**LAB 1 – MD5 Collision Attack Lab**

**Due: Wednesday Sep 29, 2:30pm (before the class starts)**

**Turn in: this lab report**

**Points: 60 pts**

Please explain in detail, and make sure the screenshots are easy to read.

(20 pts) Task 1: Generating Two Different Files with the Same MD5 Hash

* (7 pts) Question 1. If the length of your prefix file is not multiple of 64, what is going to happen? (Screenshot needed)
  + md5collgen adds 00s onto the file until it reaches a multiple of 64.
* Graphical user interface, text, application, table, email, Excel

  Description automatically generated
* (7 pts) Question 2. Create a prefix file with exactly 64 bytes, and run the collision tool again, and see what happens. (Screenshot needed)
* Graphical user interface, text, application, table, email

  Description automatically generated
* (6 pts) Question 3. Are the data (128 bytes) generated by md5collgen completely different for the two output files? Please identify all the bytes that are different. (Screenshots needed, circle/highlight the different bytes)
  + The bytes are actually mostly the same. There are only 3 places that changed.

Text

Description automatically generated

(10 pts) Task 2: Understanding MD5’s Property

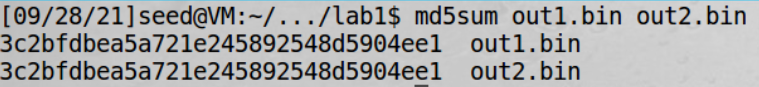
Explain in detail about how you finish Task2.

* (5 pts) Screenshots needed to show M and N are different, but their MD5 hash values are identical.
  + Different Files:

Table, calendar

Description automatically generated

* Same Hash:



* (5 pts) Have screenshots to show you concatenated the files and your MD5(M||T) and MD5(N||T) are identical. (T can be a simple .txt file)

Text

Description automatically generated

* + The files start with the same hash, and after concatenating the same file to each of them, the outputs still have the same hash.

(20 pts) Task 3: Generating Two Executable Files with the Same MD5 Hash

* (5 pts) Take a screenshot of your C/C++ file used in this task.
* Text

  Description automatically generated
* (5 pts) How did you find the end position of the prefix? (Screenshot needed to explain)
  + I looked for the list of As, then I looked at the Offset at the bottom of the Bless windows to get the character location.
  + 0x1044 = 4164
  + Graphical user interface, text, table

    Description automatically generated
* (5 pts) Take a screenshot of the commands you used to cut and glue them.

Text

Description automatically generated

* + The first 4 commands split up the file, the last 2 check to make sure the first 4 worked right. In this case, there was no difference, so the commands worked.
* (5 pts) After you have the final two files, take two screenshots to show the contents of each of them (run **bless** and just screenshot the part around the “128 bytes” which glued to the prefix and suffix, I don’t need other part of the executable).

Graphical user interface, text, application, table, email

Description automatically generated

Graphical user interface, text, application, table, email

Description automatically generated

(10 pts) Conclusion:

A summary of the lab and What have you learned in this lab?

* In this lab we learned why the MD5 hash function is not the most secure. I’ve learned that the MD5 function is iterative, so you can append data with the same hash to two files with the same hash, and their new hashes will still match. One thing that surprised me was how little you may have to change; for one of the in Task 2 only 2 characters where changed between the output binaries to create a file with the same hash.

Also answer these questions:

What is a one-way hash function?

* + A one-way hash function takes in a file of variable length as an input then outputs a fixed-length seemingly random string of characters. Ideally, it will be very hard to find a hash collision, and it is impossible to calculate the original input from any given output.

What is a collision?

* + A collision is when a file with different content hashes to the same string as another file. This is a problem for security, as you may *think* that the file has not been changed when in fact it has.