CECS 478 Assignment 3 – Exploitation

1) Describe what the following shell commands do:
   a) env
      i) allows user to print a list of environment variables or run a program in a different environment
   b) su
      i) allows user to change user accounts, you must enter the password for the new use as well
   c) echo $SHELL
      i) allows user to print to the screen SHELL variables, also SHELL is the user's preferred shell, which is the current preferred account, which is where it gets its output from
   d) cat /etc/passwd >> ~/pfl
      i) allows user to see a file that contains user name, encrypted password, user id number, full name of the user, directory, and login shell
   e) perl -e 'print "1\n5\nn\n5\n" . "A"*100 . "\x70\ x8d\x04\x08\n" . "1\nn\n" . "7\n"'
      i) allows user to create a simple perl script that prints to the screen random stuff, such as
         1
         5
         n
         5
         AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
         AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAp x8d
         1
         n
         7

2) What types of files are contained within the following Linux directories:
   a) /root
      i) bin/ dev/ home/ lost+found/ proc/ sbin/ usr/ cdrom/ opt/ vmlinuz boot/ etc/ lib/ mnt/ root/ tmp/ var/ dvd/ floppy/ initrd/ /tftpboot
      ii) these are all subdirectories of root which contains the system administrator files as well as the files necessary for mounting the partition when booting up the computer
   b) /sbin
      i) Files in this directory are usually files used for system maintenance or administrative tasks rather than application executables; this refers to all the files needed for background system processes
   c) /etc
      i) Files in this directory contain all the system related configuration files that are needed to control the operation of a program
   d) /var
      i) Files in this directory are variable data files such as system logging files, mail and print spooler directories, and temporary files
   e) /usr

    i) Files in this directory are usually big in quantity as these files pertain to the user, their created files, libraries, header files, etc…, including user programs
3) Explain what the following entry in /etc/passwd means:
    a) myroot:XXq2wKiyI43A2:0:0:me:/root:/bin/bash
        i) This entry attempts to append the following hashed encrypted password to the /etc/passwd which is a file that contains all usernames, IDs, and login shells for all users of the system
4) Section 0x321 in your textbook discusses debuggers applied to code vulnerability examination. How might you use a debugger to prepare for an exploitation?
    a) I would use a debugger to create breakpoints in the code so that I could examine what the buffers within the pointers are located at or pointing to. This way I would know where to direct the next command to exploit an open buffer. I am also able to see all the elements of a stack in a debugger enabling me to see where another execution control point does exist within the stack variables to overwrite it. Because this is more memory, when it's overwritten, it will result in a program crash.
5) How can the shell environment be useful for performing an exploitation attack?
    a) The shell environment will allow me to inject several instructions at once and if done correctly bring me back to the return execution step. This *shellcode* allows me to gain access to the root account where I can carry out my attack.
6) How might I exploit a heap? A stack? What Linux tools would you use for conducting each?
    a) Heap
        i) I would have to know the memory addresses of two buffers ad the distance between them so that I can overflow the second one with data such as a string. This way the program will add the string to a new file contained in the root directory which also gives me root permissions to access the file. From here, I can run any program to exploit the root directory.
    b) Stack
        i) I could write a program that works by corrupting memory to control execution flow. For example, if I generate a command string that executes another program with a command-line argument between single quotes, get the string's length, and concatenate the closing single quote to the end, I can execute the command string. The buffer that is generated between the single quotes is what we are trying to get to here. This way it serves a root shell which provides full control of the computer.
    c) I would use either the C or Perl language, BASH to execute my programs, a text editor, and a debugger.
7) What is the %s format parameter used for? Give an example. How might I use it to write out an exploit?
    a) %s
        i) Takes the next argument and print it as a string; it's expected to be a pointer to an array of character type, so pointer to a string
    b) printf("[BEFORE] buffer_two is at %p and contains \'%s\'\n", buffer_two, buffer_two);
8) Big-endian and little-endian are two ways that computer architectures deal with numbers. What are they? Which one does the Intel x86_64 architecture use?
    a) Big-endian and little-endian are ways of storing bytes of numbers and how they are stored in a computer's memory.

b) Intel x86 uses little-endian architecture

9) How do exploits take advantage of direct parameter access?
   a) Exploits directly access a memory location given a parameter the special character "$" which allows exploits to find exactly what is needed rather than having to step through memory looking for the beginning of the format string location.
10) Define a *short write*. Give an example of using one in an exploit.
   a) A short write is a technique that can simplify format string exploits using a two-byte word.
   b) [*] test_val @ 0x08049794 = -65515 0xffff00**15**