CSE4501 – Vulnerability Research: Lab 6

Eric Pereira

Septemer 24th, 2019

Contents

Problem 1
Problem 2
Problem 3
Problem 4
Problem 5

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 25pts.

Problem 1

Cause the CFD.

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

int target;

void vuln(char *string){
  printf(string);

  if(target){
    printf("you have modified the target :)\n");
  }
}

int main(int argc, char **argv){
  vuln(argv[1]);
}
```

Solution:

Problem 2

Set target to the specific value to win.

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

int target;

void vuln() {
  char buffer[512];

  fgets(buffer, sizeof(buffer), stdin);
  printf(buffer);

```

```
14   if (target == 64){
15     printf("You have modified the target :)\n");
16   } else {
17     printf("target is %d :(\n", %target);
18   }
19
20 }
21
22 int main(){
23   vuln();
24 }
```

Solution:

Problem 3

Set target to specific value to win.

```
1  #include <stdlib.h>
2  #include <unistd.h>
3  #include<stdio.h>
4  #include<string.h>
5
6  int target;
7
8  void printbuffer(char *string){
9    printf(string);
10 }
11
12 void vuln(){
13   char buffer[512];
14
15   fgets(buffer,sizeof(buffer), stdin);
16
17   printbuffer(buffer);
18
19   if(target==0x01025544){
20     printf("you have modified the target :)\n");
21   } else {
22     printf("target is %08x :(\n", target);
23   }
24 }
25
26 int main(int argc, char **argv){
27   vuln();
28 }
29
30 void main() {
31   char *name[2];
32
33   name[0] = "/bin/sh";
34   name[1] = NULL;
35   execve(name[0], name, NULL);
36 }
```

Solution:

Problem 4

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

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

int target;

void win(){
  printf("Code execution redirected! you win\n");
}

void vuln(){
  char buffer[512];

  fgets(buffer, sizeof(buffer), stdin);

  printf(buffer);

  exit(1);
}

int main(int argc, char **argv){
  vuln();
}
```

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>

int target;

void GetAddr(char* Name){Z
  printf("%s is at %p\n", Name, getenv(Name));
}

```

```
12 void vuln()
13 {
14   char buffer[512];
15
16   fgets(buffer, sizeof(buffer), stdin);
17
18   printf(buffer);
19
20   exit(1);
21 }
22
23 int main(int argc, char** argv){
24   GetAddr(argv[1]);
25   vuln();
26 }
```

Solution: