CSE4501 – Vulnerability Research: Lab 3

Eric Pereira

Septemer 24th, 2019

Contents

Use the necessary tools to perform analysis of the compiled source code generated. Since we are working in 32-bit use “-m32” for your gcc compilation options. For compilation I would suggest using compiler option “`-fno-stack-protector`” and “`-z execstack`”. Please submit write-ups of your analysis. Your points will be based on completeness of analysis, your understanding of the problems, and if you were able to achieve the goals. Each problem is worth 5 pts for a total of 35pts.

Problem 1

Compile the below program; open in disassembler and compare source code with assembled instructions. Examine the program; What address did the control flow deviate to?

```
#include<string.h>

void function(char *str) {
  char buffer[16];

  strcpy(buffer,str);
}

void main() {
  char large_string[256];
  int i;

  for(i = 0; i < 255; i++)
    large_string[i] = "A";

  function(large_string);
}
```

Solution:

Problem 2

Compile the below program; open in disassembler and compare source code with assembled instructions. Examine the program; If you add a local variable in function can you still cause control flow deviation?

```
#include<stdio.h>

void function(int a, int b, int c) {
  char buffer1[5];
  char buffer2[10];
  int *ret;

  ret = buffer1 + 13;
  (*ret) += 10;
```

```c
}

void main() {
  int x;

  x = 0;
  function(1,2,3);
  x = 1;
  printf("%d\n",x);
}
```

Solution:

Problem 3

Compile the below program and run it. Who are you in this shell?

```c
#include <stdio.h>
#include <unistd.h>

void main() {
  char *name[2];

  name[0] = "/bin/sh";
  name[1] = NULL;
  execve(name[0], name, NULL);
}
```

Solution:
I am the user **re** in the shell.

Problem 4

Compile the below program; open in debugger. Examine the different shell-codes; What are each of them doing?

```c
/*char shellcode[] =
  "\xeb\x1f\x5e\x89\x76\x08\x31\xc0\x88\x46\x07\x89\x46\x0c"
  "\xb0\x0b\x89\xf3\x8d\x4e\x08\x8d\x56\x0c\xcd\x80\x31\xdb"
  "\x89\xd8\x40\xcd\x80\xe8\xdc\xff\xff\xff/bin/sh";
*/
/*char shellcode[] =
  "\x31\xc0\x31\xdb\x31\xc9\x99\xb0\xa4\xcd\x80\x6a\x0b\x58"
  "\x51\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89\xe3\x51"
  "\x89\xe2\x53\x89\xe1\xcd\x80";
*/
char shellcode[] =

  "\x31\xc0\x50\x68\x2f\x2f\x73\x68\x68\x2f\x62\x69\x6e\x89
  \xe3\x50\x89\xe2\x53\x89\xe1\xb0\x0b\xcd\x80";
```

```c

void main() {
  int *ret;

  ret = (int *)&ret + 2;
  (*ret) = (int)shellcode;

}
```

Solution:

Problem 5

Compile the below program; Cause a control flow deviation to “win”

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

void win()
{
  printf("code flow successfully changed\n");
}

int main(int argc, char **argv)
{
  volatile int (*fp)();
  char buffer[64];

  fp = 0;

  gets(buffer);

  if(fp) {
    printf("calling function pointer, jumping to 0x%08x\n", fp);
    fp();
  }
}
```

Solution:

Problem 6

Compile the below program; Cause a control flow deviation to “secretFunction”

```c
#include <stdio.h>

```

```c
void secretFunction()
{
  printf("Congratulations!\n");
  printf("You have entered in the secret function!\n");
}

void echo()
{
  char buffer[20];

  printf("Enter some text:\n");
  scanf("%s", buffer);
  printf("You entered: %s\n", buffer);
}

int main()
{
  echo();
  return 0;
}
```

Solution:

Problem 7

Compile the below program; Cause a control flow deviation to a shell?.

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

void func(char *name)
{
  char buf[100];
  strcpy(buf, name);
  printf("Welcome %s\n", buf);
}
int main(int argc, char *argv[])
{
  func(argv[1]);
  return 0;
}
```

Solution: