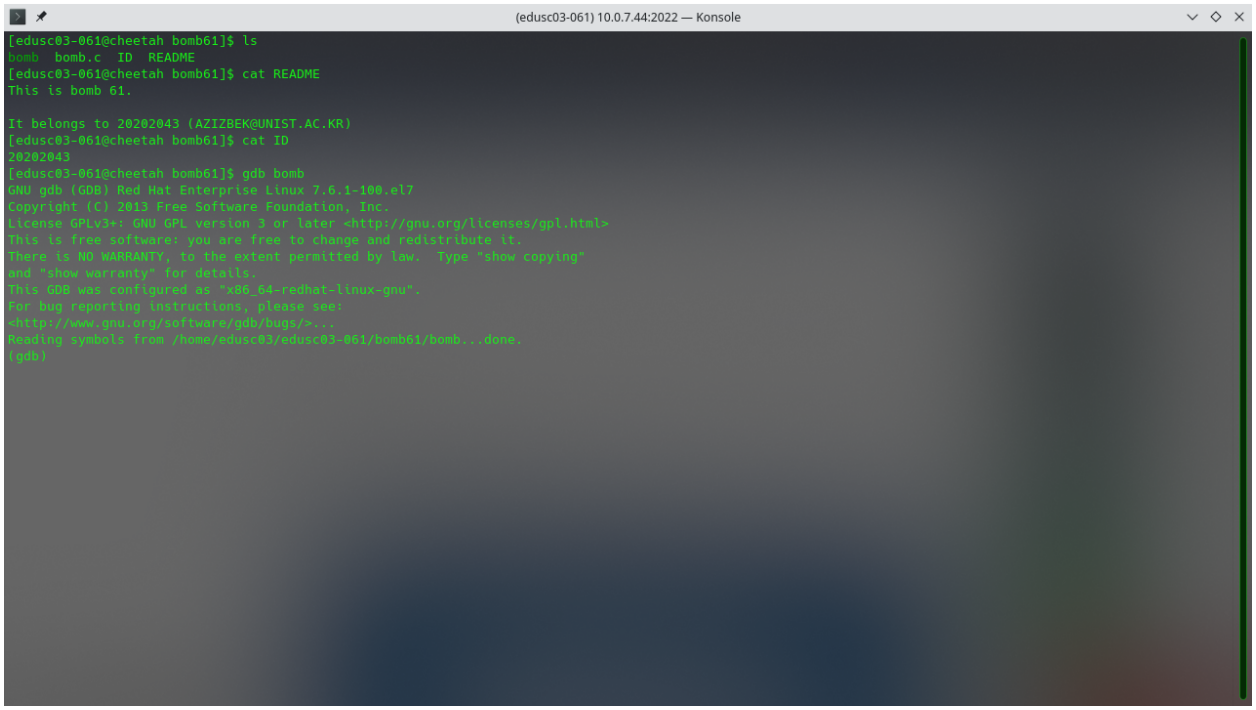
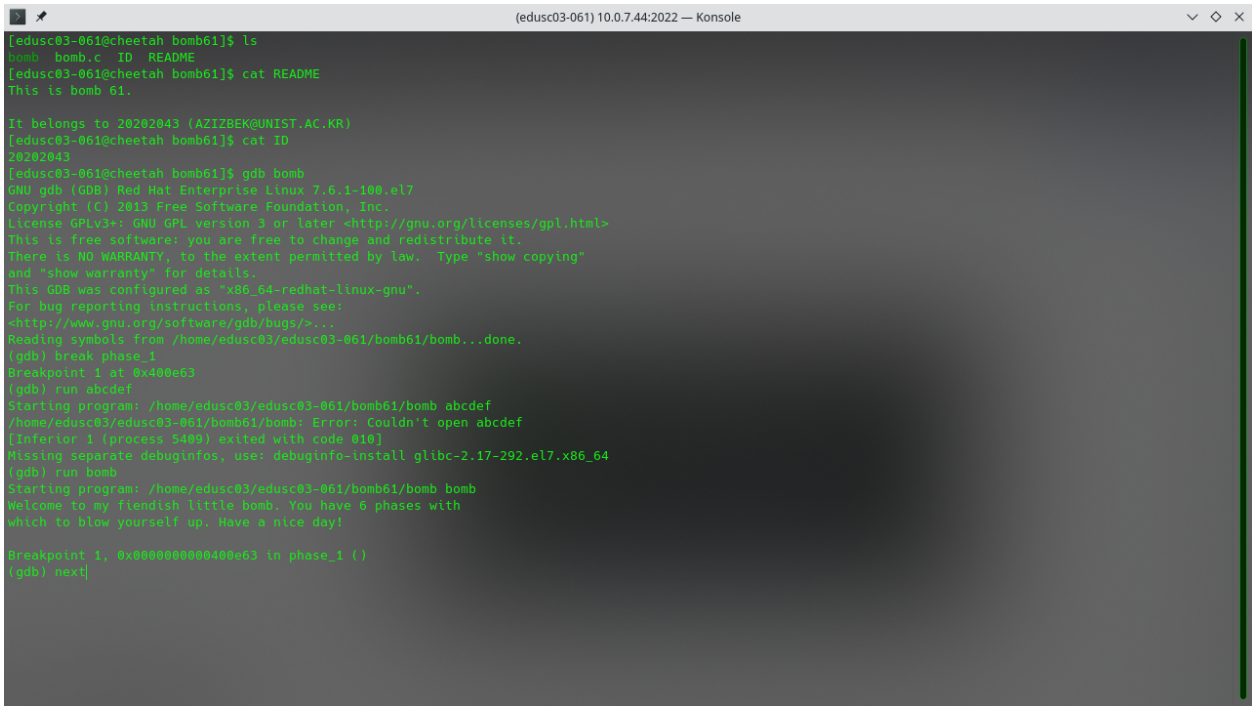
Phase 0

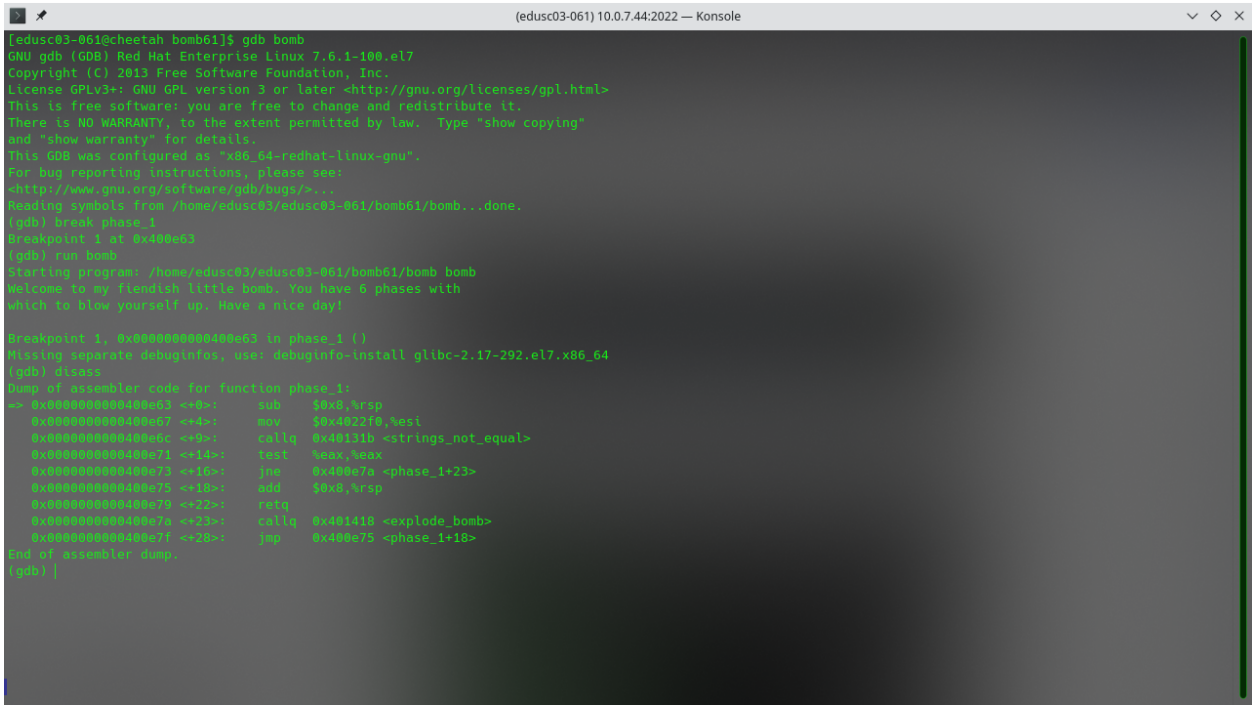


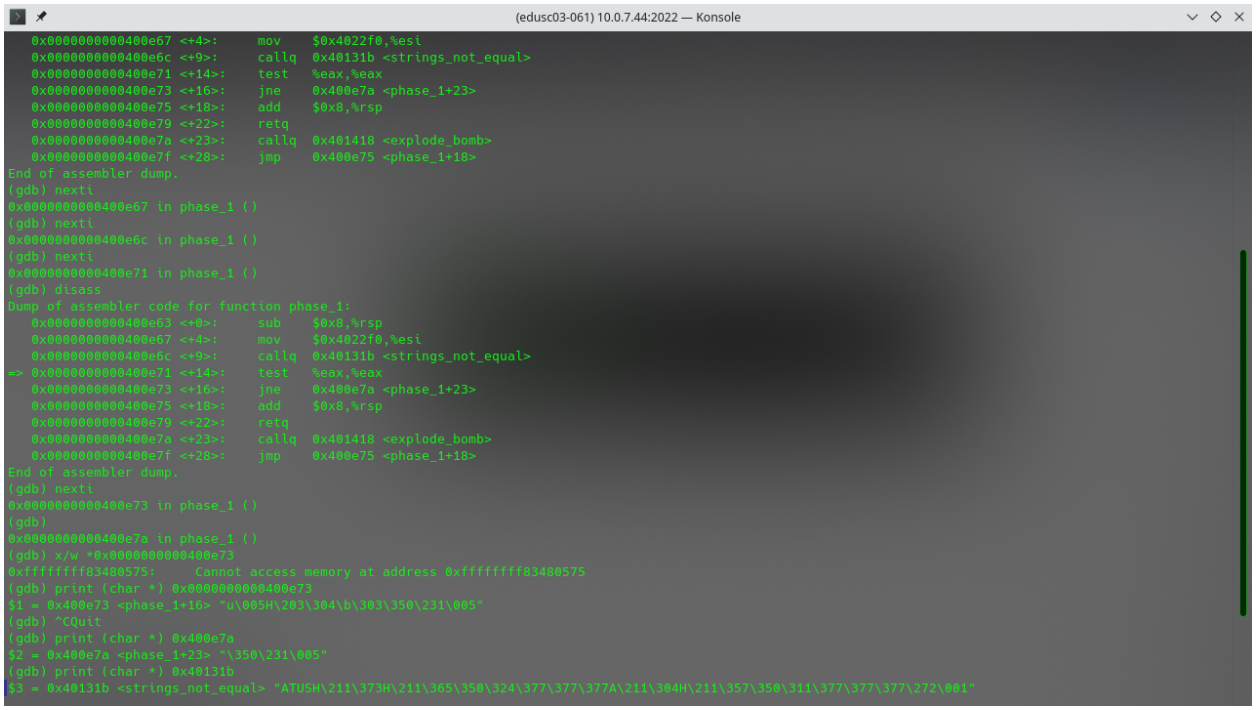
Phase 1

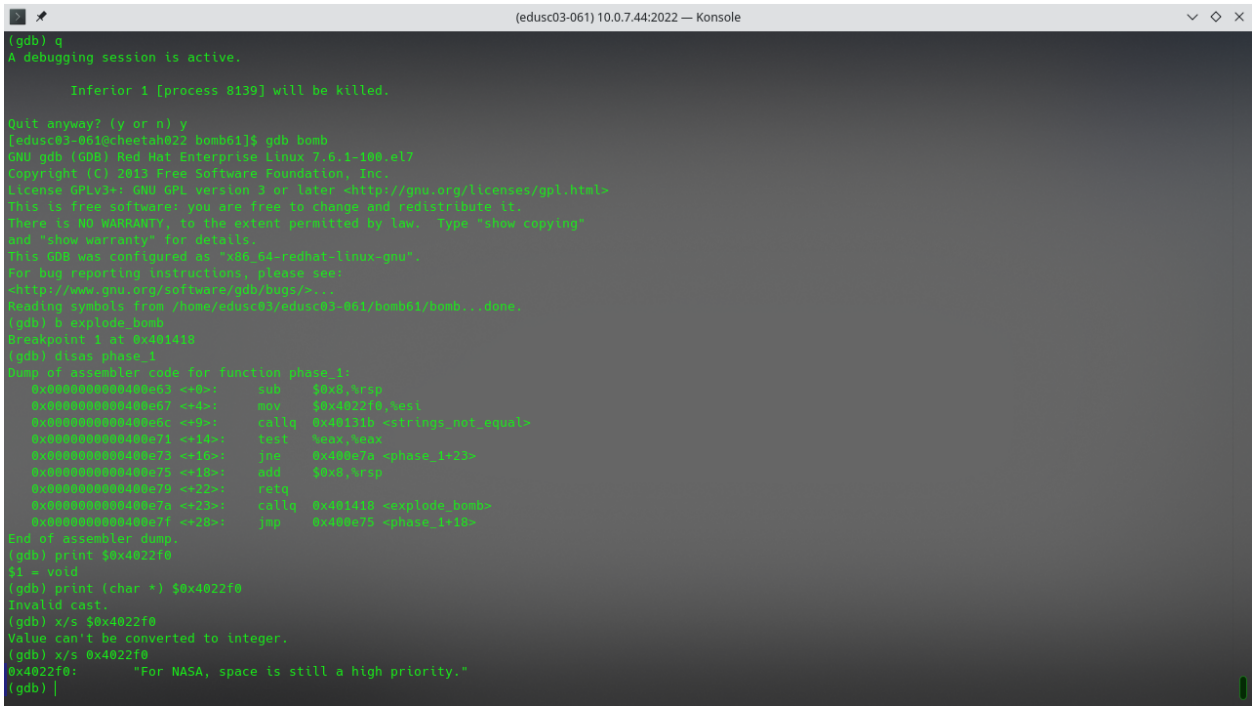


Set breakpoint at phase_1

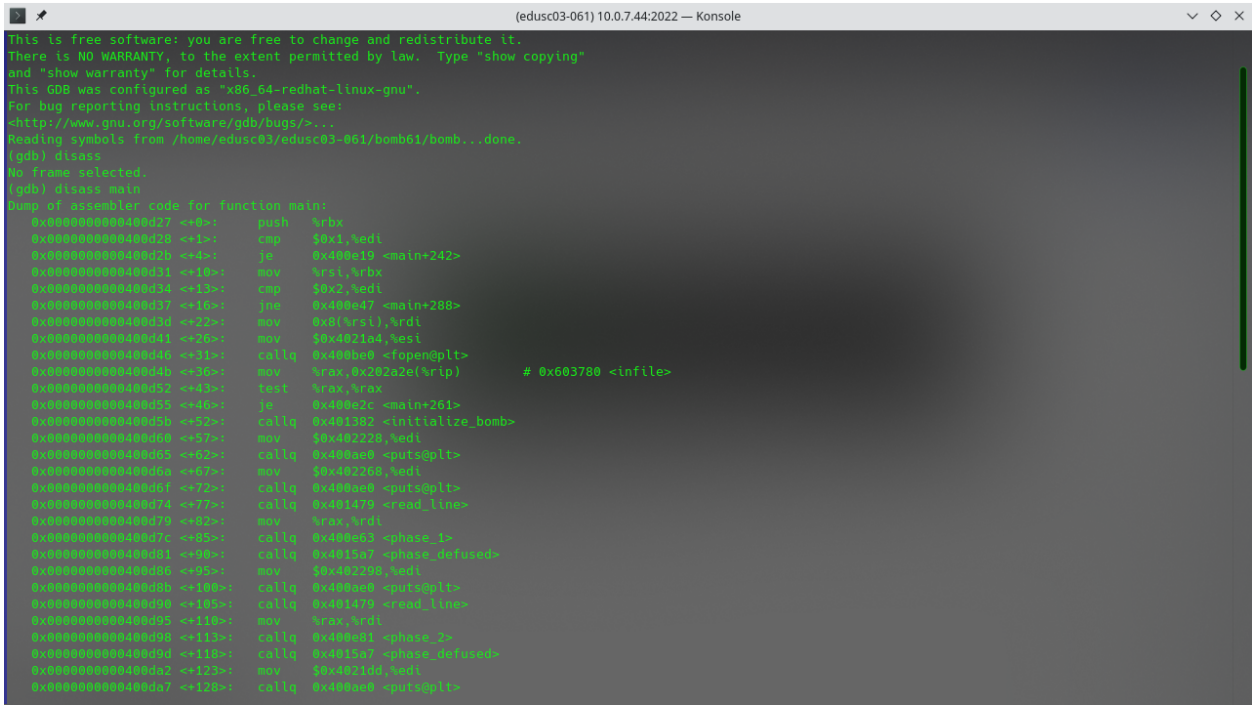


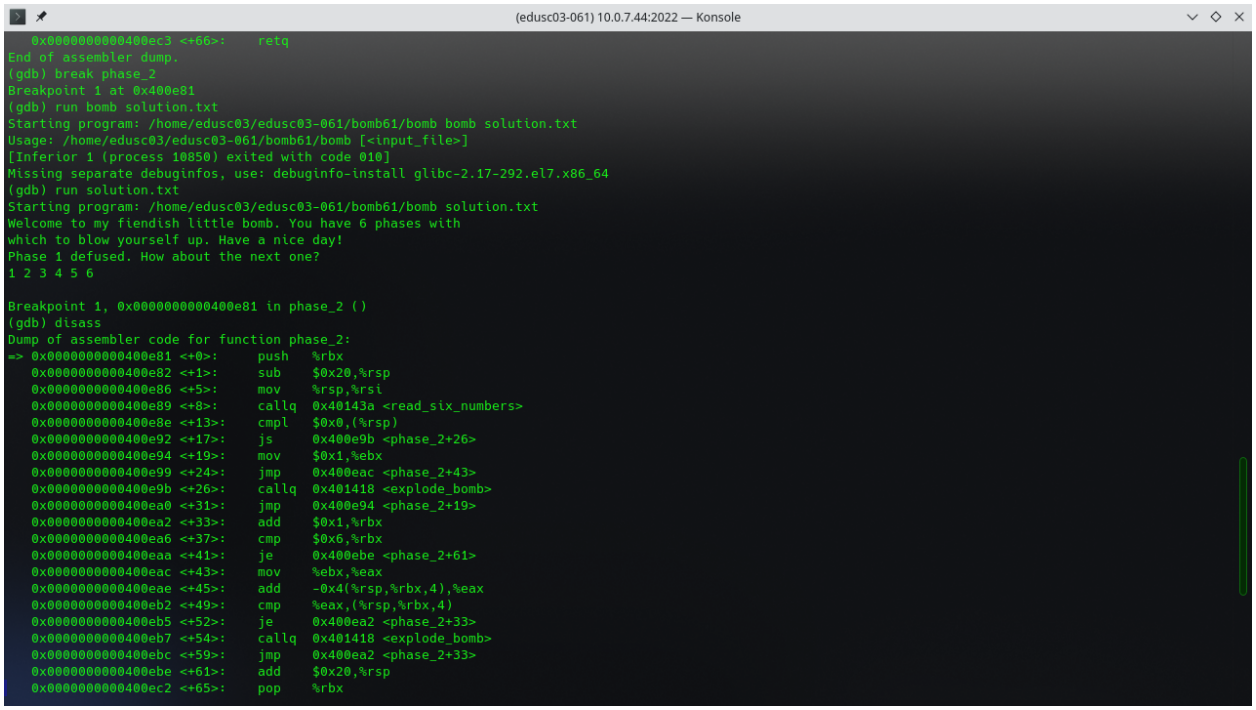


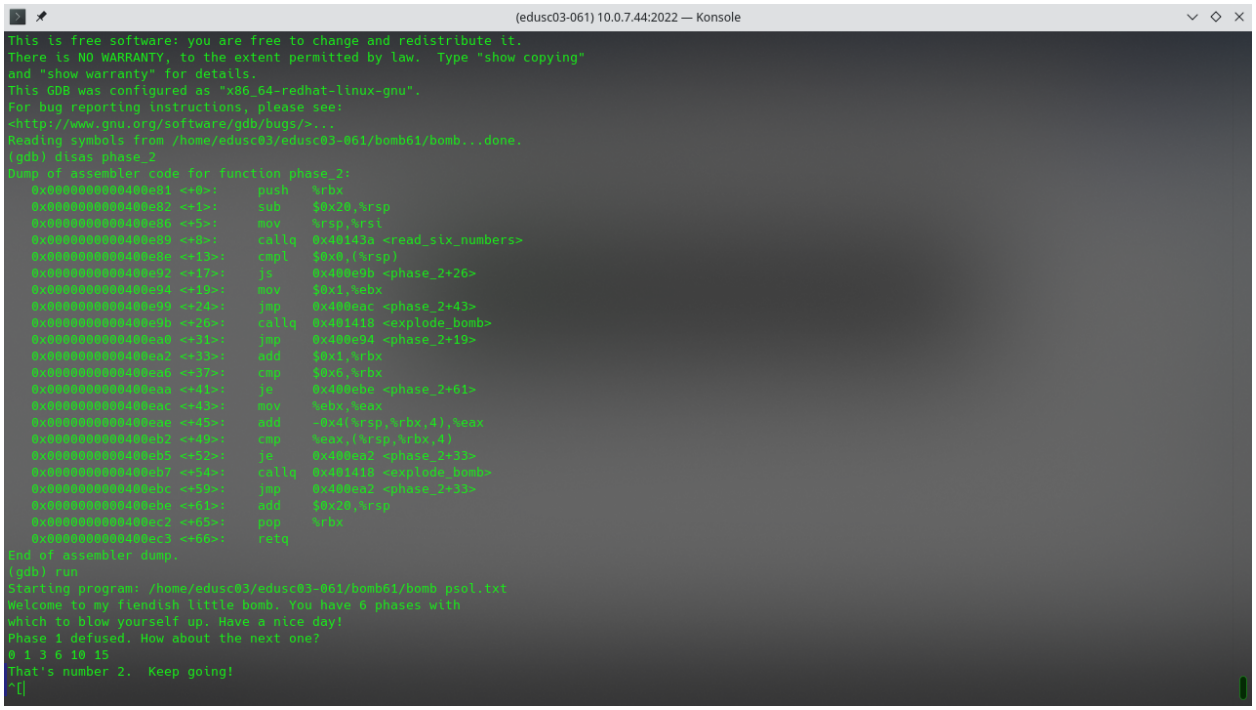

(edusc03-061) 10.0.7.44:2022 — Konsole
0x0000000000400e67 <+4>: mov $0x4022f0,%esi
0x0000000000400e6c <+9>: callq 0x40131b <strings_not_equal>
0x0000000000400e71 <+14>: test %eax,%eax
0x0000000000400e73 <+16>: jne 0x400e7a <phase_1+23>
0x0000000000400e75 <+18>: add $0x8,%rsp
0x0000000000400e79 <+22>: retq
0x0000000000400e7a <+23>: callq 0x401418 <explode_bomb>
0x0000000000400e7f <+28>: jmp 0x400e75 <phase_1+18>
End of assembler dump.
(gdb) nexti
0x0000000000400e67 in phase_1 ()
(gdb) nexti
0x0000000000400e6c in phase_1 ()
(gdb) nexti
0x0000000000400e71 in phase_1 ()
(gdb) disass
Dump of assembler code for function phase_1:
0x0000000000400e63 <+0>: sub $0x8,%rsp
0x0000000000400e67 <+4>: mov $0x4022f0,%esi
0x0000000000400e6c <+9>: callq 0x40131b <strings_not_equal>
=> 0x0000000000400e71 <+14>: test %eax,%eax
0x0000000000400e73 <+16>: jne 0x400e7a <phase_1+23>
0x0000000000400e75 <+18>: add $0x8,%rsp
0x0000000000400e79 <+22>: retq
0x0000000000400e7a <+23>: callq 0x401418 <explode_bomb>
0x0000000000400e7f <+28>: jmp 0x400e75 <phase_1+18>
End of assembler dump.
(gdb) nexti
0x0000000000400e73 in phase_1 ()
(gdb)
0x0000000000400e7a in phase_1 ()
(gdb) x/w *0x0000000000400e73
0xffffffff83480575: Cannot access memory at address 0xffffffff83480575
(gdb) print (char *) 0x0000000000400e73
$1 = 0x400e73 <phase_1+16> "u\005H\203\304\b\303\350\231\005"
(gdb) ^CQuit
(gdb) print (char *) 0x400e7a
$2 = 0x400e7a <phase_1+23> "\350\231\005"
(gdb) print (char *) 0x40131b
$3 = 0x40131b <strings_not_equal> "ATUSH\211\373H\211\365\350\324\377\377\377A\211\304H\211\357\350\311\377\377\377\272\001"


(edusc03-061) 10.0.7.44:2022 — Konsole
(gdb) q
A debugging session is active.
Inferior 1 [process 8139] will be killed.
Quit anyway? (y or n) y
[edusc03-061@cheetah022 bomb61]$ gdb bomb
GNU gdb (GDB) Red Hat Enterprise Linux 7.6.1-100.el7
Copyright (C) 2013 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-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/edusc03/edusc03-061/bomb61/bomb...done.
(gdb) b explode_bomb
Breakpoint 1 at 0x401418
(gdb) disas phase_1
Dump of assembler code for function phase_1:
0x0000000000400e63 <+0>: sub $0x8,%rsp
0x0000000000400e67 <+4>: mov $0x4022f0,%esi
0x0000000000400e6c <+9>: callq 0x40131b <strings_not_equal>
0x0000000000400e71 <+14>: test %eax,%eax
0x0000000000400e73 <+16>: jne 0x400e7a <phase_1+23>
0x0000000000400e75 <+18>: add $0x8,%rsp
0x0000000000400e79 <+22>: retq
0x0000000000400e7a <+23>: callq 0x401418 <explode_bomb>
0x0000000000400e7f <+28>: jmp 0x400e75 <phase_1+18>
End of assembler dump.
(gdb) print $0x4022f0
$1 = void
(gdb) print (char *) $0x4022f0
Invalid cast.
(gdb) x/s $0x4022f0
Value can't be converted to integer.
(gdb) x/s 0x4022f0
0x4022f0: "For NASA, space is still a high priority."
(gdb)

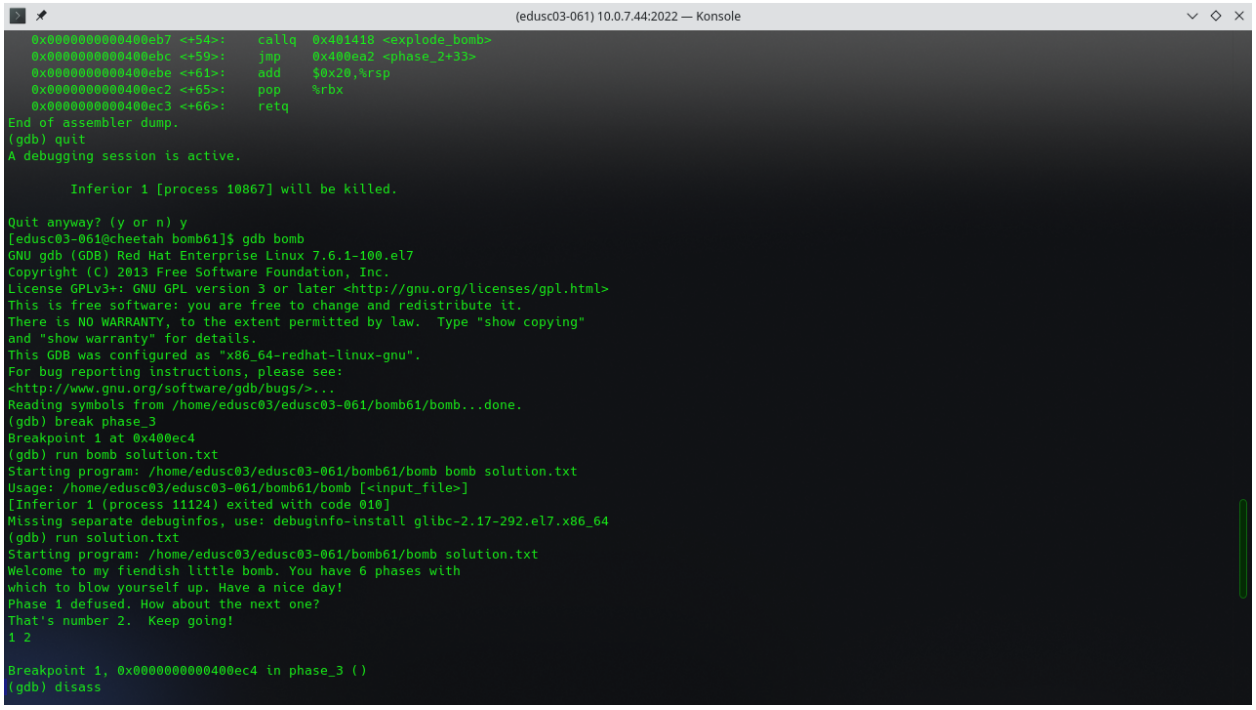
Phase 2


(edusc03-061) 10.0.7.44:2022 — Konsole
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-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/edusc03/edusc03-061/bomb61/bomb...done.
(gdb) disass
No frame selected.
(gdb) disass main
Dump of assembler code for function main:
0x0000000000400d27 <+0>: push %rbx
0x0000000000400d28 <+1>: cmp $0x1,%edi
0x0000000000400d2b <+4>: je 0x400e19 <main+242>
0x0000000000400d31 <+10>: mov %rsi,%rbx
0x0000000000400d34 <+13>: cmp $0x2,%edi
0x0000000000400d37 <+16>: jne 0x400e47 <main+288>
0x0000000000400d3d <+22>: mov 0x8(%rsi),%rdi
0x0000000000400d41 <+26>: mov $0x4021a4,%esi
0x0000000000400d46 <+31>: callq 0x400be0 <fopen@plt>
0x0000000000400d4b <+36>: mov %rax,0x202a2e(%rip) # 0x603780 <infile>
0x0000000000400d52 <+43>: test %rax,%rax
0x0000000000400d55 <+46>: je 0x400e2c <main+261>
0x0000000000400d5b <+52>: callq 0x401382 <initialize_bomb>
0x0000000000400d60 <+57>: mov $0x402228,%edi
0x0000000000400d65 <+62>: callq 0x400ae0 <puts@plt>
0x0000000000400d6a <+67>: mov $0x402268,%edi
0x0000000000400d6f <+72>: callq 0x400ae0 <puts@plt>
0x0000000000400d74 <+77>: callq 0x401479 <read_line>
0x0000000000400d79 <+82>: mov %rax,%rdi
0x0000000000400d7c <+85>: callq 0x400e63 <phase_1>
0x0000000000400d81 <+90>: callq 0x4015a7 <phase_defused>
0x0000000000400d86 <+95>: mov $0x402298,%edi
0x0000000000400d8b <+100>: callq 0x400ae0 <puts@plt>
0x0000000000400d90 <+105>: callq 0x401479 <read_line>
0x0000000000400d95 <+110>: mov %rax,%rdi
0x0000000000400d98 <+113>: callq 0x400e81 <phase_2>
0x0000000000400d9d <+118>: callq 0x4015a7 <phase_defused>
0x0000000000400da2 <+123>: mov $0x4021dd,%edi
0x0000000000400da7 <+128>: callq 0x400ae0 <puts@plt>
(edusc03-061) 10.0.7.44:2022 — Konsole
0x0000000000400e5e <+311>: callq 0x400c00 <exit@plt>
End of assembler dump.
(gdb) quit
[edusc03-061@cheetah bomb61]$ gdb bomb
GNU gdb (GDB) Red Hat Enterprise Linux 7.6.1-100.el7
Copyright (C) 2013 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-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/edusc03/edusc03-061/bomb61/bomb...done.
(gdb) disass phase_2
Dump of assembler code for function phase_2:
0x0000000000400e81 <+0>: push %rbx
0x0000000000400e82 <+1>: sub $0x20,%rsp
0x0000000000400e86 <+5>: mov %rsp,%rsi
0x0000000000400e89 <+8>: callq 0x40143a <read_six_numbers>
0x0000000000400e8e <+13>: cmpl $0x0,(%rsp)
0x0000000000400e92 <+17>: js 0x400e9b <phase_2+26>
0x0000000000400e94 <+19>: mov $0x1,%ebx
0x0000000000400e99 <+24>: jmp 0x400eac <phase_2+43>
0x0000000000400e9b <+26>: callq 0x401418 <explode_bomb>
0x0000000000400ea0 <+31>: jmp 0x400e94 <phase_2+19>
0x0000000000400ea2 <+33>: add $0x1,%rbx
0x0000000000400ea6 <+37>: cmp $0x6,%rbx
0x0000000000400eaa <+41>: je 0x400ebe <phase_2+61>
0x0000000000400eac <+43>: mov %ebx,%eax
0x0000000000400eae <+45>: add -0x4(%rsp,%rbx,4),%eax
0x0000000000400eb2 <+49>: cmp %eax,(%rsp,%rbx,4)
0x0000000000400eb5 <+52>: je 0x400ea2 <phase_2+33>
0x0000000000400eb7 <+54>: callq 0x401418 <explode_bomb>
0x0000000000400ebc <+59>: jmp 0x400ea2 <phase_2+33>
0x0000000000400ebe <+61>: add $0x20,%rsp
0x0000000000400ec2 <+65>: pop %rbx
0x0000000000400ec3 <+66>: retq
End of assembler dump.
(gdb)


(edusc03-061) 10.0.7.44:2022 — Konsole
0x0000000000400ec3 <+66>: retq
End of assembler dump.
(gdb) break phase_2
Breakpoint 1 at 0x400e81
(gdb) run bomb solution.txt
Starting program: /home/edusc03/edusc03-061/bomb61/bomb bomb solution.txt
Usage: /home/edusc03/edusc03-061/bomb61/bomb [<input_file>]
[Inferior 1 (process 10850) exited with code 010]
Missing separate debuginfos, use: debuginfo-install glibc-2.17-292.el7.x86_64
(gdb) run solution.txt
Starting program: /home/edusc03/edusc03-061/bomb61/bomb solution.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?
1 2 3 4 5 6
Breakpoint 1, 0x0000000000400e81 in phase_2 ()
(gdb) disass
Dump of assembler code for function phase_2:
=> 0x0000000000400e81 <+0>: push %rbx
0x0000000000400e82 <+1>: sub $0x20,%rsp
0x0000000000400e86 <+5>: mov %rsp,%rsi
0x0000000000400e89 <+8>: callq 0x40143a <read_six_numbers>
0x0000000000400e8e <+13>: cmpl $0x0,(%rsp)
0x0000000000400e92 <+17>: js 0x400e9b <phase_2+26>
0x0000000000400e94 <+19>: mov $0x1,%ebx
0x0000000000400e99 <+24>: jmp 0x400eac <phase_2+43>
0x0000000000400e9b <+26>: callq 0x401418 <explode_bomb>
0x0000000000400ea0 <+31>: jmp 0x400e94 <phase_2+19>
0x0000000000400ea2 <+33>: add $0x1,%rbx
0x0000000000400ea6 <+37>: cmp $0x6,%rbx
0x0000000000400eaa <+41>: je 0x400ebe <phase_2+61>
0x0000000000400eac <+43>: mov %ebx,%eax
0x0000000000400eae <+45>: add -0x4(%rsp,%rbx,4),%eax
0x0000000000400eb2 <+49>: cmp %eax,(%rsp,%rbx,4)
0x0000000000400eb5 <+52>: je 0x400ea2 <phase_2+33>
0x0000000000400eb7 <+54>: callq 0x401418 <explode_bomb>
0x0000000000400ebc <+59>: jmp 0x400ea2 <phase_2+33>
0x0000000000400ebe <+61>: add $0x20,%rsp
0x0000000000400ec2 <+65>: pop %rbx
0x0000000000400ec3 <+66>: retq
End of assembler dump.
(gdb) stepi
0x0000000000400e82 in phase_2 ()
(gdb) disass
Dump of assembler code for function phase_2:
0x0000000000400e81 <+0>: push %rbx
=> 0x0000000000400e82 <+1>: sub $0x20,%rsp
0x0000000000400e86 <+5>: mov %rsp,%rsi
0x0000000000400e89 <+8>: callq 0x40143a <read_six_numbers>
0x0000000000400e8e <+13>: cmpl $0x0,(%rsp)
0x0000000000400e92 <+17>: js 0x400e9b <phase_2+26>
0x0000000000400e94 <+19>: mov $0x1,%ebx
0x0000000000400e99 <+24>: jmp 0x400eac <phase_2+43>
0x0000000000400e9b <+26>: callq 0x401418 <explode_bomb>
0x0000000000400ea0 <+31>: jmp 0x400e94 <phase_2+19>
0x0000000000400ea2 <+33>: add $0x1,%rbx
0x0000000000400ea6 <+37>: cmp $0x6,%rbx
0x0000000000400eaa <+41>: je 0x400ebe <phase_2+61>
0x0000000000400eac <+43>: mov %ebx,%eax
0x0000000000400eae <+45>: add -0x4(%rsp,%rbx,4),%eax
0x0000000000400eb2 <+49>: cmp %eax,(%rsp,%rbx,4)
0x0000000000400eb5 <+52>: je 0x400ea2 <phase_2+33>
0x0000000000400eb7 <+54>: callq 0x401418 <explode_bomb>
0x0000000000400ebc <+59>: jmp 0x400ea2 <phase_2+33>
0x0000000000400ebe <+61>: add $0x20,%rsp
0x0000000000400ec2 <+65>: pop %rbx
0x0000000000400ec3 <+66>: retq
End of assembler dump.
(gdb)


(edusc03-061) 10.0.7.44:2022 — Konsole
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-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/edusc03/edusc03-061/bomb61/bomb...done.
(gdb) disas phase_2
Dump of assembler code for function phase_2:
   0x0000000000400e81 <+0>:   push   %rbx
   0x0000000000400e82 <+1>:   sub $0x20,%rsp
   0x0000000000400e86 <+5>:   mov   %rsp,%rsi
   0x0000000000400e89 <+8>:   callq 0x40143a <read_six_numbers>
   0x0000000000400e8e <+13>:   cmpl $0x0,(%rsp)
   0x0000000000400e92 <+17>:   js   0x400e9b <phase_2+26>
   0x0000000000400e94 <+19>:   mov $0x1,%ebx
   0x0000000000400e99 <+24>:   jmp   0x400eac <phase_2+43>
   0x0000000000400e9b <+26>:   callq 0x401418 <explode_bomb>
   0x0000000000400ea0 <+31>:   jmp   0x400e94 <phase_2+19>
   0x0000000000400ea2 <+33>:   add $0x1,%rbx
   0x0000000000400ea6 <+37>:   cmp $0x6,%rbx
   0x0000000000400eaa <+41>:   je   0x400ebe <phase_2+61>
   0x0000000000400eac <+43>:   mov   %ebx,%eax
   0x0000000000400eae <+45>:   add   -0x4(%rsp,%rbx,4),%eax
   0x0000000000400eb2 <+49>:   cmp   %eax,(%rsp,%rbx,4)
   0x0000000000400eb5 <+52>:   je   0x400ea2 <phase_2+33>
   0x0000000000400eb7 <+54>:   callq 0x401418 <explode_bomb>
   0x0000000000400ebc <+59>:   jmp   0x400ea2 <phase_2+33>
   0x0000000000400ebe <+61>:   add $0x20,%rsp
   0x0000000000400ec2 <+65>:   pop   %rbx
   0x0000000000400ec3 <+66>:   retq
End of assembler dump.
(gdb) run
Starting program: /home/edusc03/edusc03-061/bomb61/bomb psol.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?
0 1 3 6 10 15
That's number 2. Keep going!
^[]

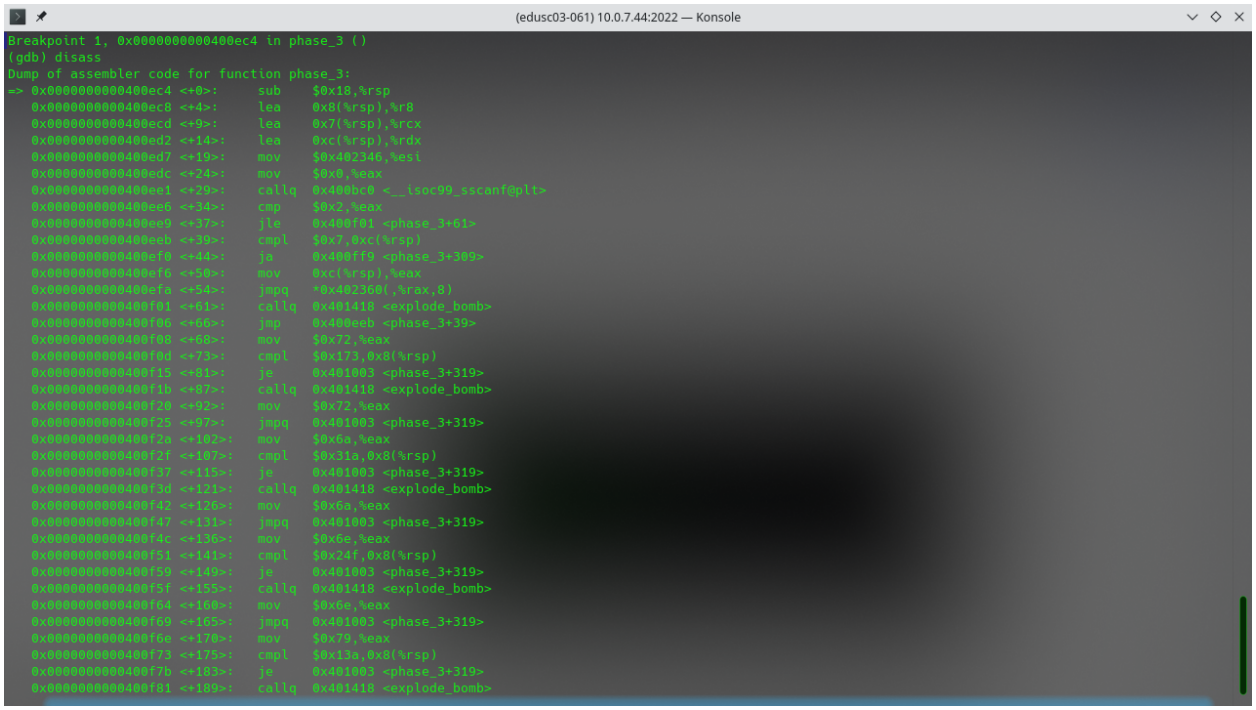
Phase 3

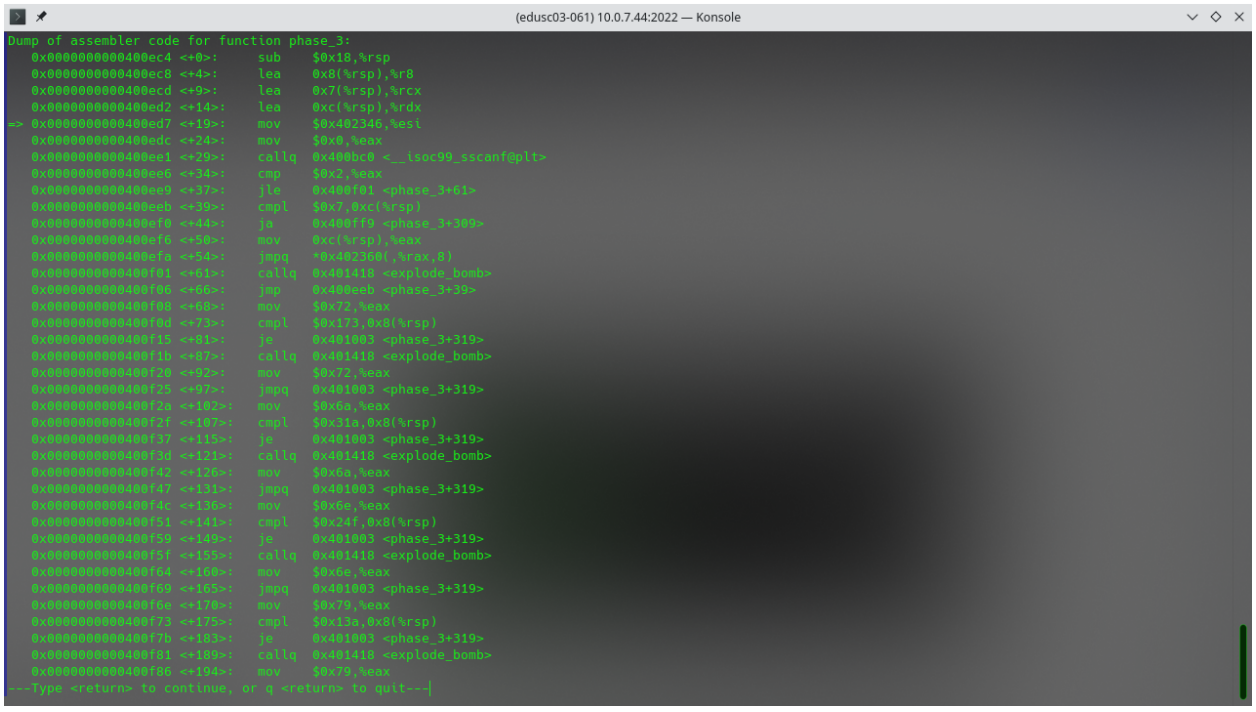

(edusc03-061) 10.0.7.44:2022 — Konsole
   0x0000000000400eb7 <+54>:   callq 0x401418 <explode_bomb>
   0x0000000000400ebc <+59>:   jmp   0x400ea2 <phase_2+33>
   0x0000000000400ebe <+61>:   add $0x20,%rsp
   0x0000000000400ec2 <+65>:   pop   %rbx
   0x0000000000400ec3 <+66>:   retq
End of assembler dump.
(gdb) quit
A debugging session is active.

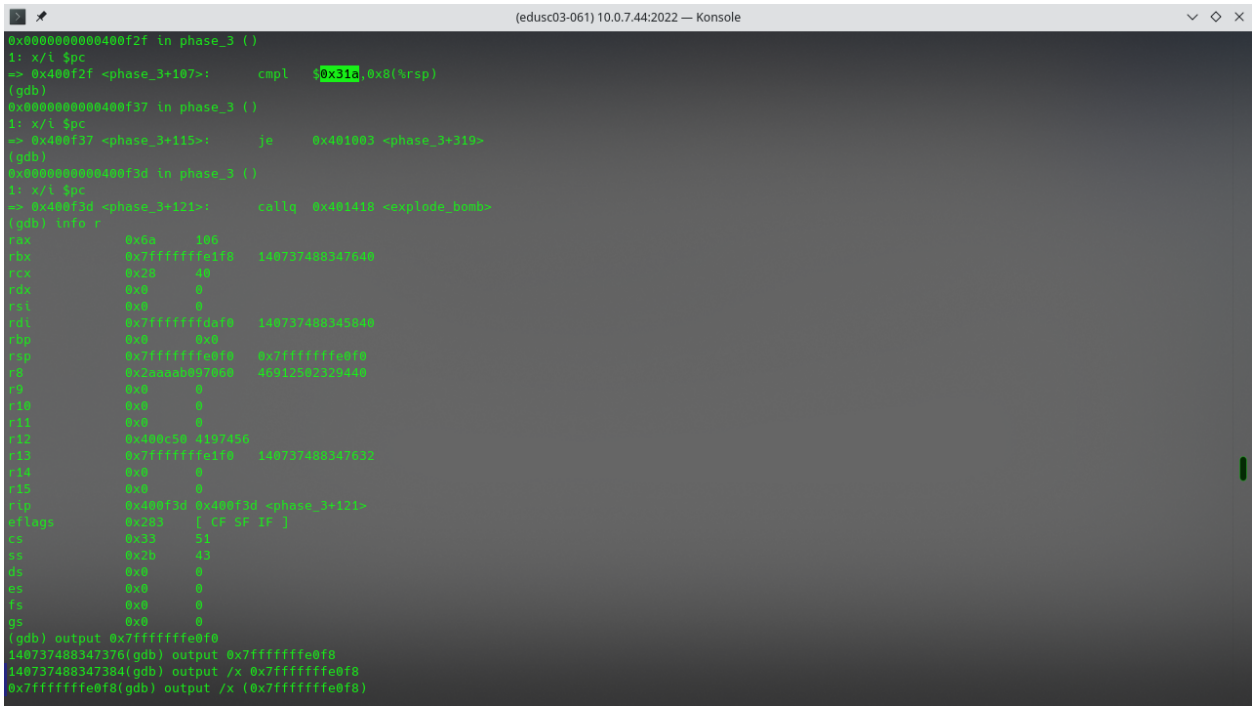
        Inferior 1 [process 10867] will be killed.

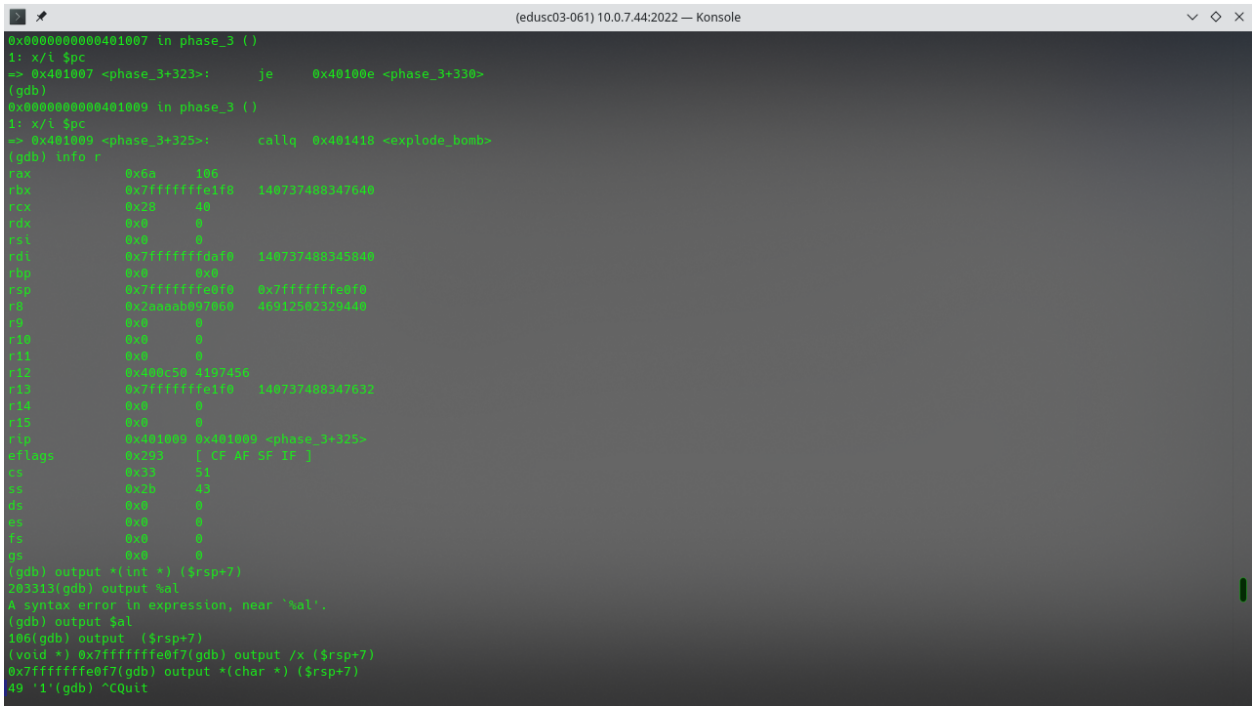
Quit anyway? (y or n) y
[edusc03-061@cheetah bomb61]$ gdb bomb
GNU gdb (GDB) Red Hat Enterprise Linux 7.6.1-100.el7
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute 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-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/edusc03/edusc03-061/bomb61/bomb...done.
(gdb) break phase_3
Breakpoint 1 at 0x400ec4
(gdb) run bomb solution.txt
Starting program: /home/edusc03/edusc03-061/bomb61/bomb bomb solution.txt
Usage: /home/edusc03/edusc03-061/bomb61/bomb [<input_file>]
[Inferior 1 (process 11124) exited with code 010]
Missing separate debuginfos, use: debuginfo-install glibc-2.17-292.el7.x86_64
(gdb) run solution.txt
Starting program: /home/edusc03/edusc03-061/bomb61/bomb solution.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!
1 2

Breakpoint 1, 0x0000000000400ec4 in phase_3 ()
(gdb) disass

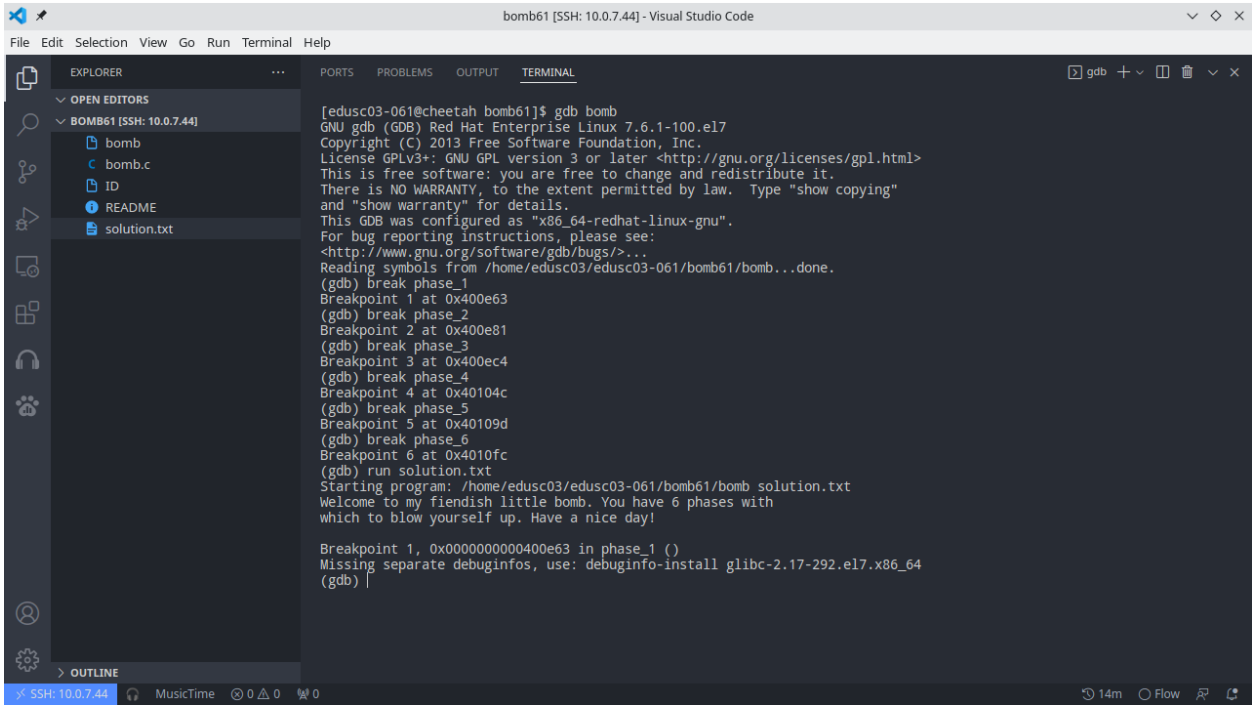

(edusc03-061) 10.0.7.44:2022 — Konsole
Breakpoint 1, 0x0000000000400ec4 in phase_3 ()
(gdb) disass
Dump of assembler code for function phase_3:
=> 0x0000000000400ec4 <+0>: sub $0x18,%rsp
0x0000000000400ec8 <+4>: lea 0x8(%rsp),%r8
0x0000000000400ecd <+9>: lea 0x7(%rsp),%rcx
0x0000000000400ed2 <+14>: lea 0xc(%rsp),%rdx
0x0000000000400ed7 <+19>: mov $0x402346,%esi
0x0000000000400edc <+24>: mov $0x0,%eax
0x0000000000400ee1 <+29>: callq 0x400bc0 <__isoc99_sscanf@plt>
0x0000000000400ee6 <+34>: cmp $0x2,%eax
0x0000000000400ee9 <+37>: jle 0x400f01 <phase_3+61>
0x0000000000400eeb <+39>: cmpl $0x7,0xc(%rsp)
0x0000000000400ef0 <+44>: ja 0x400ff9 <phase_3+309>
0x0000000000400ef6 <+50>: mov 0xc(%rsp),%eax
0x0000000000400efa <+54>: jmpq *0x402360(,%rax,8)
0x0000000000400f01 <+61>: callq 0x401418 <explode_bomb>
0x0000000000400f06 <+66>: jmp 0x400eeb <phase_3+39>
0x0000000000400f08 <+68>: mov $0x72,%eax
0x0000000000400f0d <+73>: cmpl $0x173,0x8(%rsp)
0x0000000000400f15 <+81>: je 0x401003 <phase_3+319>
0x0000000000400f1b <+87>: callq 0x401418 <explode_bomb>
0x0000000000400f20 <+92>: mov $0x72,%eax
0x0000000000400f25 <+97>: jmpq 0x401003 <phase_3+319>
0x0000000000400f2a <+102>: mov $0x6a,%eax
0x0000000000400f2f <+107>: cmpl $0x31a,0x8(%rsp)
0x0000000000400f37 <+115>: je 0x401003 <phase_3+319>
0x0000000000400f3d <+121>: callq 0x401418 <explode_bomb>
0x0000000000400f42 <+126>: mov $0x6a,%eax
0x0000000000400f47 <+131>: jmpq 0x401003 <phase_3+319>
0x0000000000400f4c <+136>: mov $0x6e,%eax
0x0000000000400f51 <+141>: cmpl $0x24f,0x8(%rsp)
0x0000000000400f59 <+149>: je 0x401003 <phase_3+319>
0x0000000000400f5f <+155>: callq 0x401418 <explode_bomb>
0x0000000000400f64 <+160>: mov $0x6e,%eax
0x0000000000400f69 <+165>: jmpq 0x401003 <phase_3+319>
0x0000000000400f6e <+170>: mov $0x79,%eax
0x0000000000400f73 <+175>: cmpl $0x13a,0x8(%rsp)
0x0000000000400f7b <+183>: je 0x401003 <phase_3+319>
0x0000000000400f81 <+189>: callq 0x401418 <explode_bomb>

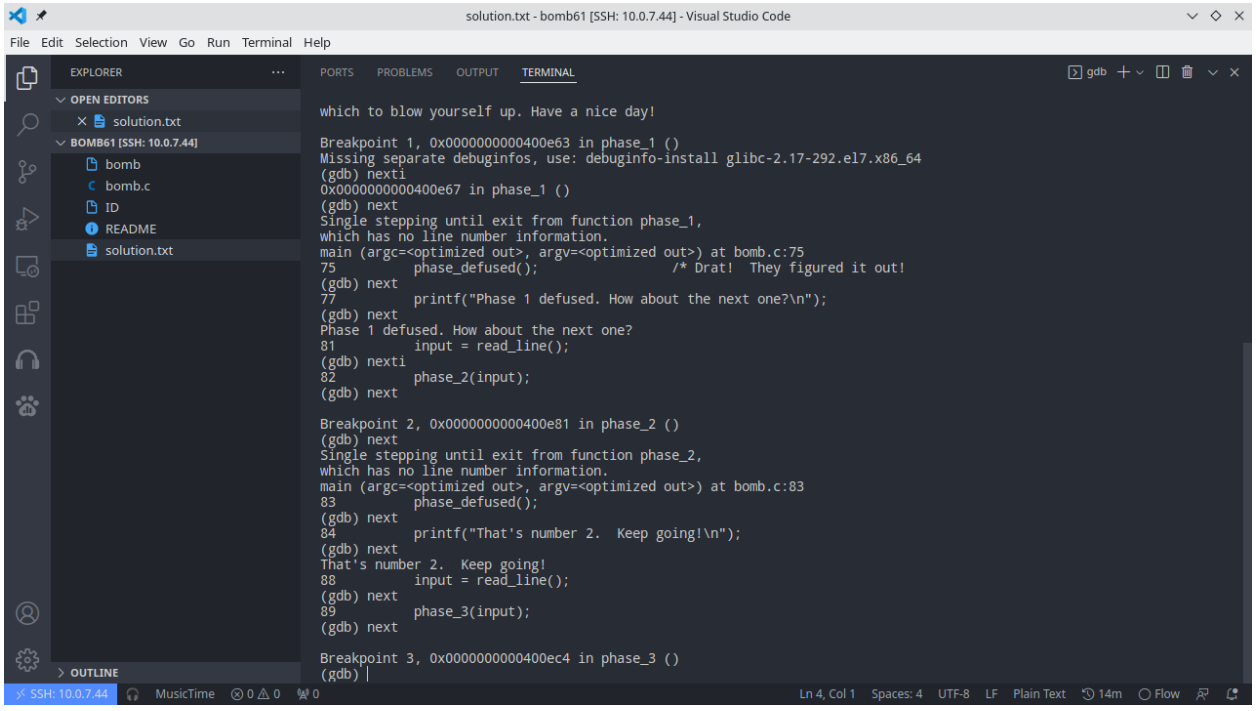

(edusc03-061) 10.0.7.44:2022 — Konsole
Dump of assembler code for function phase_3:
0x0000000000400ec4 <+0>: sub $0x18,%rsp
0x0000000000400ec8 <+4>: lea 0x8(%rsp),%r8
0x0000000000400ecd <+9>: lea 0x7(%rsp),%rcx
0x0000000000400ed2 <+14>: lea 0xc(%rsp),%rdx
=> 0x0000000000400ed7 <+19>: mov $0x402346,%esi
0x0000000000400edc <+24>: mov $0x0,%eax
0x0000000000400ee1 <+29>: callq 0x400bc0 <__isoc99_sscanf@plt>
0x0000000000400ee6 <+34>: cmp $0x2,%eax
0x0000000000400ee9 <+37>: jle 0x400f01 <phase_3+61>
0x0000000000400eeb <+39>: cmpl $0x7,0xc(%rsp)
0x0000000000400ef0 <+44>: ja 0x400ff9 <phase_3+309>
0x0000000000400ef6 <+50>: mov 0xc(%rsp),%eax
0x0000000000400efa <+54>: jmpq *0x402360(,%rax,8)
0x0000000000400f01 <+61>: callq 0x401418 <explode_bomb>
0x0000000000400f06 <+66>: jmp 0x400eeb <phase_3+39>
0x0000000000400f08 <+68>: mov $0x72,%eax
0x0000000000400f0d <+73>: cmpl $0x173,0x8(%rsp)
0x0000000000400f15 <+81>: je 0x401003 <phase_3+319>
0x0000000000400f1b <+87>: callq 0x401418 <explode_bomb>
0x0000000000400f20 <+92>: mov $0x72,%eax
0x0000000000400f25 <+97>: jmpq 0x401003 <phase_3+319>
0x0000000000400f2a <+102>: mov $0x6a,%eax
0x0000000000400f2f <+107>: cmpl $0x31a,0x8(%rsp)
0x0000000000400f37 <+115>: je 0x401003 <phase_3+319>
0x0000000000400f3d <+121>: callq 0x401418 <explode_bomb>
0x0000000000400f42 <+126>: mov $0x6a,%eax
0x0000000000400f47 <+131>: jmpq 0x401003 <phase_3+319>
0x0000000000400f4c <+136>: mov $0x6e,%eax
0x0000000000400f51 <+141>: cmpl $0x24f,0x8(%rsp)
0x0000000000400f59 <+149>: je 0x401003 <phase_3+319>
0x0000000000400f5f <+155>: callq 0x401418 <explode_bomb>
0x0000000000400f64 <+160>: mov $0x6e,%eax
0x0000000000400f69 <+165>: jmpq 0x401003 <phase_3+319>
0x0000000000400f6e <+170>: mov $0x79,%eax
0x0000000000400f73 <+175>: cmpl $0x13a,0x8(%rsp)
0x0000000000400f7b <+183>: je 0x401003 <phase_3+319>
0x0000000000400f81 <+189>: callq 0x401418 <explode_bomb>
0x0000000000400f86 <+194>: mov $0x79,%eax
---Type <return> to continue, or q <return> to quit---

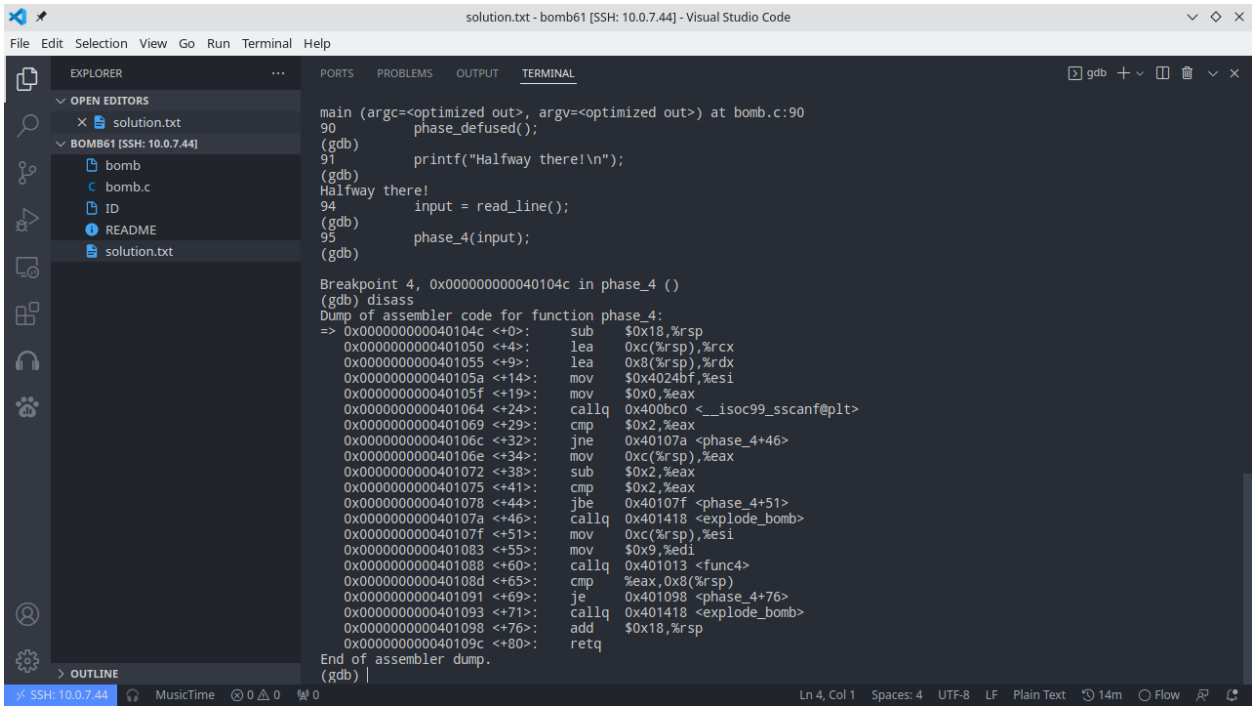

(edusc03-061) 10.0.7.44:2022 — Konsole
0x0000000000400f2f in phase_3 ()
1: x/i $pc
=> 0x400f2f <phase_3+107>:	cmpl $0x31a,0x8(%rsp)
(gdb)
0x0000000000400f37 in phase_3 ()
1: x/i $pc
=> 0x400f37 <phase_3+115>:	je 0x401003 <phase_3+319>
(gdb)
0x0000000000400f3d in phase_3 ()
1: x/i $pc
=> 0x400f3d <phase_3+121>:	callq 0x401418 <explode_bomb>
(gdb) info r
rax   0x6a   106
rbx   0x7fffffffe1f8   140737488347640
rcx   0x28   40
rdx   0x0   0
rsi   0x0   0
rdi   0x7fffffffdaf0   140737488345840
rbp   0x0   0x0
rsp   0x7fffffffe0f0   0x7fffffffe0f0
r8   0x2aaaab097060   46912502329440
r9   0x0   0
r10   0x0   0
r11   0x0   0
r12   0x400c50 4197456
r13   0x7fffffffe1f8   140737488347632
r14   0x0   0
r15   0x0   0
rip   0x400f3d 0x400f3d <phase_3+121>
eflags   0x283   [CF SF IF]
cs   0x33   51
ss   0x2b   43
ds   0x0   0
es   0x0   0
fs   0x0   0
gs   0x0   0
(gdb) output 0x7fffffffe0f0
140737488347376(gdb) output 0x7fffffffe0f8
140737488347384(gdb) output /x 0x7fffffffe0f8
0x7fffffffe0f8(gdb) output /x (0x7fffffffe0f8)

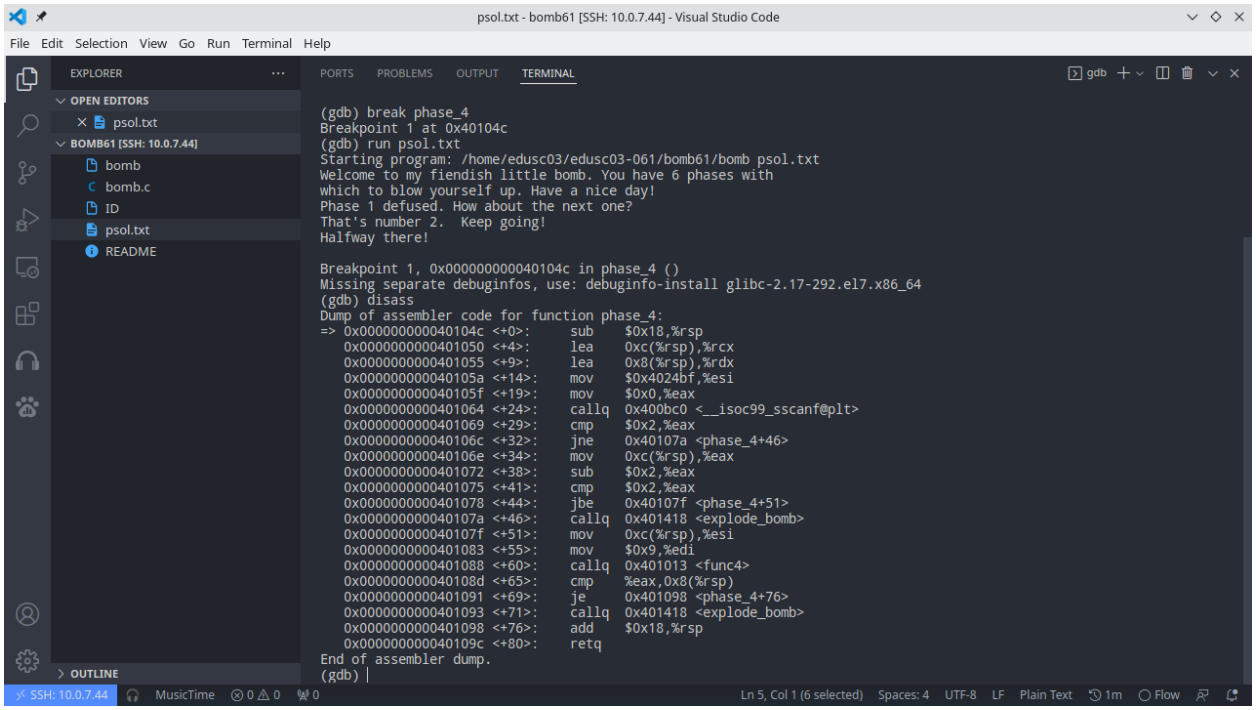

(edusc03-061) 10.0.7.44:2022 — Konsole
0x0000000000401007 in phase_3 ()
1: x/i $pc
=> 0x401007 <phase_3+323>:	je 0x40100e <phase_3+330>
(gdb)
0x0000000000401009 in phase_3 ()
1: x/i $pc
=> 0x401009 <phase_3+325>:	callq 0x401418 <explode_bomb>
(gdb) info r
rax   0x6a   106
rbx   0x7fffffffe1f8   140737488347640
rcx   0x28   40
rdx   0x0   0
rsi   0x0   0
rdi   0x7fffffffdaf0   140737488345840
rbp   0x0   0x0
rsp   0x7fffffffe0f0   0x7fffffffe0f0
r8   0x2aaaab097060   46912502329440
r9   0x0   0
r10   0x0   0
r11   0x0   0
r12   0x400c50 4197456
r13   0x7fffffffe1f8   140737488347632
r14   0x0   0
r15   0x0   0
rip   0x401009 0x401009 <phase_3+325>
eflags   0x293   [CF AF SF IF]
cs   0x33   51
ss   0x2b   43
ds   0x0   0
es   0x0   0
fs   0x0   0
gs   0x0   0
(gdb) output *(int *) ($rsp+7)
203313(gdb) output %al
A syntax error in expression, near `%al'.
(gdb) output $al
106(gdb) output ($rsp+7)
(void *) 0x7fffffffe0f7(gdb) output /x ($rsp+7)
0x7fffffffe0f7(gdb) output *(char *) ($rsp+7)
49 '1'(gdb) ^CQuit

Phase 4

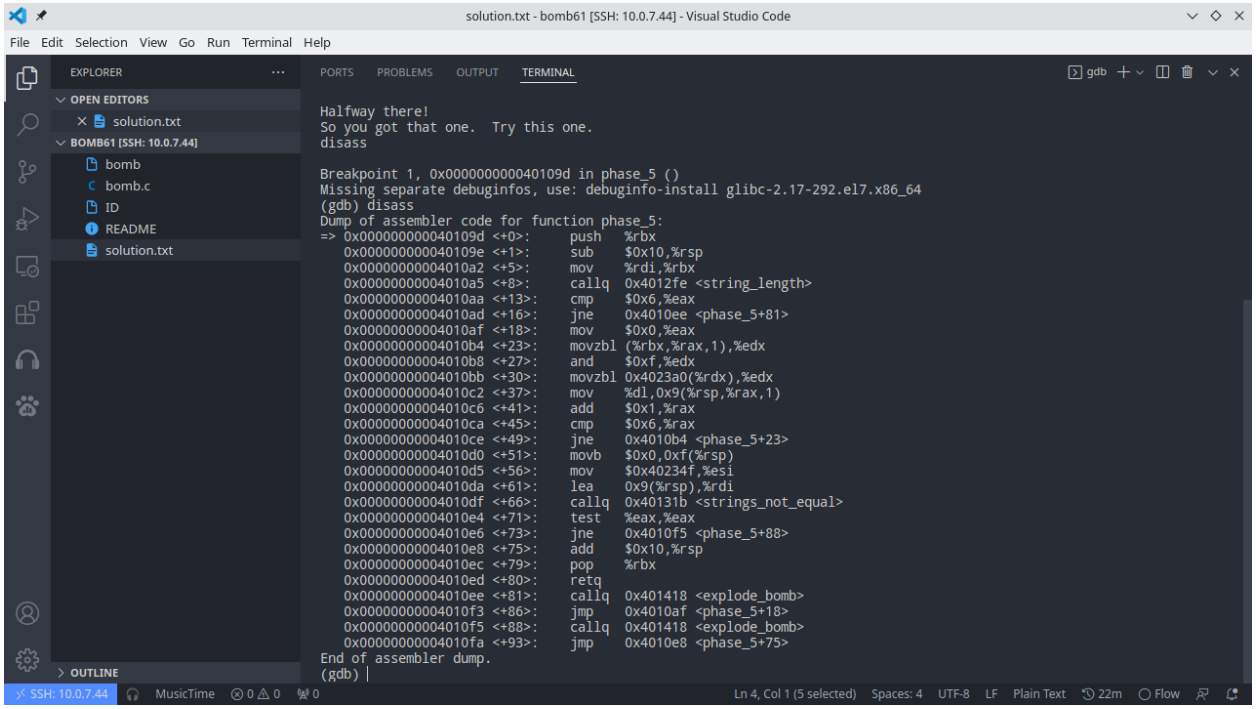

bomb61 [SSH: 10.0.7.44] - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
BOMB61 [SSH: 10.0.7.44]
bomb
bomb.c
ID
README
solution.txt
OUTLINE
PORTS PROBLEMS OUTPUT TERMINAL
gdb
[edusc03-061@cheetah bomb61]$ gdb bomb
GNU gdb (GDB) Red Hat Enterprise Linux 7.6.1-100.el7
Copyright (C) 2013 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-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/edusc03/edusc03-061/bomb61/bomb...done.
(gdb) break phase_1
Breakpoint 1 at 0x400e63
(gdb) break phase_2
Breakpoint 2 at 0x400e81
(gdb) break phase_3
Breakpoint 3 at 0x400ec4
(gdb) break phase_4
Breakpoint 4 at 0x40104c
(gdb) break phase_5
Breakpoint 5 at 0x40109d
(gdb) break phase_6
Breakpoint 6 at 0x4010fc
(gdb) run solution.txt
Starting program: /home/edusc03/edusc03-061/bomb61/bomb solution.txt
Welcome to my fiendish little bomb. You have 6 phases with
which to blow yourself up. Have a nice day!
Breakpoint 1, 0x0000000000400e63 in phase_1 ()
Missing separate debuginfos, use: debuginfo-install glibc-2.17-292.el7.x86_64
(gdb)
SSH: 10.0.7.44 MusicTime 0 0 0
14m Flow

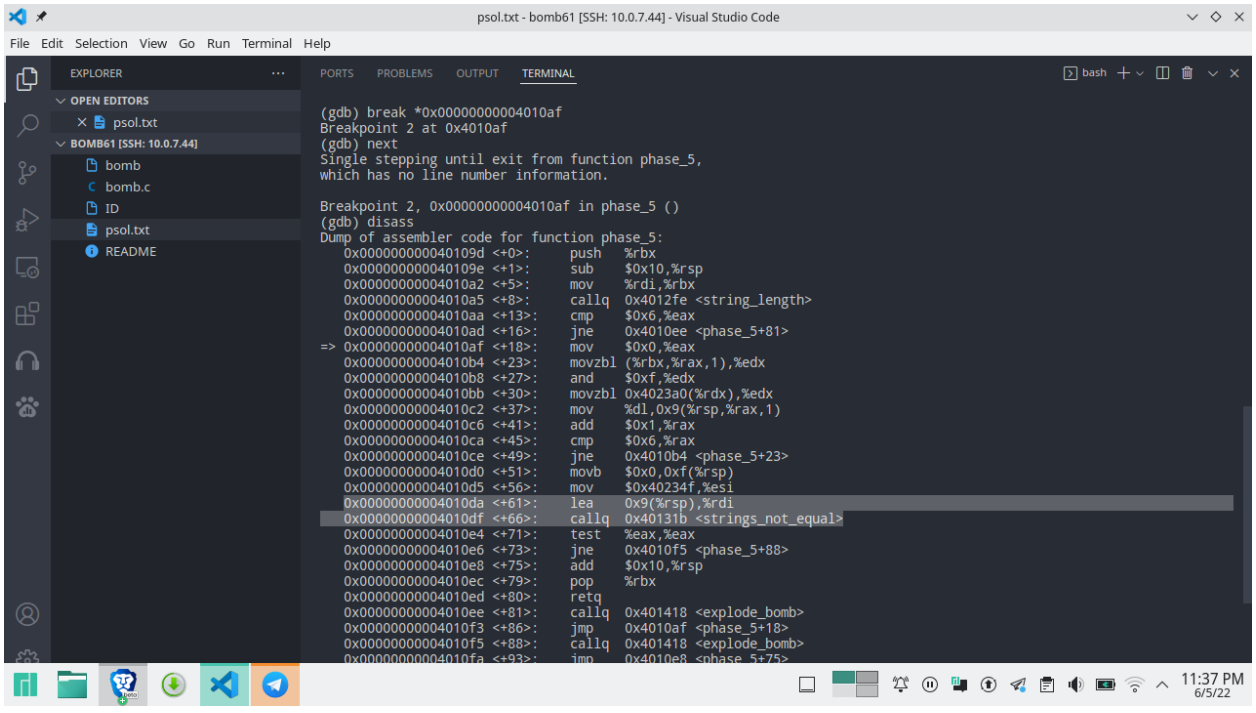

solution.txt - bomb61 [SSH: 10.0.7.44] - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
solution.txt
BOMB61 [SSH: 10.0.7.44]
bomb
bomb.c
ID
README
solution.txt
OUTLINE
PORTS PROBLEMS OUTPUT TERMINAL
gdb
which to blow yourself up. Have a nice day!
Breakpoint 1, 0x0000000000400e63 in phase_1 ()
Missing separate debuginfos, use: debuginfo-install glibc-2.17-292.el7.x86_64
(gdb) nexti
0x0000000000400e67 in phase_1 ()
(gdb) next
Single stepping until exit from function phase_1,
which has no line number information.
main (argc=<optimized out>, argv=<optimized out>) at bomb.c:75
75 phase_defused(); /* Drat! They figured it out!
(gdb) next
77 printf("Phase 1 defused. How about the next one?\n");
(gdb) next
Phase 1 defused. How about the next one?
81 input = read_line();
(gdb) nexti
82 phase_2(input);
(gdb) next
Breakpoint 2, 0x0000000000400e81 in phase_2 ()
(gdb) next
Single stepping until exit from function phase_2,
which has no line number information.
main (argc=<optimized out>, argv=<optimized out>) at bomb.c:83
83 phase_defused();
(gdb) next
84 printf("That's number 2. Keep going!\n");
(gdb) next
That's number 2. Keep going!
88 input = read_line();
(gdb) next
89 phase_3(input);
(gdb) next
Breakpoint 3, 0x0000000000400ec4 in phase_3 ()
(gdb)
SSH: 10.0.7.44 MusicTime 0 0 0
Ln 4, Col 1 Spaces: 4 UTF-8 LF Plain Text 14m Flow


solution.txt - bomb61 [SSH: 10.0.7.44] - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
solution.txt
BOMB61 [SSH: 10.0.7.44]
bomb
bomb.c
ID
README
solution.txt
OUTLINE
PORTS PROBLEMS OUTPUT TERMINAL
gdb
main (argc=<optimized out>, argv=<optimized out>) at bomb.c:90
90 phase_defused();
(gdb)
91 printf("Halfway there!\n");
(gdb)
Halfway there!
94 input = read_line();
(gdb)
95 phase_4(input);
(gdb)
Breakpoint 4, 0x000000000040104c in phase_4 ()
(gdb) disass
Dump of assembler code for function phase_4:
=> 0x000000000040104c <+0>: sub $0x18,%rsp
0x0000000000401050 <+4>: lea 0xc(%rsp),%rcx
0x0000000000401055 <+9>: lea 0x8(%rsp),%rdx
0x000000000040105a <+14>: mov $0x4024bf,%esi
0x000000000040105f <+19>: mov $0x0,%eax
0x0000000000401064 <+24>: callq 0x400bc0 <__isoc99_sscanf@plt>
0x0000000000401069 <+29>: cmp $0x2,%eax
0x000000000040106c <+32>: jne 0x40107a <phase_4+46>
0x000000000040106e <+34>: mov 0xc(%rsp),%eax
0x0000000000401072 <+38>: sub $0x2,%eax
0x0000000000401075 <+41>: cmp $0x2,%eax
0x0000000000401078 <+44>: jbe 0x40107f <phase_4+51>
0x000000000040107a <+46>: callq 0x401418 <explode_bomb>
0x000000000040107f <+51>: mov 0xc(%rsp),%esi
0x0000000000401083 <+55>: mov $0x9,%edi
0x0000000000401088 <+60>: callq 0x401013 <func4>
0x000000000040108d <+65>: cmp %eax,0x8(%rsp)
0x0000000000401091 <+69>: je 0x401098 <phase_4+76>
0x0000000000401093 <+71>: callq 0x401418 <explode_bomb>
0x0000000000401098 <+76>: add $0x18,%rsp
0x000000000040109c <+80>: retq
End of assembler dump.
(gdb)
SSH: 10.0.7.44 MusicTime 0 0 0
Ln 4, Col 1 Spaces: 4 UTF-8 LF Plain Text 14m Flow


psol.txt - bomb61 [SSH: 10.0.7.44] - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
psol.txt
BOMB61 [SSH: 10.0.7.44]
bomb
bomb.c
ID
psol.txt
README
OUTLINE
PORTS PROBLEMS OUTPUT TERMINAL
gdb
(gdb) break phase_4
Breakpoint 1 at 0x40104c
(gdb) run psol.txt
Starting program: /home/edusc03/edusc03-061/bomb61/bomb psol.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!
Breakpoint 1, 0x000000000040104c in phase_4 ()
Missing separate debuginfos, use: debuginfo-install glibc-2.17-292.el7.x86_64
(gdb) disass
Dump of assembler code for function phase_4:
=> 0x000000000040104c <+0>: sub $0x18,%rsp
0x0000000000401050 <+4>: lea 0xc(%rsp),%rcx
0x0000000000401055 <+9>: lea 0x8(%rsp),%rdx
0x000000000040105a <+14>: mov $0x4024bf,%esi
0x000000000040105f <+19>: mov $0x0,%eax
0x0000000000401064 <+24>: callq 0x400bc0 <__isoc99_sscanf@plt>
0x0000000000401069 <+29>: cmp $0x2,%eax
0x000000000040106c <+32>: jne 0x40107a <phase_4+46>
0x000000000040106e <+34>: mov 0xc(%rsp),%eax
0x0000000000401072 <+38>: sub $0x2,%eax
0x0000000000401075 <+41>: cmp $0x2,%eax
0x0000000000401078 <+44>: jbe 0x40107f <phase_4+51>
0x000000000040107a <+46>: callq 0x401418 <explode_bomb>
0x000000000040107f <+51>: mov 0xc(%rsp),%esi
0x0000000000401083 <+55>: mov $0x9,%edi
0x0000000000401088 <+60>: callq 0x401013 <func4>
0x000000000040108d <+65>: cmp %eax,0x8(%rsp)
0x0000000000401091 <+69>: je 0x401098 <phase_4+76>
0x0000000000401093 <+71>: callq 0x401418 <explode_bomb>
0x0000000000401098 <+76>: add $0x18,%rsp
0x000000000040109c <+80>: retq
End of assembler dump.
(gdb)
SSH: 10.0.7.44 MusicTime 0 0 0
Ln 5, Col 1 (6 selected) Spaces: 4 UTF-8 LF Plain Text 1m Flow

Phase 5


solution.txt - bomb61 [SSH: 10.0.7.44] - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
solution.txt
BOMB61 [SSH: 10.0.7.44]
bomb
bomb.c
ID
README
solution.txt
OUTLINE
PORTS PROBLEMS OUTPUT TERMINAL
gdb
Halfway there!
So you got that one. Try this one.
disass
Breakpoint 1, 0x000000000040109d in phase_5 ()
Missing separate debuginfos, use: debuginfo-install glibc-2.17-292.el7.x86_64
(gdb) disass
Dump of assembler code for function phase_5:
=> 0x000000000040109d <+0>: push %rbx
0x000000000040109e <+1>: sub $0x10,%rsp
0x00000000004010a2 <+5>: mov %rdi,%rbx
0x00000000004010a5 <+8>: callq 0x4012fe <string_length>
0x00000000004010aa <+13>: cmp $0x6,%eax
0x00000000004010ad <+16>: jne 0x4010ee <phase_5+81>
0x00000000004010af <+18>: mov $0x0,%eax
0x00000000004010b4 <+23>: movzbl (%rbx,%rax,1),%edx
0x00000000004010b8 <+27>: and $0xf,%edx
0x00000000004010bb <+30>: movzbl 0x4023a0(%rdx),%edx
0x00000000004010c2 <+37>: mov %dl,0x9(%rsp,%rax,1)
0x00000000004010c6 <+41>: add $0x1,%rax
0x00000000004010ca <+45>: cmp $0x6,%rax
0x00000000004010ce <+49>: jne 0x4010b4 <phase_5+23>
0x00000000004010d0 <+51>: movb $0x0,0xf(%rsp)
0x00000000004010d5 <+56>: mov $0x40234f,%esi
0x00000000004010da <+61>: lea 0x9(%rsp),%rdi
0x00000000004010df <+66>: callq 0x40131b <strings_not_equal>
0x00000000004010e4 <+71>: test %eax,%eax
0x00000000004010e6 <+73>: jne 0x4010f5 <phase_5+88>
0x00000000004010e8 <+75>: add $0x10,%rsp
0x00000000004010ec <+79>: pop %rbx
0x00000000004010ed <+80>: retq
0x00000000004010ee <+81>: callq 0x401418 <explode_bomb>
0x00000000004010f3 <+86>: jmp 0x4010af <phase_5+18>
0x00000000004010f5 <+88>: callq 0x401418 <explode_bomb>
0x00000000004010fa <+93>: jmp 0x4010e8 <phase_5+75>
End of assembler dump.
(gdb)
SSH: 10.0.7.44 MusicTime 0 0 0
Ln 4, Col 1 (5 selected) Spaces: 4 UTF-8 LF Plain Text 22m Flow


psol.txt - bomb61 [SSH: 10.0.7.44] - Visual Studio Code
File Edit Selection View Go Run Terminal Help
EXPLORER
OPEN EDITORS
psol.txt
BOMB61 [SSH: 10.0.7.44]
bomb
bomb.c
ID
psol.txt
README
PORTS PROBLEMS OUTPUT TERMINAL
bash
(gdb) break *0x00000000004010af
Breakpoint 2 at 0x4010af
(gdb) next
Single stepping until exit from function phase_5,
which has no line number information.
Breakpoint 2, 0x00000000004010af in phase_5 ()
(gdb) disass
Dump of assembler code for function phase_5:
0x000000000040109d <+0>: push %rbx
0x000000000040109e <+1>: sub $0x10,%rsp
0x00000000004010a2 <+5>: mov %rdi,%rbx
0x00000000004010a5 <+8>: callq 0x4012fe <string_length>
0x00000000004010aa <+13>: cmp $0x6,%eax
0x00000000004010ad <+16>: jne 0x4010ee <phase_5+81>
=> 0x00000000004010af <+18>: mov $0x0,%eax
0x00000000004010b4 <+23>: movzbl (%rbx,%rax,1),%edx
0x00000000004010b8 <+27>: and $0xf,%edx
0x00000000004010bb <+30>: movzbl 0x4023a0(%rdx),%edx
0x00000000004010c2 <+37>: mov %dl,0x9(%rsp,%rax,1)
0x00000000004010c6 <+41>: add $0x1,%rax
0x00000000004010ca <+45>: cmp $0x6,%rax
0x00000000004010ce <+49>: jne 0x4010b4 <phase_5+23>
0x00000000004010d0 <+51>: movb $0x0,0xf(%rsp)
0x00000000004010d5 <+56>: mov $0x40234f,%esi
0x00000000004010da <+61>: lea 0x9(%rsp),%rdi
0x00000000004010df <+66>: callq 0x40131b <strings_not_equal>
0x00000000004010e4 <+71>: test %eax,%eax
0x00000000004010e6 <+73>: jne 0x4010f5 <phase_5+88>
0x00000000004010e8 <+75>: add $0x10,%rsp
0x00000000004010ec <+79>: pop %rbx
0x00000000004010ed <+80>: retq
0x00000000004010ee <+81>: callq 0x401418 <explode_bomb>
0x00000000004010f3 <+86>: jmp 0x4010af <phase_5+18>
0x00000000004010f5 <+88>: callq 0x401418 <explode_bomb>
0x00000000004010fa <+93>: jmp 0x4010e8 <phase_5+75>
11:37 PM
6/5/22