Network Security Assignment #2

Written Assignment-1 & Programming Assignment-2

PART A - Written Assignment -1

Research on the following and write a short write up on what you can find about the following:

1. What is the difference between a binary file and a text file?
2. What is the ELF FILE Format? 2. What is a hexdump?
3. What kind of information is stored in ahexdump?
4. **Diagrammatically** represent how the hex characters "DEADBEEF" will be represented in Big Endian and Little Endian formats.
5. There's questions embedded in red in the programming assignment which you will have to answer as well ☺

PART B - Programming Assignment -2

For the purposes of this assignment install openssl, nasm and ghex if you already don't have it. First off, read through the entire write up before you start performing the steps. See if you can get an idea on where this is headed. The steps are for an x86 system. Since the steps are all very simplistic am assuming that it will work well for any other architecture, though I haven't tested it out on anything other than an x86.

The output of the program and your write up of the written assignment can also be included as a part of the written assignment in the final submission.

1. Write a simple Hello World program in c. The program should have nothing but a #include and a print statement in *main*. Write a makefile to generate an executable called 'hello' Record the size in number of bytes of the hello world program

2. Read and dump the ELF header from the executable type the command
   readelf -h hello

3. To show all the dynamic linked libraries use the command ldd
   ldd hello

4. To get a description of what the file does do
   file hello
   The output of the file command will show 'not stripped'

5. Strip the file using the following command
   What does the strip command do?
   strip –s hello

6. Now, open a file called hello.asm and copy the following lines of code into it.

```
SECTION .data
msg: db "Hi World",10 len: equ $-msg
SECTION .text
global main main:
mov edx,len mov ecx,msg mov ebx,1 mov eax,4
int 0x80 mov ebx,0 mov eax,1 int 0x80
```

7. Compiling the asm
   nasm -f elf hello.asm

8. Compile & strip
   gcc -o hello hello.o -nostartfiles -nostdlib –nodefaultlibs
   strip –s hello

9. Open the ghex hexeditor. Using the editor replace the strings "Hi World" with the string "BUSTED". Also embed a signature "DEADBEEFDEAD". Remove all remaining bytes after the end of the signature.

10. Record the number of bytes on the executable. It should have come down to less than 300 bytes

11. Write a C program that:
    - Reads the *'hello'* program as bytes and store its contents in memory.
    - Dump the contents of the file and compare it with what you see in the ghex editor and hexdump command to see if it matches.
      Write out your comments about what you see
    - Looks for any binary *.bin file, overwrites the contents of the binary file with the hello program. The overwritten file should also be given execute permissions if it already doesn't have it. Running this program should display BUSTED as its output
    - Binary files may either be hand created using the ghex editor or can be created very simplistically using a perl or python script.

12. Write a scanner c program (a process that runs in a loop) that scans all files in a directory and computes an MD5 hash of all the files in the directory and stores it. Every T seconds the scans to see if the hash of a file is now different from the original hash. IF yes, it looks to see if the signature DEADBEEFDEAD exists in the file. The program can display an alert stating that the file has been infected. Typing Q should kill the process. The scanner program is run first and then the virus program is run to detect the infected files.