Sander Brattinga
Nico Glas
Gerjo Meier

Software Security

Assignment 1: Buffer Overflow

A buffer overflow happens when there is not enough memory assigned to a variable and everything gets written over other variables that are in the memory.

Example code:

```
void unsafe()
{
       srand(time(NULL));
       number = rand() % 10;
       char guessedRight = '0';
       char buffer[2];

       while(guessedRight == '0') {
              gets(buffer);

              hint(atoi(buffer));
              if(atoi(buffer) == number)
                     guessedRight = '1';

              printf("Status: %c \n", guessedRight);
       }
}
```

In this example you can see that the gets() function is used. The gets() function does not check the size of the input and this will make it possible to overwrite the memory.

```
void safe()
{
       srand(time(NULL));
       number = rand() % 10;
       char guessedRight = '0';
       char buffer[2];

       while(guessedRight == '0') {
              // NOTE: gets_s is only since recent in the C standard. (ISO/IEC
9899:2011)
              gets_s(buffer, 2);

              hint(atoi(buffer));

              if(atoi(buffer) == number)
                     guessedRight = '1';

              printf("Status: %c\n", guessedRight);
       }
}
```

In this code example you can see that gets_s() does check the size of the string to make sure it’s not possible to overflow the buffer.