Project 2: Crack the Code

CS/COE 0449 — Introduction to System Software

See Due Date on CourseWeb

For this lab, create a new directory named `project2` under your `cs449` directory and create your program there:

```
mkdir project2
cd project2
```

For this lab, you can get the decrypted file (`code.bin`) to your directory using the following command:

```
cp /afs/cs.pitt.edu/usr0/tkosiyat/public/cs0449/code.bin .
```

For those who works on your own computer, the decrypted file is also available on the CourseWeb.

Description

Throughout most of your CS or CoE studies, you work creating or modifying programs or computers in a word: building. However, sometimes the best way to learn about something is to break it.

Part 1: Decrypt (40 points)

In the first part, you are going to write a program that can be used to decrypt a specific type of encryption. Your `decrypt` program should take a filename from the command line and read the bytes of the file, looking for strings of printable characters (ASCII values between 32 and 126 decimal including 10 which is the newline character). A string is a run of at least 4 consecutive printable characters and ends whenever a non-printable character is encountered. Whenever you find such a string, print it out. Note that newline is also a printable character in our case. So, when you print your result, do not go to the newline unless a set of 4 or more printable characters contain newline(s).

You can check the operation of your program via the program named `strings` using the following command:

```
strings code.bin
```

Note that the `strings` program will print the set of 4 or more characters one line at a time and it does not count newline character as a printable character. So, the result will be a little different. Make sure your program can handle strings that are arbitrarily long. Your job is to successfully decrypt the given file named `code.bin`. To see the content of the file in hexadecimal format, you can use the program named `od` as follows:

```
od -x code.bin
```

Part 2: What are those number? (60 points)

Each of you will receive an email from me with an attached file named `USERNAME_program.tar` where `USERNAME` will be your Pitt user ID. Save/Copy the attached file in your `project2` directory and decompress it using the following command:

```
tar -xvf USERNAME_program.tar
```

The result is an executable file named `USERNAME_program`. Try to run this program using the following command:

```
./USERNAME_program
```

If the console says `"Permission Denied..."`, then execute the following command:

```
chmod +x USERNAME_program
```

When you run this program, you will be asked to enter your Pitt user ID to verify that you receive the right executable file. If all when well, the program will show a random 10-digit session number. Then the program will ask you to enter a number five times. Each time if you enter a wrong number, the program will terminate immediately. There is a relation between each number that you have to enter and the given session number. Therefore, every time you run your program, the numbers that you have to enter keep changing. Note that the numbers that you have to enter may be changed each time but the relation between the numbers and the given session number will always the same. Your job is to crack this code. Try to find the relation between the given session number and each number that you have to enter.

The most obvious tool you will need is `gdb` just like what you did in lab 6. `objdump` can also be used if you want to print out the assembly code of the `main()` function. For example:

```
objdump -d USERNAME_program
```

The above command will dump the whole assembly code of your executable file. One long section will be the `main()` function as shown below:

```
080485a4 <main>:
 80485a4:       55                              push   %ebp
 80485a5:       89 e5                           mov    %esp,%ebp
 80485a7:       83 e4 f0                        and    $0xfffffff0,%esp
 80485aa:       83 ec 60                        sub    $0x60,%esp
 80485ad:       65 a1 14 00 00 00               mov    %gs:0x14,%eax
 80485b3:       89 44 24 5c                     mov    %eax,0x5c(%esp)
 :
 804890e:       90                              nop
 804890f:       90                              nop

08048910 <__libc_csu_init>:
```

You can cut and paste just the `main()` section in any text editor and print it out which may help you trace your program.

What to Hand In

Create a text file named `readme.txt` in your `project2` directory. In there, explain the relation between the session number and each of the number that you have to enter. Also include an example of a session number and numbers that you have to enter. For example, your `readme.txt` file may look like the followoing:

```
Pitt ID: abc123

The first number is [...]
The second number is [...]
The third number is [...]
The forth number is [...]
The fifty number is [...]

Example:
Session Number: x
Number 1: x
Number 2: x
Number 3: x
Number 4: x
Number 5: x
```

where `[...]`s are relation between a session number and each specific number that you have to enter and `x`s are numbers. The rest of the process are the usual steps. First, let us go back up to our cs449 directory:

```
cd ..
```

Now, let us first make the archive. Type your username for the USERNAME part of the filename:

```
tar cvf USERNAME_project2.tar project2
```

And then we can compress it:

```
gzip USERNAME_project2.tar
```

Which will produce a `USERNAME_project2.tar.gz` file.

If you work on `cs449.cs.pitt.edu` (`thoth`) you can skip to the next section. **If you use your own machine, you need to transfer the file to `cs449.cs.pitt.edu` first**. This can simply be done by a command line. For example, assume that your username is `abc123` and you are in the same directory as the file `abc123_project2.tar.gz`. To transfer the file to `cs449.cs.pitt.edu` use the following command:

```
scp abc123_project2.tar.gz abc123@cs449.cs.pitt.edu:.
```

The above command will copy the file to your home directory in `cs449.cs.pitt.edu`. If you want to copy it to your `private` directory, use the following command:

```
scp abc123_project2.tar.gz abc123@cs449.cs.pitt.edu:./private/.
```

Copy File to Submission Directory

We will then submit that file to the submission directory:

```
cp USERNAME_project2.tar.gz /afs/cs.pitt.edu/public/incoming/CS0449/tkosiyat/sec1
```

Once a file is copied into that directory, you cannot change it, rename it, or delete it. If you make a mistake, resubmit a new file with slightly different name, being sure to include your username. For example `USERNAME_project2_2.tar.gz`. **Check the due date of this project in our CourseWeb under Labs/Recitations**.