Understanding a Function Pointer Overwrite

Dr. Jared DeMott
CHIEF HACKING OFFICER

@jareddemott www.vdalabs.com

Overview

Discuss Bug Classes

- Basic Stack Overflow

Exploit Function Pointer

- Demo

Some Bug Classes

Traditional Stack Overflow

Function pointer Overwrite

Traditional Heap Overflow

Off-by-Ones

Format String Exceptions

Uninitialized Variables

Integer Errors

UaF

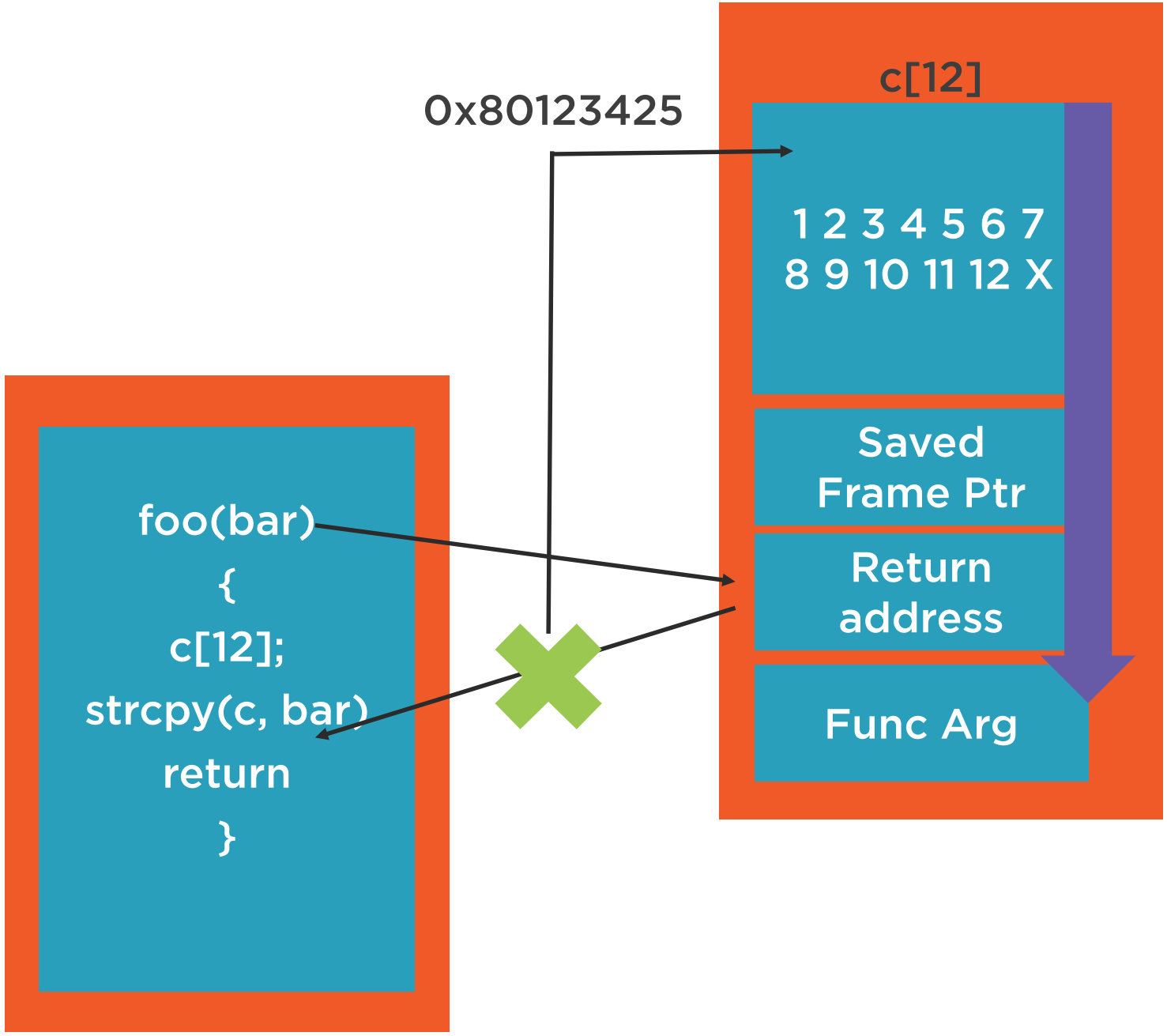
Double Fetch

Type Confusion

Traditional Stack Overflow

Traditional stack overflow

```
void foo (char *bar)
{
   char  c[12];

   strcpy(c, bar);  // no bounds checking...
}

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


0x80123425
c[12]
1 2 3 4 5 6 7
8 9 10 11 12 X
Saved
Frame Ptr
Return
address
Func Arg
foo(bar)
{
c[12];
strcpy(c, bar)
return
}

Function Pointers

C

- Index into an array of functions

C++ vtables

- Ptr is dereferenced to call another function

If Overwritten

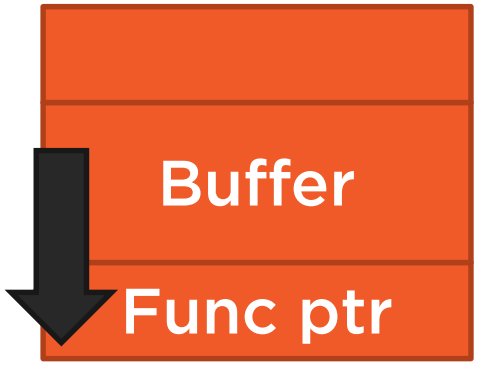
- Call to arbitrary locations
- Call any function in virtual address space
  - E.g. call the *loggedin()* routine to bypass the *login()*

```
int game(int);
int jackpot();
void foo(char *);


int main(int argc, char *argv[]) {
   if(argc < 2) {
      printf("Usage: %s <a number 1 - 32000>\n", argv[0]);
      printf("use %s help or %s -h for more help.\n", argv[0], argv[0]);
      exit(0);
   }
  foo(argv[1]);
  return 0;
}
```

```
void foo(char * input) {


 int (*function_ptr) (int user_pick);

 char buffer[20];

 srand(time(NULL));

 function_ptr = game;

 strcpy(buffer, input);
```

```
if((!strcmp(buffer, "help")) ||
(!strcmp(buffer, "-h")))

{

 printf("Help Text:\n\n");

 printf("This is a game of chance.\n");

 printf("To play, simply guess a number
1 through 32000\n");

 printf("If you guess the number I am
thinking of you win.\n");

}

 else

   function_ptr(atoi(buffer));

}
```

Simple Stack Frame



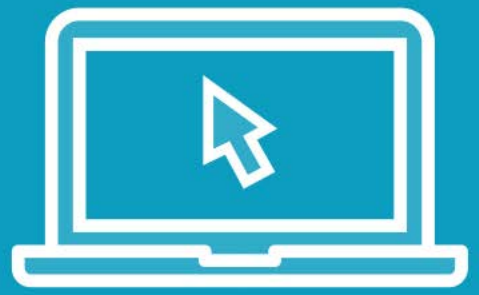
```
int game(int user_pick)
{
   int rand_pick;

if((user_pick < 1) || (user_pick > 32000)) {
 printf("You must pick a value from 1 - 32000\n");
 printf("Use help or -h for help\n");
 return 0;
}
```

```
printf("Playing the game of chance..\n");
rand_pick = (rand()% 32000) + 1;
printf("You picked: %d\n", user_pick);
printf("Random Value: %d\n", rand_pick);

 if(user_pick == rand_pick)
   jackpot();
 else
   printf("Sorry, you didn't win this time..\n");
}
```

IDA or Debugger to Find Address

```
int jackpot()
{
   printf("You just won!\n");
   printf("Congratulations!\n");

   return 0;
}
```

Demo



Function Pointer Exploit

- Look at Bug in IDA
- Automate the exploit with python
- Exploit with the address of Jackpot

Lab 2

- Review the same techniques from the demonstration

Summary

Hijacked Control Flow

- Using Corrupted Function Pointer
- Next:
  - Add in Shellcode for first arbitrary code execution attack