Assignment 3

Secure Programming, 2016

Elias Athanasopoulos - Sanjay Rawat
i.a.athanasopoulos@vu.nl - s.rawat@vu.nl

Question 1:

In the class, we discussed the following C code that has a buffer overflow bug. How will you patch the code to remove the bug?

```
int get_cookie()
{
    return 0x41424343;
}
int main()
{
    int guess;
    char name[20];
    guess=get_cookie();//ABCD
    printf("Enter your name..\n");
    gets(name);
    if(guess == 0x41424344)
        printf("You win %s\n",name);
    else printf("Better luck next time %s :(\n",name);
    return 0;
}
```

Question 2:

In general, which of the CIA properties may be violated by a buffer overflow bug?

(i) Confidentiality (ii) Integrity (iii) Availability (iv) C & I only (v) All (vi) None

Question 3:

Consider the following code:

```
void main(){
      unsigned short a, b,c
      scanf("%hu %hu", &a, &b);
      c=a+b;
      malloc(c);
      …..}
```

What is the problem in this code and how will you fix that?

Question 4:

Consider the C code given below ("offby1.c"). The code is supposed to take a name and then a secret code as an input from the user and based on the correctness of the supplied secret code, it allows/disallows a user. However, it has got few bugs, which allows an attacker to either crash the code OR enter the *system* by getting the right secret code!

A. Can you identify how these two scenarios can be achieved by the attacker? You need to supply the corresponding inputs that help attacker to

(i) crash the code (by overflowing RET address) (ii) know the secret code (a specific input will make secret getting printed!!).

B. How will you patch the code to remove these bugs (you are not allowed to change the buffer size of any variables though)? As a hint, look for lines that are tagged with "//BUG"! As an answer to this, provide the patched code.

```
----------------- The offby1.c code -------------
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

void get_name(char *name, char *pr)
{
    char local[20];
    printf("%s:",pr);
    gets(local);// BUG
    strncat(name,local,20);
}


int foo ()
{
    char name[28]="Hello..";
    char secret[12]="TOP SECRET";
    char buf[24];
    char n1[]="Enter your name";
    char n2[]="Enter Secret code";
    get_name(name,n1);
    memset(buf, 0, sizeof(buf));
    // Lets ONLY use strncpy for better control!!!
    strncpy(buf, name, sizeof(buf));//BUG
    printf("%s\n", buf);
    memset(name,0,sizeof(name));
    get_name(name, n2);
    if (strncmp(secret,name,10)==0)
        printf("Welcome and %s\n",buf);
    else {printf("Wrong code, better try again..\n");}


    return 0;


}

int main(int argc, char **argv)
{
    foo();
    printf("Bye\n");
    return 0;
}
```