Assignment 7, Fall 2015
CS 4630, Defense Against the Dark Arts
Inserting Virus Code
Injectus Viroso Curse

Purpose

In the previous assignment you modified a binary to subvert control to some virus code. However, the virus code was already in the binary. In this assignment, you will inject some additional code. The problem is to find a place for the code which requires some ingenuity on your part.

Due Date

This assignment is due on Wednesday, 11/11/2015, at 10:00am

Inject Virus Code (Injectus Viroso Curse)

You should inject code into *target_code.bin* so that when the infected file is executed, it prints the following messages:

```
$ ./infected-target.exe
Initialize application.
Cavman virus activating!
You have been infected with the Cavman virus!
Begin application execution.
Terminate application.
$
```

Basically, you should use the tricky jump as before, but instead of transferring to the virus code already in the executable, you will jump to some new code that you have injected that prints the new message

```
Cavman virus activating!
```

After printing this message, control is transferred to the existing virus code as before. The target file (*target_code.bin*) is available on the Collab. This file is different from the file used for Assignment 06. This new *target_code.bin* is specially compiled with extra cavities. Please be sure and use it as this file.

Assignment Details

1. Write a C program called *infect.c* that when compiled and executed reads a Linux executable and produces new infected executable where the “cavman” virus has been inserted. See the previous example output to see the output the infected executable should produce.

2. Your C program will take two parameters. The first parameter is the filename of the to-be-infected Linux executable. The second parameter is the name of the infected Linux executable that your program generates. For example, in the following example, *target_code.bin* is infected and a new executable *infected_target_code.bin* is generated.

```
$ ls
target_code.bin
$ ./infect target_code.bin infected_target_code.bin
$ ls
target_code.bin infected_target_code.bin
```

3. Your program should not change the size of the executable. That is, *target_code.bin* and *infected_target_code.bin* should have identical sizes. The size of file *target_code.bin* is 5660 bytes.

4. Your should infect the *target_code.bin* you download from the collab. **DO NOT compile your own *target_code.bin*.** We provide source code only to help you understand the target_code.bin.

Methodology and Hints

1. You can inject your virus code and the new output string to replace nops.

2. Remember to call VirusCode at the end of your injected code.

3. Instruction “xchg %ax, %ex” is equivalent to a “nop”. In fact, x86 does not have “nop” originally. Today's “nop”, which is “0x90” in machine code, is actually “xchg %eax, %eax”.

Submission Guideline

1. **WARNING: YOUR SUBMISSION ***MUST*** FOLLOW THIS GUIDELINE. THERE WILL BE 40% PENALTY FOR THOSE WHO FAIL TO FOLLOW THIS GUIDELINE.**
2. Submit your *infect.c* to collab. **THE FILENAME MUST BE infect.c.**
3. Your program will be compiled with the following command:

   ```
   gcc -m32 -o infect infect.c
   ```

   You will receive 0 points if your program fails to compile with this command.
4. **Your program MUST take two parameters.** The first parameter is the filename of the to-be-infected executable. The second parameter is the filename of the infected executable that your program generates. More information can be found in Assignment Details.