CSEC-S

C programming Series

Starter Code

Our task is to build a simple C program from ground up. This is necessary to get you hands on experience with syntax (and logic) for new programmers

Skipping the obligatory ‘Hello World’ program, we will write a cooler program. We will be building an authentication program using a ‘naïve’ approach. In later meetings, we will discuss the naïve-ness of this approach.

Without further ado, lets get to business. Copy the right slide into a file and save as ‘auth.c’

```
/**
 * The program holds a copy of the correct password in the source. User supplies
 * password on request. Password supplied is checked against password hard-
 * coded in source. On match we print ACCESS GRANTED, else we print ACCESS
 * DENIED.
 * The correct password is s3cr3t
 */

// insert necessary include lines


// insert function prototypes (if any)


// implement the main function


// implement any prototypes
```

Live Programming Notes

- Standard format of a C program
  - Include statements: link relevant libraries. Eg
    - #include <stdio.h> // standard input/output; printf e.t.c
  - Define statements and global variables
  - Function prototypes or declarations
    - Functions like variables must be declared before use
    - Can declare a function by giving it's full implementation or declaration
      - int add(int, int); // implementation
      - int void add(int op1, int op2) { // implementation
        return op1 + op2;
        }
  - main() function: a.k.a entry point
    - int main(int argc, char * argv[]) {}
  - Functions earlier declared as prototypes
- Syntax 101
  - Statements end in ";"
  - Operators: ==, !=, >=, >, <, !, &&, ||, ++(var), --(var)
  - (var)++, (var)--, tenary op
  - 1: True, 0: False, (void *)0: Null
- Syntax 101 continued
  - Code blocks enclosed in {}, blocks are a sequence of statements executed in batch. E.g body of functions, if/while/for, structs
  - Standard data types: char, int, void, float, double
  - Others: unsigned, [], pointers etc
  - // this is a single line comment.
  - /* this comment may span multiple lines*/
  - Bitwise operators: |, ~, ^, &, etc
- Standard file extension: .c
- Standard compile (using gcc; GNU C compiler):
  - Creates executable called 'a.out'
    - gcc –Wall –g <source_file>
  - Creates executable with specified name <out_file>
    - Gcc –Wall –g <source_file> -o <out_file>
- Standard debugging (using gdb):
  - Source must be compiled with the –g flag

auth.c

Overthewire 6-10

```
// insert necessary include lines
#include <stdio.h>
#include <string.h>

// passwordchar * correct_pass = "s3cr3t";

// insert function prototypes
int auth(char *);

// implement the main function
int main(int argc, char * argv[]){
   char buff[20];
   // fetch password argument
   if (argc == 2) { // prog_name + first_arg
      strcpy(buff, argv[1]); // copy password into buff

      // save in buffer and check with auth
      int i = auth(buff);

      // check access
      if (i) {
        printf("ACCESS GRANTED\n");
      } else {
        printf("ACCESS DENIED\n");
      }
   } else {
      printf("Usage: auth <password>\n");
   }
   return 0;
}

int auth(char *passwd){
   return (strcmp(correct_pass, passwd) == 0); // strcmp returns 0 on match
}
```