❏ HW 2

❏ lev1

- ❏ I first downloaded tunnelblick and accessed the vpn.
- ❏ I ssh'd into razor while running tunnelblick.
- ❏ I read the writeup, copied the hashed password into a file on kali, and ran 'john filename.'
- ❏ John the ripper-the-ripper cracked the password and gave me 'security' which I entered and put me into group lev1.
- ❏ I then used the command 'l33t' to permanently add me to lev1.
- ❏ I logged out and then in to make sure I was permanently added.

❏ lev1-lev2

- ❏ I used 'ls' to see what was in /var/challenge/level1.
- ❏ I see that '1.c' takes a file called '.secret' and hashes the contents with md5. If the hashed contents match the hash in the file it execs a shell
- ❏ I ran the file '1' and saw that it was trying to cat out a file '.secret' that I didn't have. I used ls -al to see that there was a hidden file '.secret' that had a group id of 'lev2'
- ❏ I took the hash out of '1.c' and used the online md5 cracker 'http://www.hashkiller.co.uk/md5-decrypter.aspx' to return to me a the hex of the cracked md5 input. I took that input and converted it to ascii and echo'd 'grabthis' into .secret.
- ❏ I ran '1.c' again and it gave me a shell that inherited the group 'lev2'. I ran 'l33t' and was added to group 'lev2.'

❏ alternate lev2 (aka. HOME redirect)

    - Change the HOME environment variable to the path where .secret is in in /var/challenge/level1
    - then run ‘./1’ and it will spawn a shell because the unhashed password was hidden in .secret
- lev2-lev3
    - Looking at the code from ‘2.c’ I noticed the execlp was called. From lecture I realized you could change the search path of execlp to where ever you wanted.
    - I also noticed that the sgid bit was set which meant that when I run the program I will run as the group owner of the file.
    - In my home directory I created a file called ‘tidy’ and in the file I initially put ‘bin/sh’ so that when it was called it would execute a shell.
    - I then changed path with: ‘export PATH=~’ to my home directory so that execlp would execute the tidy found there.
    - When I ran it nothing happened, so after doing some reading I discovered that that execlp will switch the current process to whatever command is called. This also means that the new process retains the privileges, if sgid is set, of the group that owns it.
    - I figured then that I could exec l33t in a higher group because the process was owned by the higher group.
    - In my tidy I replace ‘/bin/sh’ with ‘l33t’ and ran it once again.
    - I checked the score and I had advance to lev3.
- lev3-4 (format string attack) ***actually not a format string attack***
    - This is an exploit of the ‘find’ command spawning a subshell and executing files in that subshell.

  - ❑ At first I thought this was a format string attack because of snprintf but Professor Black corrected me.
  - ❑ He suggest I take a look at ‘find -exec’ which I did and that was the solution: ./3 “tidy (which has l33t in it) -exec {} +”.
  - ❑ You have to put all of the args in one string otherwise they will be run individually. You have to use {} + instead of \; because the program is checking for those things.
- ❑ lev4-5 (not a buffer overflow but a directory traversal exploit)
  - ❑ Again, I wasted a bunch of time trying to figure out what this was.
  - ❑ I spoke to Professor Black and he mentioned a directory traversal which I thought I had tried before but to no success. What I found out from someone else is that I wasn’t traversing enough directories to to do it right.
  - ❑ The program has a path of /var/challenge/level4/bin/devel. Knowing this you need to traverse back 5 directories to / (the root directory) and go into /usr/local/bin/l33t so that 4.c will execute that command
  - ❑ Code: ./4 ../../../../../usr/local/bin/l33t
  - ❑ The ../ changes the path variable found in 4.c. When it is sent into execv the path really become /usr/local/bin/ and not /var/challenge…