

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

Initial few steps:

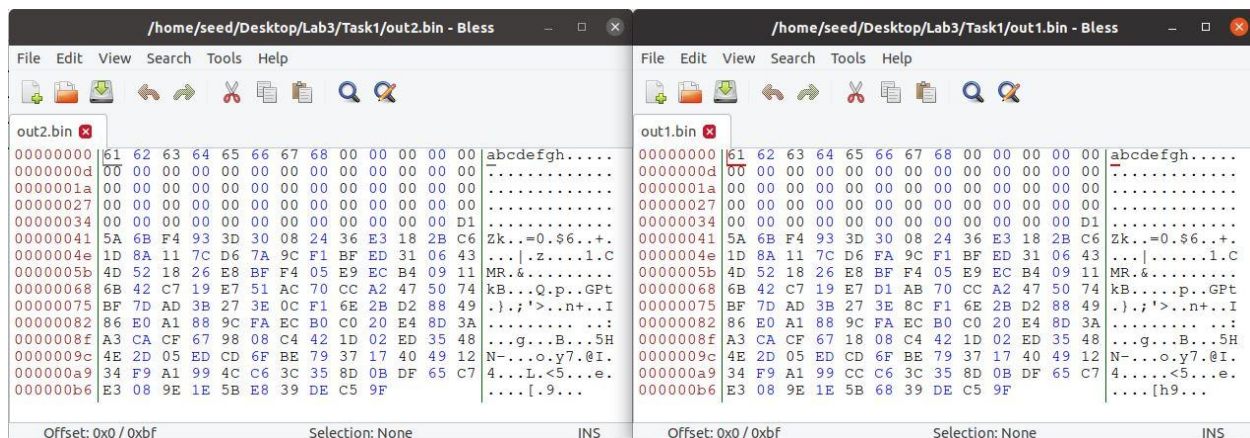
1. Creating files using prefix.txt whose value is "abcdefgh".
2. Checking difference between them

```
seed@VM: ~/.../Task1
[06/22/22]seed@VM:~/.../Task1$ md5collgen -p prefix.txt -o out1.bin out2.bin
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'out1.bin' and 'out2.bin'
Using prefixfile: 'prefix.txt'
Using initial value: a0b50a6e2edc375684f9575884b87364

Generating first block: .
Generating second block: W.....
Running time: 2.09926 s
[06/22/22]seed@VM:~/.../Task1$ diff out1.bin out2.bin
Binary files out1.bin and out2.bin differ
[06/22/22]seed@VM:~/.../Task1$ md5sum out1.bin
c458c0654bb51636ea5c9e2a0404e731 out1.bin
[06/22/22]seed@VM:~/.../Task1$ md5sum out2.bin
c458c0654bb51636ea5c9e2a0404e731 out2.bin
```

3. Showing each of the files using 'bless' editor



```
/home/seed/Desktop/Lab3/Task1/out2.bin - Bless
File Edit View Search Tools Help
out2.bin
00000000 61 62 63 64 65 66 67 68 00 00 00 00 00 00 00 00 |abcdefgh....
0000000d 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
0000001a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
00000027 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
00000034 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
00000041 5A 6B F4 93 3D 30 08 24 36 E3 18 2B C6 Zk..=0.$6..+.
0000004e 1D 8A 11 7C D6 7A 9C F1 BF ED 31 06 43 ...|.z....1.C
0000005b 4D 52 18 26 E8 BF F4 05 E9 EC B4 09 11 MR.&.....
00000068 6B 42 C7 19 E7 51 AC 70 CC A2 47 50 74 kB...Q.p..GPt
00000075 BF 7D AD 3B 27 3E 0C F1 6E 2B D2 88 49 .).;'>..n+..I
00000082 86 E0 A1 88 9C FA EC B0 C0 20 E4 8D 3A .....
0000008f A3 CA CF 67 98 08 C4 42 1D 02 ED 35 48 ...g...B...5H
0000009c 4E 2D 05 ED CD 6F BE 79 37 17 40 49 12 N-...o.y7.@I.
000000a9 34 F9 A1 99 4C C6 3C 35 8D 0B DF 65 C7 4....<5...e.
000000b6 E3 08 9E 1E 5B E8 39 DE C5 9F ....[.9...

Offset: 0x0 / 0xbf Selection: None INS

/home/seed/Desktop/Lab3/Task1/out1.bin - Bless
File Edit View Search Tools Help
out1.bin
00000000 61 62 63 64 65 66 67 68 00 00 00 00 00 00 00 00 |abcdefgh....
0000000d 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
0000001a 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
00000027 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
00000034 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....
00000041 5A 6B F4 93 3D 30 08 24 36 E3 18 2B C6 Zk..=0.$6..+.
0000004e 1D 8A 11 7C D6 FA 9C F1 BF ED 31 06 43 ...|.z....1.C
0000005b 4D 52 18 26 E8 BF F4 05 E9 EC B4 09 11 MR.&.....
00000068 6B 42 C7 19 E7 D1 AB 70 CC A2 47 50 74 kB...p..GPt
00000075 BF 7D AD 3B 27 3E 8C F1 6E 2B D2 88 49 .).;'>..n+..I
00000082 86 E0 A1 88 9C FA EC B0 C0 20 E4 8D 3A .....
0000008f A3 CA CF 67 18 08 C4 42 1D 02 ED 35 48 ...g...B...5H
0000009c 4E 2D 05 ED CD 6F BE 79 37 17 40 49 12 N-...o.y7.@I.
000000a9 34 F9 A1 99 CC C6 3C 35 8D 0B DF 65 C7 4....<5...e.
000000b6 E3 08 9E 1E 5B 68 39 DE C5 9F ....[h9...
```

Ques1. When the prefix file is not in the multiple of 64, it will be padded with values such as '1' or '0' like the one above with only 5Bytes value.

Ques2.

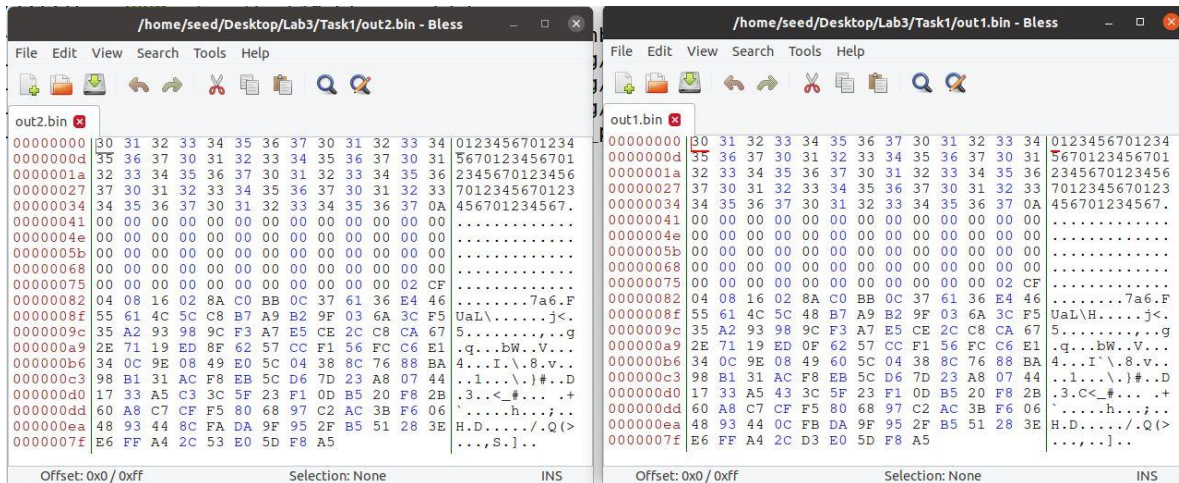
Initial code :

```
[06/22/22]seed@VM:~/.../Task1$ md5collgen -p prefix.txt -o out1.bin out2.bin
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'out1.bin' and 'out2.bin'
Using prefixfile: 'prefix.txt'
Using initial value: 1b6a290fd586f52bb2c2b14af5eedccd

Generating first block: .....
Generating second block: S10.....
Running time: 51.2414 s
[06/22/22]seed@VM:~/.../Task1$ diff out1.bin out2.bin
Binary files out1.bin and out2.bin differ
[06/22/22]seed@VM:~/.../Task1$ md5sum out1.bin
34f0c54f0741bab5ebd59da553dbceb8 out1.bin
[06/22/22]seed@VM:~/.../Task1$ md5sum out2.bin
34f0c54f0741bab5ebd59da553dbceb8 out2.bin
```

Bless editor :



No padding is observed amongst the two files.

Ques3.

01234.....	01234.....
.....
.....
.....
.....hh
C...+.&0.....	C...+.&0.....
...m..C...MN	...m..C...MN
q.C.(...L.{b	q.C.(...L.{b
.Oq...Ws3/...>	.Oq...r3/...>
[.O.Kq'.....	[.O.Kq'.....
`.....UF{l.s	`.....UF{l.s
...G...L.y..	...G...L.y..
M ..=P?...gV	M ..=P?...gV
1...\$....q.o.	1...\$....q.o.
3k.S.....	3k.S.j....

What we observe here, that prefix file and padding consist of same particular data. The only difference is of P and Q which are differences like 'Ws' and 'r'

Task 2: Understanding MD5's Property

After applying the following steps:

1. Creating two files with MD5 hash
2. Checking whether they have the same Hash values.
3. Creating a new file
4. Concatenating the new file as a suffix
5. Checking whether the hash values changed

```
seed@VM: ~/.../Task2
[06/23/22]seed@VM:~/.../Task2$ echo -n "01234" > p1.txt
[06/23/22]seed@VM:~/.../Task2$ md5collgen -p p1.txt -o e1.txt e2.txt
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'e1.txt' and 'e2.txt'
Using prefixfile: 'p1.txt'
Using initial value: ed166457be6f75c4eb07fd002255f072

Generating first block: .....
Generating second block: S11...
Running time: 29.3287 s
[06/23/22]seed@VM:~/.../Task2$ md5sum e1.txt e2.txt
8a0e7bbe37d6b4acebaeaaafb43fa307  e1.txt
8a0e7bbe37d6b4acebaeaaafb43fa307  e2.txt
[06/23/22]seed@VM:~/.../Task2$ echo -n "56" > file.txt
[06/23/22]seed@VM:~/.../Task2$ cat e1.txt file.txt > e1.txt
[06/23/22]seed@VM:~/.../Task2$ cat e2.txt file.txt > e2.txt
[06/23/22]seed@VM:~/.../Task2$ md5sum e1.txt e2.txt
9f61408e3afb633e50cdf1b20de6f466  e1.txt
9f61408e3afb633e50cdf1b20de6f466  e2.txt
```

From the above experiment we can conclude that **the hash value doesn't change after the addition of a suffix and this property holds true for MD5.**

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

Applying the aforementioned steps:

1. Initializing the values of the array
2. Generating the output in the file 'Task.out'
3. Splitting the file 'Task.out' into 'prefix' and 'suffix'
4. Generating two hash file from prefix, 'first' and 'second'
5. Checking each of their values
6. Adding suffix in them
7. Checking their values


```

3 unsigned char xyz[200] = {
4     0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
5     0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
6     0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
7     0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
8     0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
9     0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
10    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
11    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
12    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
13    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
14    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
15    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
16    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
17    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
18    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
19    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
20    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
21    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
22    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
23    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
24 };

```

(step1)

```

seed@VM: ~/.../Task3
[06/23/22]seed@VM:~/.../Task3$ head -c 3200 task.out > prefix
[06/23/22]seed@VM:~/.../Task3$ tail -c 100 task.out > suffix
[06/23/22]seed@VM:~/.../Task3$ tail -c 3300 task.out > suffix
[06/23/22]seed@VM:~/.../Task3$ md5collgen -p prefix -o first second
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'first' and 'second'
Using prefixfile: 'prefix'
Using initial value: 1cc4ada64bcac7187470dc76f3cb20a6

Generating first block: .....
Generating second block: 501...
Running time: 70.5609 s
[06/23/22]seed@VM:~/.../Task3$ md5sum first second
a9d6177b14dab8334556ed115b0eb31e first
a9d6177b14dab8334556ed115b0eb31e second
[06/23/22]seed@VM:~/.../Task3$ cat first suffix > first
[06/23/22]seed@VM:~/.../Task3$ cat second suffix > second
[06/23/22]seed@VM:~/.../Task3$ md5sum first second
24049f4c72e5857437fb0472f558e039 first
24049f4c72e5857437fb0472f558e039 second

```

(step 2 to 7)

```

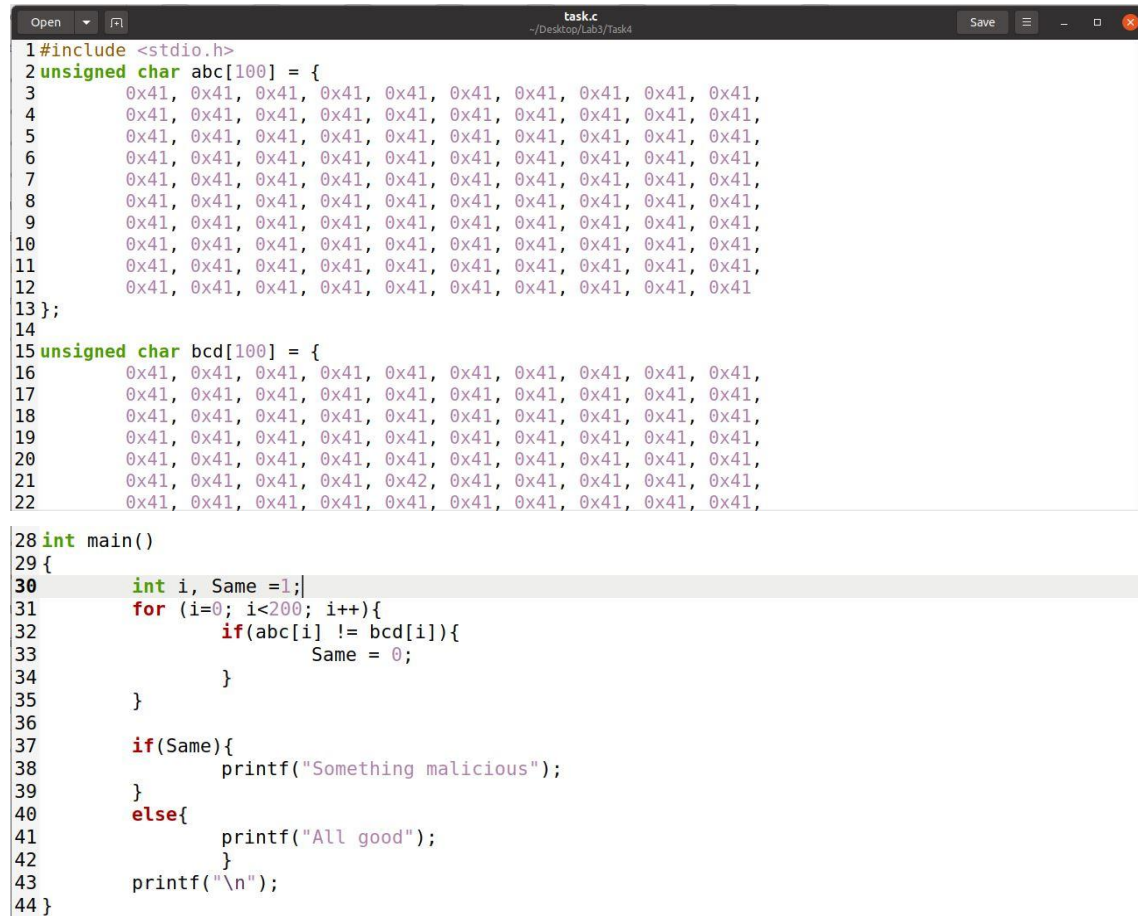
[06/23/22]seed@VM:~/.../Task3$ diff first second
Binary files first and second differ

```

Generation of two files from one executable file does lead with the **same MD5 hash values with the presence of different prefix and suffix values.**

Task 4: Making the Two Programs Behave Differently

Making an executable C program:



```
1#include <stdio.h>
2unsigned char abc[100] = {
3    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
4    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
5    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
6    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
7    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
8    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
9    0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
10   0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
11   0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
12   0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41
13};
14
15unsigned char bcd[100] = {
16   0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
17   0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
18   0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
19   0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
20   0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
21   0x41, 0x41, 0x41, 0x41, 0x42, 0x41, 0x41, 0x41, 0x41, 0x41,
22   0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41, 0x41,
23};
24
25int main()
26{
27    int i, Same = 1;
28    for (i=0; i<200; i++){
29        if(abc[i] != bcd[i]){
30            Same = 0;
31        }
32    }
33    if(Same){
34        printf("Something malicious");
35    }
36    else{
37        printf("All good");
38    }
39    printf("\n");
40}
```

Making Files with same MD5 hash codes, but resulting in different outcomes.

```
[06/23/22]seed@VM:~/../Task4$ bless a.out
Gtk-Message: 04:46:30.469: Failed to load module "canberra-gtk-module"
Could not find a part of the path '/home/seed/.config/bless/plugins'.
Could not find a part of the path '/home/seed/.config/bless/plugins'.
Could not find a part of the path '/home/seed/.config/bless/plugins'.
Could not find file "/home/seed/.config/bless/export_patterns"
[06/23/22]seed@VM:~/../Task4$ head -c 12384 a.out > prefix
[06/23/22]seed@VM:~/../Task4$ md5collgen -p prefix -o first second
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'first' and 'second'
Using prefixfile: 'prefix'
Using initial value: 435efc7f377dd8977d98e98f877603d2

Generating first block: .....
Generating second block: S10.....
Running time: 53.3781 s
```

Combining them with different suffixes, and making them do malicious activities.

```
seed@VM: ~/.../Task4
[06/25/22] seed@VM:~/.../Task4$ tail -c +12532 a.out > badCode
[06/25/22] seed@VM:~/.../Task4$ tail -c +12533 a.out > goodCode
[06/25/22] seed@VM:~/.../Task4$ cat first >> badCode
[06/25/22] seed@VM:~/.../Task4$ cat second >> goodCode
[06/25/22] seed@VM:~/.../Task4$ chmod +x first second
[06/25/22] seed@VM:~/.../Task4$ ./first
Something malicious
[06/25/22] seed@VM:~/.../Task4$ ./second
All good
```

Even though they have the MD5 hash values, simple difference in suffix makes them behave differently and possibly even maliciously.