Samuel Lopez

Oct 9th, 2022

Application Security Homework 1

Bugs

Crash 1

The first bug I found was with the Num_Bytes variable in the GiftCardReader program. This variable is an integer type, and integers have a value limit of **2147483647**. When Num_Bytes is made larger than this it becomes negative so when Num_Bytes is passed into malloc() as this negative value, it will return a NULL pointer and will cause the following line to crash. I fixed this by checking to see if Num_Bytes is negative prior to passing it into the malloc(), and returning error text to the user if so.

Crash 2

The second bug I found was with the command line input when trying to run the gift card reader. This bug causes the program to crash if no gift card is passed into the program, this happened due to a lack of argument checking in the main function. I fixed this by checking if the gift card argument was NULL in the main function and returning error text to the user if so.

Hang

The hang case I found was in the animate function under case 0x09. When we pass in [0x09, (130), 0xff], case 0x09 will take the integer 130 and cast it to a char which will cause it to overflow. Since the pc variable is being increased by this overflow, it will be increased by -3. Then the case will break and the pc variable will be increased by 3 which will cause case 0x09 to be rerun and this will happen on a loop. The fix for this is to only run case 0x09 if arg1 is less than 15 as this will ensure that the pc variable won't go outside the bounds.

Testing

Cov1

For the first coverage gift card I ran the animate function with case 0x00, and case 0x09. I also ran the gift card info with type of record equal to 1 and 3 which yielded about 60% of line coverage.

Cov2

For the second coverage gift card, I ran the animate function with the remainder of the cases {0x01, 0x02, 0x03, 0x04, 0x06, 0x07, 0x08}, I also ran the reader with the card type record 2 which further increased the coverage to about 74%.

Fuzz1

The first crash case I found was in the animate function in case 0x03, the reason why I believed this case was causing a crash is because the arg1 variable is being made larger than the mptr can handle so when the value of arg1 is casted to a char and added to the mptr it's out of bounds and crashes. I fixed this by checking to see if the msg variable + 31 would be less than the mptr, as this would made sure it doesn't go out of bounds after the addition.

Fuzz2

The second crash case I found was with the animate function case 0x01. In this scenario I believe that arg1 variable was made out of the bounds of the regs[16] array, so when we do regs[arg1], we're indexing out of bounds of the array and this causes a crash. I fixed his by bounding the arg1 variable to be between 0 and 15, inclusive.