**Crypto and Network Security**

**Lab 9**

**Name: GAURAV SETTY**

**Email: settgm01@pfw.edu**

**Task 1 : Generating Two Different Files with the Same MD5 Hash**

Let's make a prefix.txt file and fill it with text. Now let's look at the file size. And it’s 57 bytes.

Text

Description automatically generated

Let's now execute the md5collegen command to create two binary files with the same md5 hash.

The generation of the files took 0.69 seconds.

Now let’s compare the files to see if they are similar and let’s generate MD5 Checksums for the files.

Text

Description automatically generated

As we can see, the MD5 checksums are identical.

Check the files on bless, both of them.

A screenshot of a computer

Description automatically generated

The files' differences are visible right away. The chosen characters are visible to us.

**Question1 : If the length of your prefix file is not multiple of 64, what is going to happen?**

**Answer:** The text in the Prefix.txt file is given padding to make it a multiple of 64.

**Question2 : Create a prefix file with exactly 64 bytes, and run the collision tool again, and see what happens.**

**Answer:**

Updated the existing prefix.txt file to make it 64bytes.

Text

Description automatically generated

Now execute md5collgen:

Text

Description automatically generated with low confidence

Text

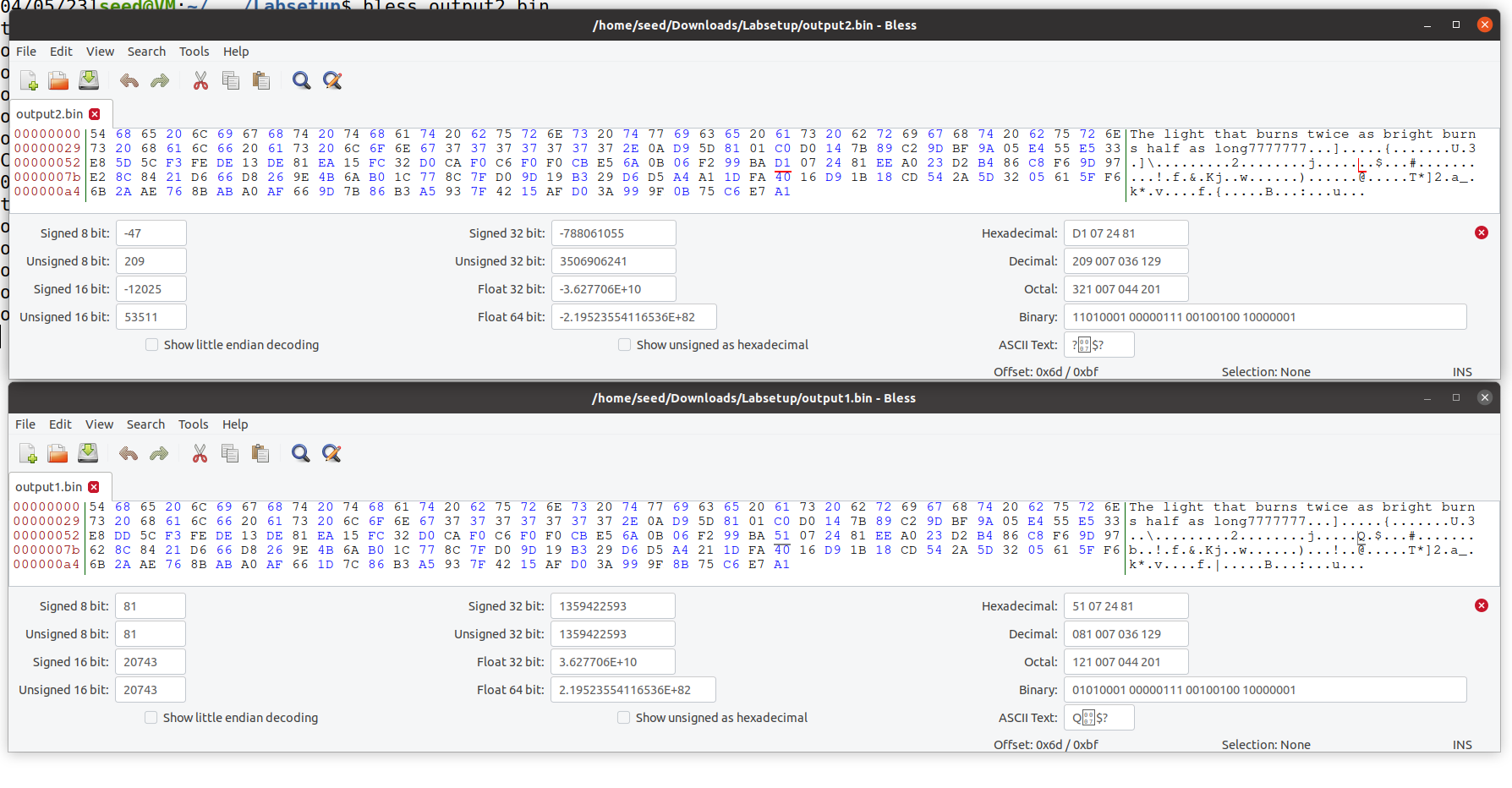
Description automatically generated

The file is same as before.

**Q3 : Are the data (128 bytes) generated by md5collgen completely different for the two output files? Please identify all the bytes that are different.**

**Answer:**

Let’s check the bless output.



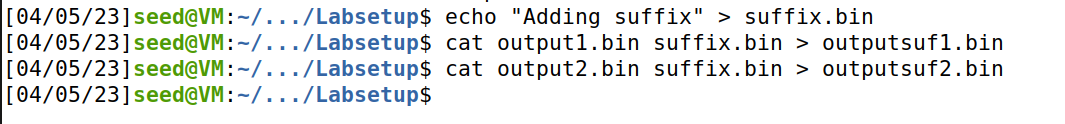
**TASK 2: Understanding MD5’s Property**

We'll use the two binary files we created in the previous task—output1.bin and output2.bin—since we already know they're distinct but share an identical md5 sequence.

Text

Description automatically generated

Now let's make a file that will serve as a suffix and add it to the two bin files that were previously displayed.



Check the md5 hashes of the two fresh files right away.

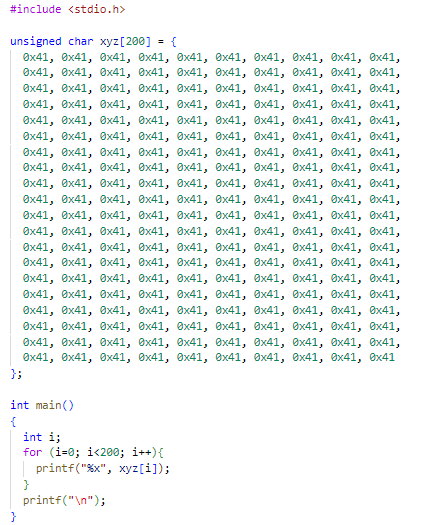
Here, we can see that the hash does not change even after the bin folders have a suffix added.

Text

Description automatically generated

**Task 3: Generating Two Executable Files with the Same MD5 Hash**

Code:



Let's now assemble this code to create the file a.out.



Now, let’s check the bless output.

Find the offset so that the array's six rightmost numbers are zero now.

A screenshot of a computer

Description automatically generated

The file will now be divided at 12352, which we will use as a prefix.

After that, create two folders with the same md5 hash using md5Collgen. Also, we can see the files generated.

Text

Description automatically generated

The out1.bin and out2.bin are precisely 128 bytes larger than the prefix, with a combined size of 12480. This indicates that the 128 was inserted as a suffix by the md5collgen.

Let's save additions from md5collegen files onto 2 new files and name them P and Q and append into new files, a1.out and a2.out, and take everything after 12481 offset from a.out and put it in a new file and call it suffix.

Text

Description automatically generated

Let’s compare a1.out and a2.out. It shows that files are not same.

Also, Let’s take a look at a1.out and a2.out

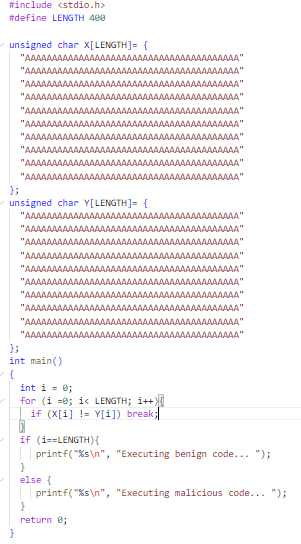
A picture containing chart

Description automatically generated

Here, we can see that both are same.

**Task 4: Making the Two Programs Behave Differently**

Code:

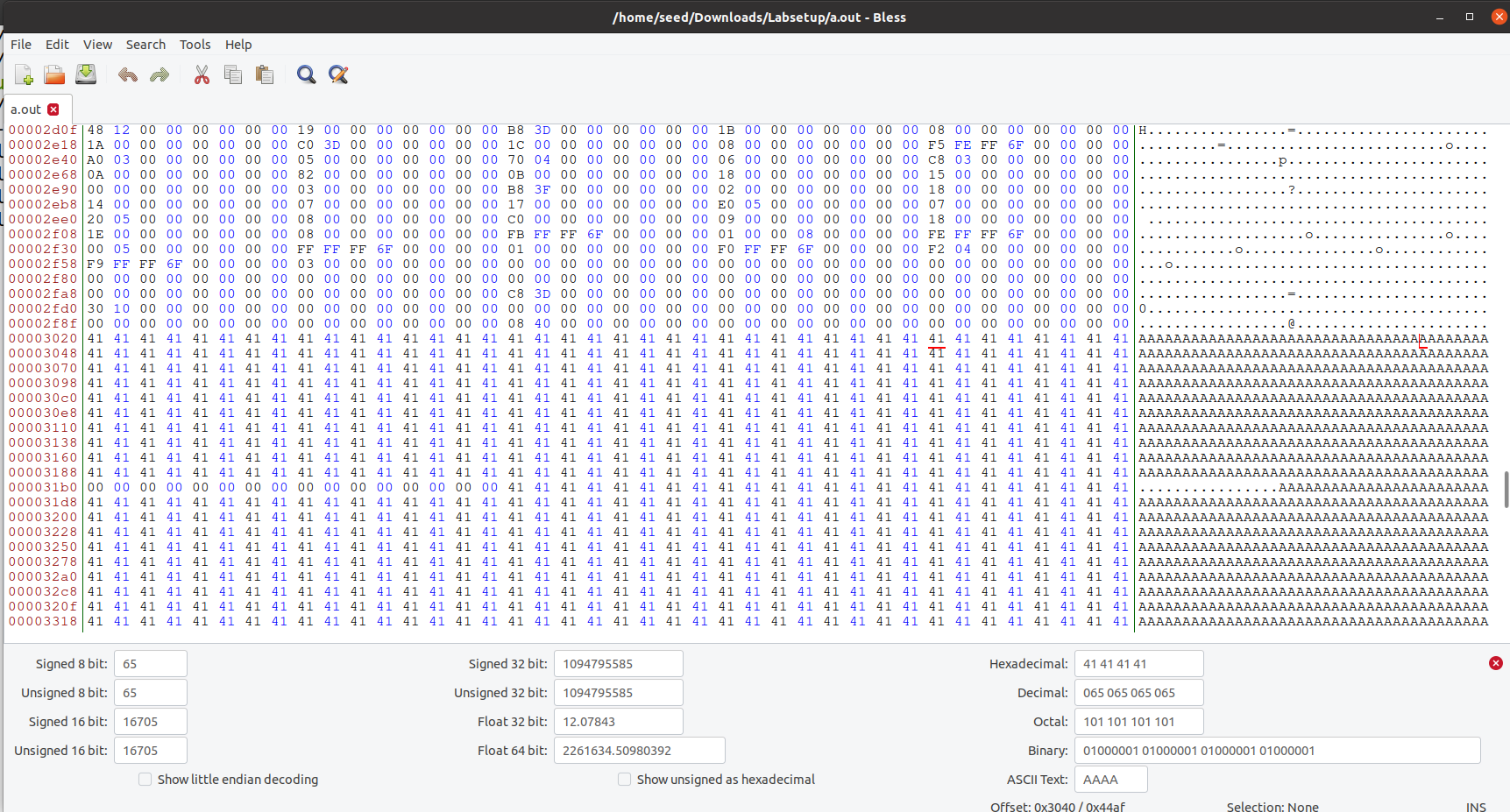


Compile the code to generate a.out file.

Text, Word

Description automatically generated

Let’s take a look at the bless output.



Find an interval that is just before the multiple of 64. We'll use 12352 as our example.

Now let's execute md5collgen on the first 12352 bytes of the program that were copied into prefix.

Text

Description automatically generated

The following 128 bytes of the file created by the md5Collgen generator should now be replaced with P. Since there are 128 bytes left after 12352, we select 12481.

The last 128 bytes produced by the md5collgen are used to create P and Q. With 128 bytes separating them, we will divide the suffix file into suffix1 and suffix2. This will make the array's contents malicious because they will be the same in one instance and different in the other.

Text

Description automatically generated

Text

Description automatically generated

Text

Description automatically generated