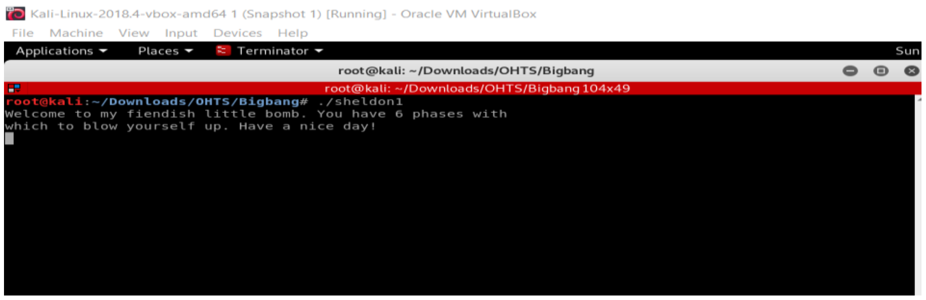
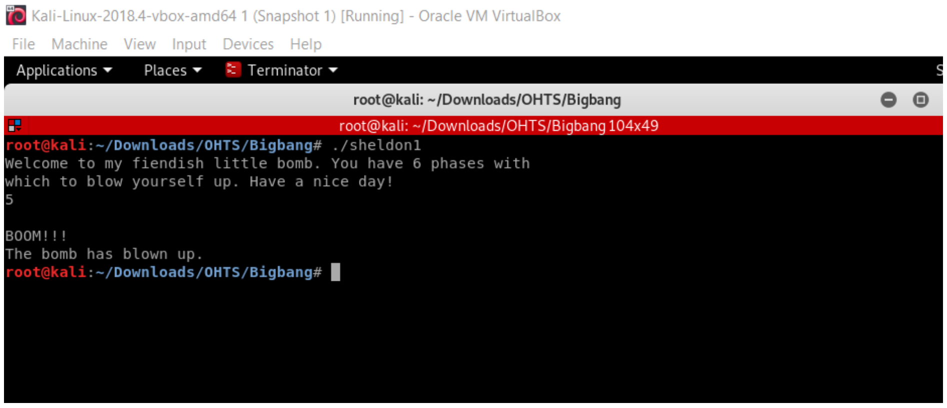
Bigbangtheory

First we have to execute the program and find the output.


Kali-Linux-2018.4-vbox-amd64 1 (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminator
root@kali: ~/Downloads/OHTS/Bigbang
root@kali: ~/Downloads/OHTS/Bigbang 104x49

```
root@kali:~/Downloads/OHTS/Bigbang# ./sheldon1
Welcome to my fiendish little bomb. You have 6 phases with
which to blow yourself up. Have a nice day!
```

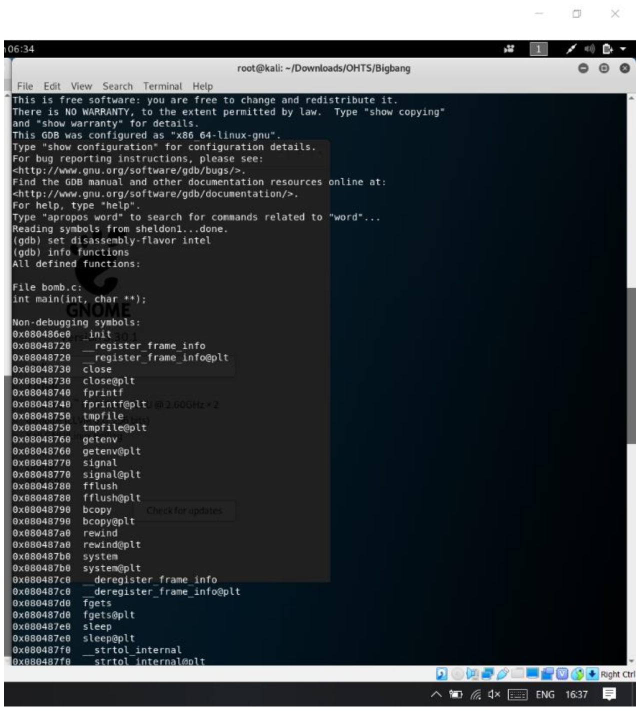
When we enter something, the bomb will blow up.


Kali-Linux-2018.4-vbox-amd64 1 (Snapshot 1) [Running] - Oracle VM VirtualBox
File Machine View Input Devices Help
Applications Places Terminator
root@kali: ~/Downloads/OHTS/Bigbang
root@kali: ~/Downloads/OHTS/Bigbang 104x49

```
root@kali:~/Downloads/OHTS/Bigbang# ./sheldon1
Welcome to my fiendish little bomb. You have 6 phases with
which to blow yourself up. Have a nice day!
5

BOOM!!!
The bomb has blown up.
root@kali:~/Downloads/OHTS/Bigbang# 
```

Then we have to analyze the assembly code of the code.

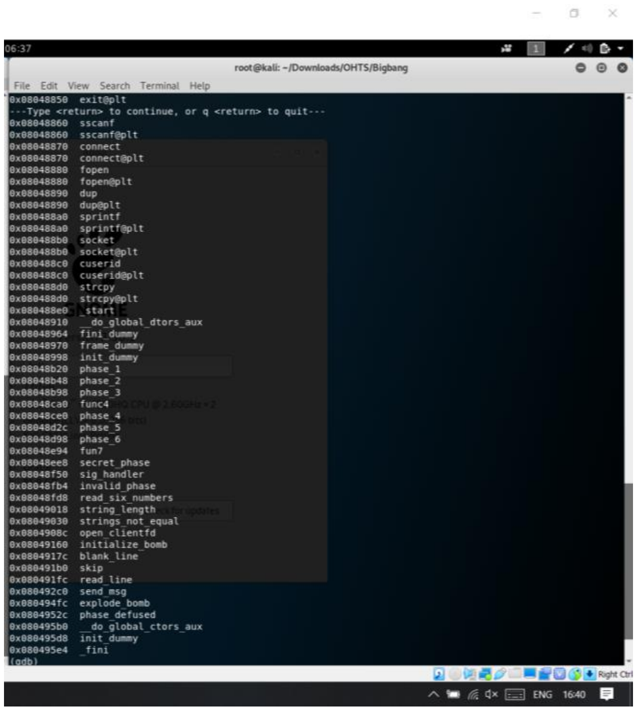


```
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<http://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 sheldon1...done.
(gdb) set disassembly-flavor intel
(gdb) info functions
All defined functions:

File bomb.c:
int main(int, char **);

Non-debugging symbols:
0x080486e0  _init
0x08048720  __register_frame_info
0x08048720  __register_frame_info@plt
0x08048730  close
0x08048730  close@plt
0x08048740  fprintf
0x08048740  fprintf@plt
0x08048750  tmpfile
0x08048750  tmpfile@plt
0x08048760  getenv
0x08048760  getenv@plt
0x08048770  signal
0x08048770  signal@plt
0x08048780  fflush
0x08048780  fflush@plt
0x08048790  bcopy
0x08048790  bcopy@plt
0x080487a0  rewind
0x080487a0  rewind@plt
0x080487b0  system
0x080487b0  system@plt
0x080487c0  __deregister_frame_info
0x080487c0  __deregister_frame_info@plt
0x080487d0  fgets
0x080487d0  fgets@plt
0x080487e0  sleep
0x080487e0  sleep@plt
0x080487f0  __strtol_internal
0x080487f0  __strtol_internal@plt
```

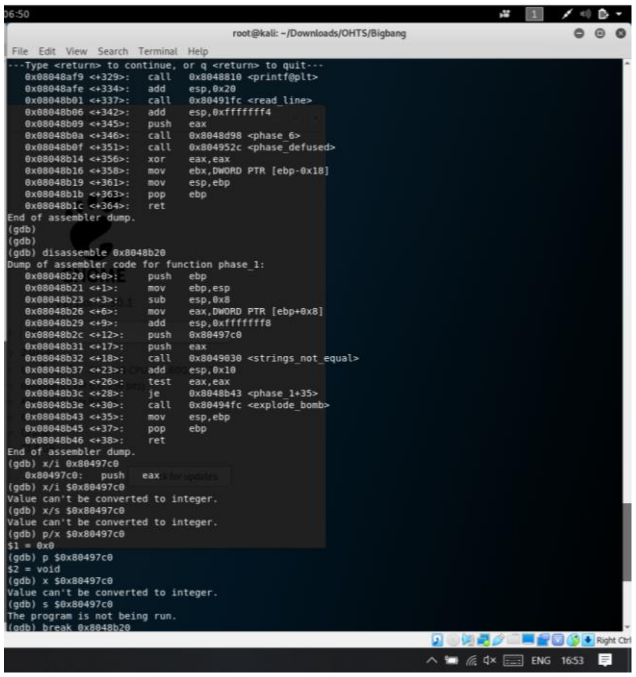
when analyzing the functions of the code we can see there is functions called phase1



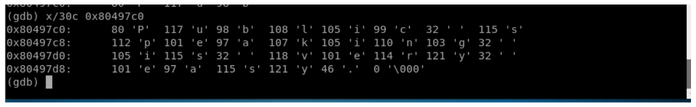
```
0x08048850  exit@plt
---Type <return> to continue, or q <return> to quit---
0x08048860  sscanf
0x08048860  sscanf@plt
0x08048870  connect
0x08048870  connect@plt
0x08048880  fopen
0x08048880  fopen@plt
0x08048890  dup
0x08048890  dup@plt
0x080488a0  sprintf
0x080488a0  sprintf@plt
0x080488b0  socket
0x080488b0  socket@plt
0x080488c0  cuserid
0x080488c0  cuserid@plt
0x080488d0  strcpy
0x080488d0  strcpy@plt
0x080488e0  _start
0x08048910  __do_global_dtors_aux
0x08048964  fini_dummy
0x08048970  frame_dummy
0x08048998  init_dummy
0x08048b20  phase_1
0x08048b48  phase_2
0x08048b98  phase_3
0x08048ca0  func4
0x08048ce0  phase_4
0x08048d2c  phase_5
0x08048d98  phase_6
0x08048e94  fun7
0x08048ee8  secret_phase
0x08048f50  sig_handler
0x08048fb4  invalid_phase
0x08048fd8  read_six_numbers
0x08049018  string_length
0x08049030  strings_not_equal
0x0804908c  open_clientfd
0x08049160  initialize_bomb
0x0804917c  blank_line
0x080491b0  skip
0x080491fc  read_line
0x080492c0  send_msg
0x080494fc  explode_bomb
0x0804952c  phase_defused
0x080495b0  __do_global_ctors_aux
0x080495d8  init_dummy
0x080495e4  _fini
(gdb)
```

Then we have to analyze the phase 1 function.

We have to disassemble from the phase1 function.



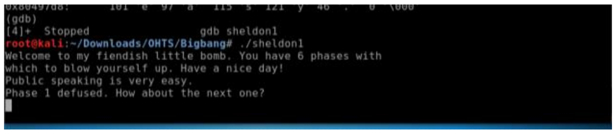
```
---Type <return> to continue, or q <return> to quit---
   0x08048af9 <+329>:   call   0x8048810 <printf@plt>
   0x08048afe <+334>:   add    esp,0x20
   0x08048b01 <+337>:   call   0x80491fc <read_line>
   0x08048b06 <+342>:   add    esp,0xfffffff4
   0x08048b09 <+345>:   push   eax
   0x08048b0a <+346>:   call   0x8048d98 <phase_6>
   0x08048b0f <+351>:   call   0x804952c <phase_defused>
   0x08048b14 <+356>:   xor    eax,eax
   0x08048b16 <+358>:   mov    ebx,DWORD PTR [ebp-0x18]
   0x08048b19 <+361>:   mov    esp,ebp
   0x08048b1b <+363>:   pop    ebp
   0x08048b1c <+364>:   ret
End of assembler dump.
(gdb)
(gdb)
(gdb) disassemble 0x8048b20
Dump of assembler code for function phase_1:
   0x08048b20 <+0>:     push   ebp
   0x08048b21 <+1>:     mov    ebp,esp
   0x08048b23 <+3>:     sub    esp,0x8
   0x08048b26 <+6>:     mov    eax,DWORD PTR [ebp+0x8]
   0x08048b29 <+9>:     add    esp,0xfffffff8
   0x08048b2c <+12>:    push   0x80497c0
   0x08048b31 <+17>:    push   eax
   0x08048b32 <+18>:    call   0x8049030 <strings_not_equal>
   0x08048b37 <+23>:    add    esp,0x10
   0x08048b3a <+26>:    test   eax,eax
   0x08048b3c <+28>:    je     0x8048b43 <phase_1+35>
   0x08048b3e <+30>:    call   0x80494fc <explode_bomb>
   0x08048b43 <+35>:    mov    esp,ebp
   0x08048b45 <+37>:    pop    ebp
   0x08048b46 <+38>:    ret
End of assembler dump.
(gdb) x/i 0x80497c0
   0x80497c0:   push   eax
(gdb) x/i $0x80497c0
Value can't be converted to integer.
(gdb) x/s $0x80497c0
Value can't be converted to integer.
(gdb) p/x $0x80497c0
$1 = 0x0
(gdb) p $0x80497c0
$2 = void
(gdb) x $0x80497c0
Value can't be converted to integer.
(gdb) s $0x80497c0
The program is not being run.
(gdb) break 0x8048b20
```



```
(gdb) x/30c 0x80497c0
0x80497c0:      80 'P'  117 'u' 98 'b'  108 'l' 105 'i' 99 'c'  32 ' '  115 's'
0x80497c8:      112 'p' 101 'e' 97 'a'  107 'k' 105 'i' 110 'n' 103 'g' 32 ' '
0x80497d0:      105 'i' 115 's' 32 ' '  118 'v' 101 'e' 114 'r' 121 'y' 32 ' '
0x80497d8:      101 'e' 97 'a'  115 's' 121 'y' 46 '.'  0 '\000'
(gdb) 
```

Now we can see the set of words are coming out from the address 0x80497c0

Let’s try those words



```
(gdb)
[4]+  Stopped                 gdb sheldon1
root@kali:~/Downloads/OHTS/Bigbang# ./sheldon1
Welcome to my fiendish little bomb. You have 6 phases with
which to blow yourself up. Have a nice day!
Public speaking is very easy.
Phase 1 defused. How about the next one?
```

Phase 1 is defused.