C PROGRAMING TRAINING COURSE

HUMAX VIETNAM R&D

Tutor: Vu Quang Trong | Henry

version: 1.0 R, updated on: 2017 Feb 8th

Day 4: Function pointer, Overflow Buffer, Debugging

Test Date: **20-Feb-2017**

Total Tests: **05 questions**

Maximum Score: **100 points**

Total Time: **120 mins**

Modify the code to make it work properly as required. Code is written in C language, C99 standard, and compiled under Linux.

Test 1

Score: 20

Use `fooptr` to print out `11` on the screen.

```
#include <stdio.h>
void f(int (*x)(int));
int myfoo(int);
int (*fooptr)(int);

int ((*foo(int)))(int);

int ((*foo(int i)))(int)
{
    return myfoo;
}

int myfoo(int i)
{
    printf("%d\n", i + 1);
}

int main()
{
    int i = 10;
    // your code goes here
}
```

Test 2

Score: 20

Not using `showMe` variable, print out `HUMAX`

```c
#include <stdio.h>

void main()
{
    char showMe[] = "HUMAX";
    int foo = 0;
    int bar = 0;
    // Your code goes here
}
```

Test 3

Score: 40

A list of callback fucntions is based on a node declared as below:

```c
struct _command {
    int *(callback)(int); // A command will excecute a callback function
    struct _command *next;
};
```

Now create a list of commands.

```c
struct _command * initCommandList();
void insertCommand(command * list, int *(callback)(int));  // if command is already in the list, skip inserting, if r
```

Assume that we want to do 4 operations:

```c
int add5(int a){return a + 5};
int sub4(int a){return a - 4};
int mul3(int a){return a * 3};
int div2(int a){return a / 2};
```

To calculating an expression like y = ((((x + 5) - 4) * 3) / 2), we can using a list of above functions in specific order.

Make a command list `struct _command *orderOfFunctions` to hold the functions. Write a fucntion `int calculate(int a, struct _command * orderOfFunctions)` to make the right calculation.

Test 4

Score: 10

It's easy to modify data on stack. `scanf()` is not safe, it can do it. `scanf()` does not check how much memory is reserved for writing, so it can write more than it's expected. What if the input string is longer than buffer?

```c
/*
The following compiler flags might be set to exploit this bug:
    -m32                    compile for 32bit
    -fno-stack-protector    disable ProPolice
    -Wl,-z,norelro          disable relro
*/

#include <stdio.h>
#include <stdlib.h>

int main(){
    long val=0x41414141;
    char buf[4];

    printf("Correct val's value from 0x41414141 -> 0xdeadbeef!\n");
    printf("Here is your chance: ");
```

```
    scanf("%8s",&buf);

    printf("buf: %s\n",buf);
    printf("val: 0x%08x\n",val);

    if(val==0xdeadbeef)             /*** <== how to write data to val ??? ***/
            printf("OK! YOU WIN!!!!\n");
    else {
            printf("WAY OFF!!!!\n");
            exit(1);
    }

    return 0;
}
```

Test 5

Score: 10

Write missing piece of code and run it. You mission is to change the function pointer, from `NULL` to `puts`. The address of function `puts` is given by entering as argument when you call the app. If the function `puts` is already in system memory, how can we find it?

Tools

GDB

Valgrind

Ltrace

GDB CheatSheet

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(int argc, char *argv[]){
        int  (*fp)(char *);

        if(argc!=2){ printf("insert 0xaddress\n"); exit(-1); }

        // read input address
        // assign it to fp

        // ok, check the pointer
        if((int(*)(char *))&puts == fp)
        {
            fp("Well done! You had 'puts'. Excecuted it for you :)");
        }
        else
        {
            exit(-1);
        }

        exit(1);
}
```

End of the test.

(c) Humax Digital

this document is written in GitHub Markdown format