Applied Cryptography (UE20CS314)

Lab 6

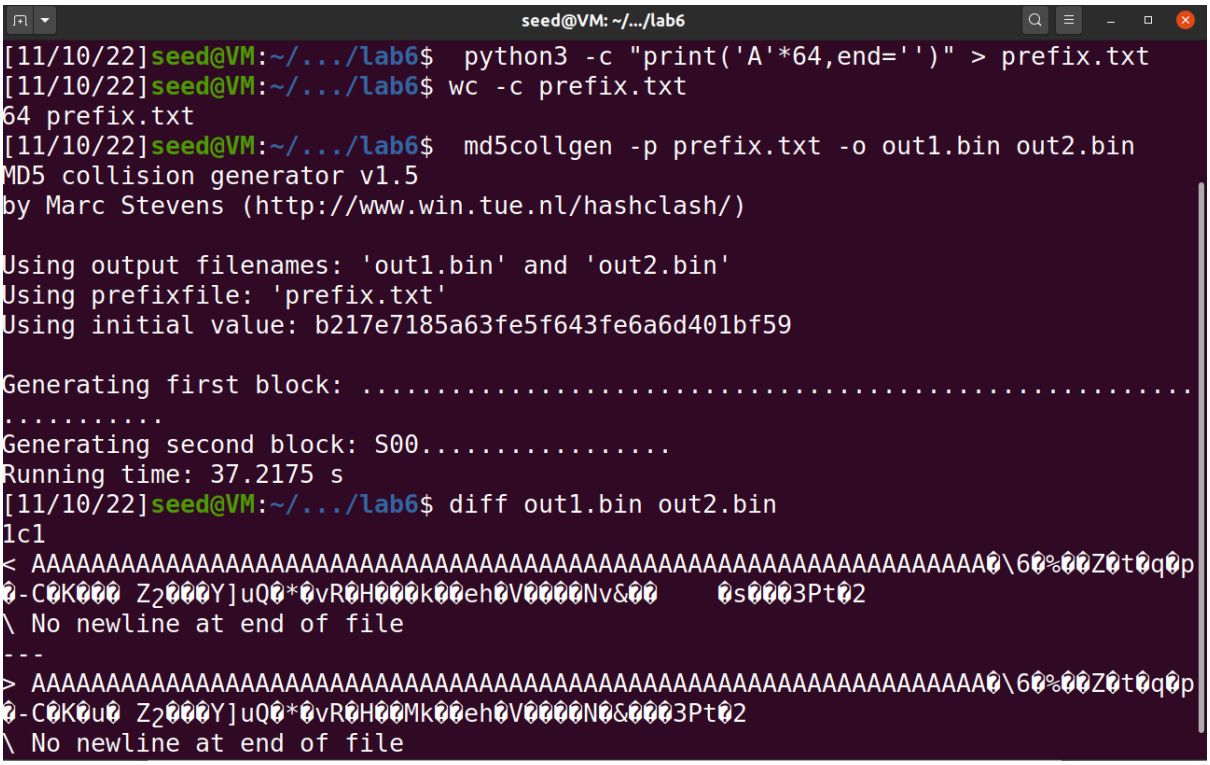
Name: Vishwa Mehul Mehta

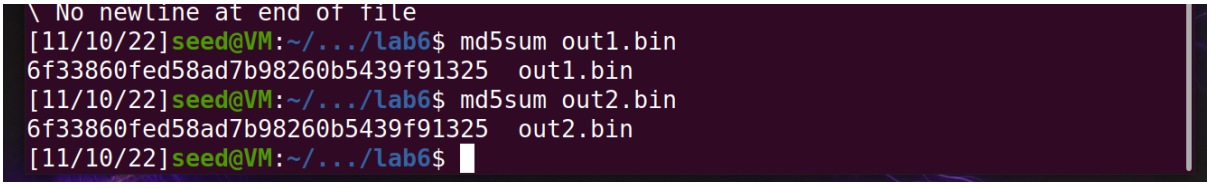
SRN: PES2UG20CS389

Section: F

Task 1:

Screenshot:





```
[11/10/22]seed@VM:~/.../lab6$  python3 -c "print('A'*50,end='')" > prefix.txt
[11/10/22]seed@VM:~/.../lab6$ wc -c prefix.txt
50 prefix.txt
[11/10/22]seed@VM:~/.../lab6$  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: ba06a663ead50f2a10fad4fcdc369853

Generating first block: ....
Generating second block: S00....
Running time: 4.02303 s
[11/10/22]seed@VM:~/.../lab6$ diff out1.bin out2.bin
Binary files out1.bin and out2.bin differ
[11/10/22]seed@VM:~/.../lab6$ md5sum out1.bin
ebd58af7ebd5aa6765aca5639bbe87b2  out1.bin
[11/10/22]seed@VM:~/.../lab6$ md5sum out2.bin
ebd58af7ebd5aa6765aca5639bbe87b2  out2.bin
[11/10/22]seed@VM:~/.../lab6$
```

Observation:

When the length is not a multiple of 64 the outputs out1.bin and out2.bin differ but they have the same md5sum.

Task 2:

Screenshot:

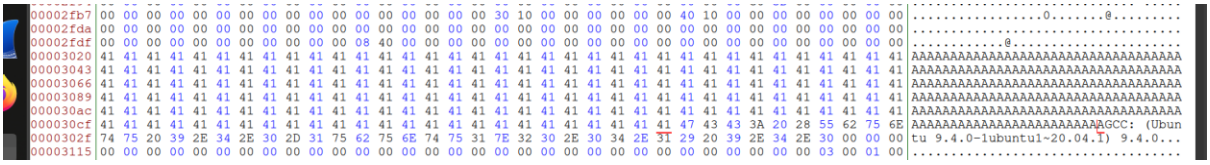
```
[11/10/22]seed@VM:~/.../lab6$ md5sum out1.bin
712f42d774e0cb06f84ef8458cb977da  out1.bin
[11/10/22]seed@VM:~/.../lab6$ md5sum out2.bin
712f42d774e0cb06f84ef8458cb977da  out2.bin
[11/10/22]seed@VM:~/.../lab6$ tail -c 128 out1.bin > P
[11/10/22]seed@VM:~/.../lab6$ tail -c 128 out2.bin > Q
[11/10/22]seed@VM:~/.../lab6$ md5sum P
369ce024587dd962a54d212118c1b660  P
[11/10/22]seed@VM:~/.../lab6$ md5sum Q
3db46e1289986ad03586c6fc89825715  Q
[11/10/22]seed@VM:~/.../lab6$ python3 -c "print('114514'*10,end='')" > suffix
[11/10/22]seed@VM:~/.../lab6$ cat out1.bin suffix > f1
[11/10/22]seed@VM:~/.../lab6$ cat out2.bin suffix > f2
[11/10/22]seed@VM:~/.../lab6$ md5sum f1
4f408cf4102243c4209d7c577d797ba9  f1
[11/10/22]seed@VM:~/.../lab6$ md5sum f2
4f408cf4102243c4209d7c577d797ba9  f2
[11/10/22]seed@VM:~/.../lab6$
```

Observation:

Here, we have shown that adding the same suffix to different prefixes will result in the same hash even though the hashes of the initial prefixes were different.

Task 3:

Screenshot:




```
[11/10/22]seed@VM:~/.../lab6$ python3
Python 3.8.10 (default, Jun 22 2022, 20:18:18)
[GCC 9.4.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> 0x3020
12320
>>> 0x30e7
12519
>>> 12320 % 64
32
>>> 12320 + 32
12352
>>> 12519 % 64
39
>>> 12519 - 39
12480
>>>
```

```
[11/10/22]seed@VM:~/.../lab6$ head -c 12352 task3 > prefix
[11/10/22]seed@VM:~/.../lab6$ tail -c +12480 task3 > suffix
[11/10/22]seed@VM:~/.../lab6$ md5collgen -p prefix -o P Q
MD5 collision generator v1.5
by Marc Stevens (http://www.win.tue.nl/hashclash/)

Using output filenames: 'P' and 'Q'
Using prefixfile: 'prefix'
Using initial value: 048c1b3c2f14dd1882f6a9cfcb2deed4

Generating first block: ............
Generating second block: W.
Running time: 8.75633 s
```

```
[11/10/22]seed@VM:~/.../lab6$ cat P suffix > arr1.out
[11/10/22]seed@VM:~/.../lab6$ cat Q suffix > arr2.out
[11/10/22]seed@VM:~/.../lab6$ sudo chmod +x arr1.out arr2.out
[11/10/22]seed@VM:~/.../lab6$ ./arr1.out > f1
[11/10/22]seed@VM:~/.../lab6$ ./arr2.out > f2
[11/10/22]seed@VM:~/.../lab6$ diff f1 f2
1c1
< 4141414141414141414141414141414141414141414141414141414141414141418c595cf4b27771c27fe5b8a49684fa5e712ddfceeed6ed1
9367c29698599d884e3f6bfda4e388cc72b45fbf49d8346279233208f4c4c2aaf413d4faf5d17a7847b61d1c65e8ad88a24b3b2e75c8da219
d7f258eb3442d0fdaa96501bedced59736e7787226e95e5b499aa3888dbe188191d152c723e4e8e2f0d8c5d542a7414141414141414141414
1414141414141414141414141414141414141414141414141414141414141414141
---
> 4141414141414141414141414141414141414141414141414141414141414141418c595cf4b27771c27fe5b8a49684fa5e712d5fceeed6ed1
9367c29698599d884e3f6bfda4e388cc72b45fbf41d8446279233208f4c4c2aaf413dcfaf5d17a7847b61d1c65e8ad88a24b3b2e75c8da219
d7fa58eb3442d0fdaa96501bedced59736e7787226e95e5b499aa388dbe188191d152c723e4e8e2f058c5d542a7414141414141414141414
1414141414141414141414141414141414141414141414141414141414141414141
[11/10/22]seed@VM:~/.../lab6$ md5sum arr1.out
0ae35ce6065b52fb9129226c3ed688a3  arr1.out
[11/10/22]seed@VM:~/.../lab6$ md5sum arr2.out
0ae35ce6065b52fb9129226c3ed688a3  arr2.out
[11/10/22]seed@VM:~/.../lab6$
```

Observation:

We have generated different values for the output but the same md5sum.

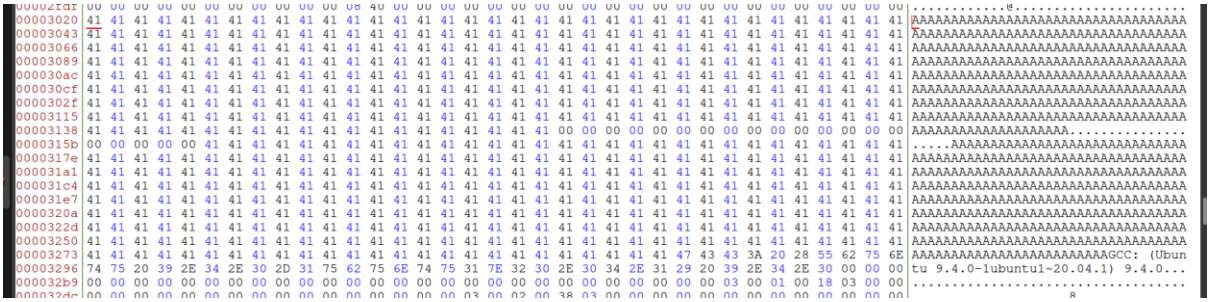
Task 4:

Screenshot:

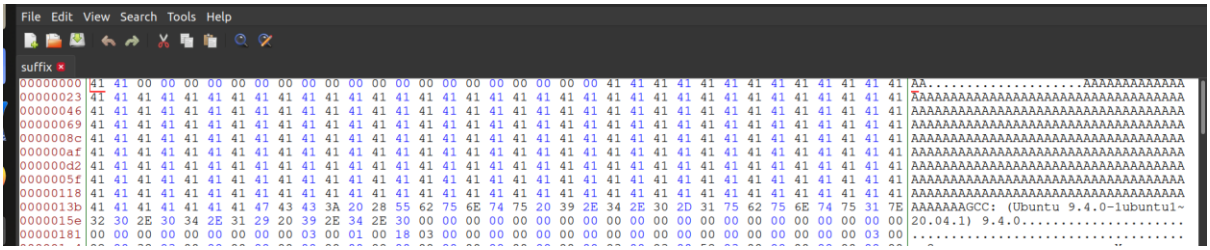
```
[11/10/22]seed@VM:~/.../lab6$ ls
Labsetup  task3  task3.c
[11/10/22]seed@VM:~/.../lab6$ vim task4.c
[11/10/22]seed@VM:~/.../lab6$ gcc task4.c -o task4
[11/10/22]seed@VM:~/.../lab6$ ./task4
Benign
[11/10/22]seed@VM:~/.../lab6$
```

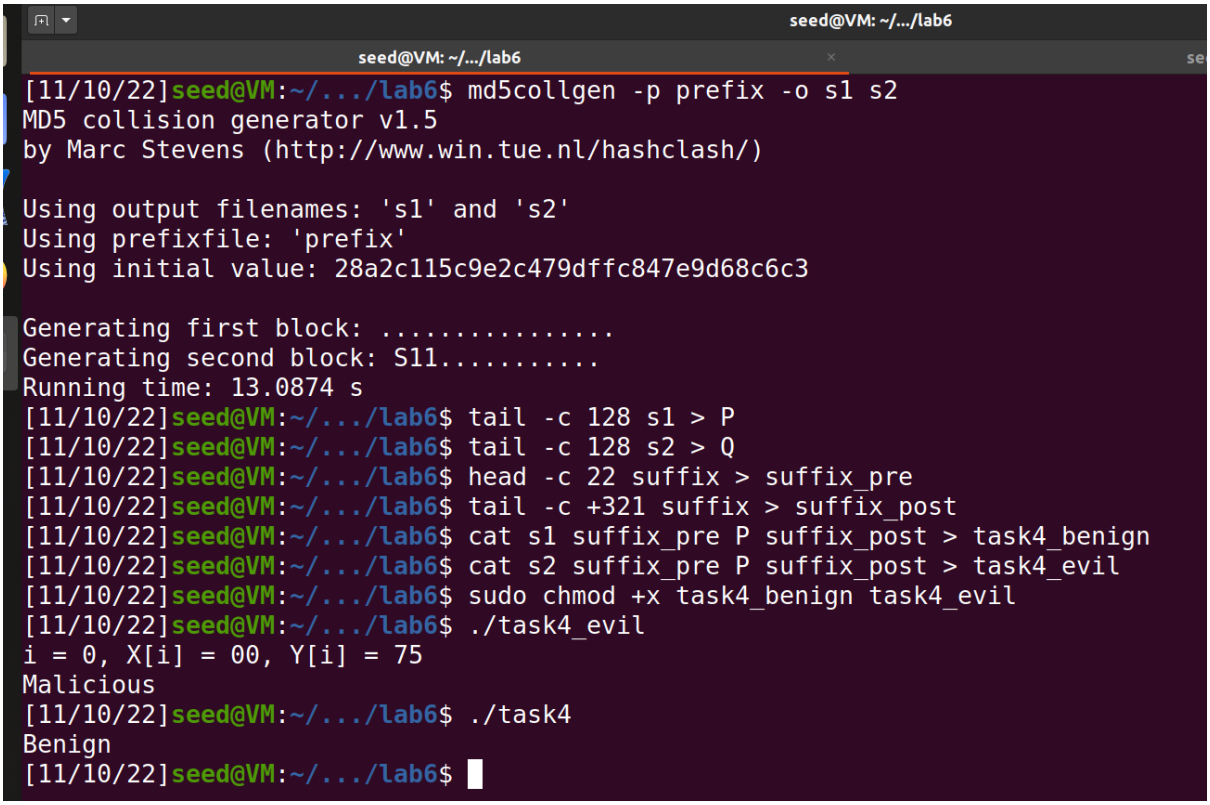
```
seed@VM: ~/.../lab6                                    seed@VM: ~/.../lab6
[11/10/22]seed@VM:~/.../lab6$ bless task4
Gtk-Message: 11:57:05.345: Failed to load module "canberra-gtk-module"
Could not find a part of the path '/home/seed/.config/bless/plugins'.
Could not find a part of the path '/home/seed/.config/bless/plugins'.
Could not find a part of the path '/home/seed/.config/bless/plugins'.
Could not find file "/home/seed/.config/bless/export_patterns"
Could not find file "/home/seed/.config/bless/history.xml"
^C
[11/10/22]seed@VM:~/.../lab6$ head -c 12320 task4 > prefix
[11/10/22]seed@VM:~/.../lab6$ tail -c +12619 task4 > suffix
[11/10/22]seed@VM:~/.../lab6$ bless suffix
Gtk-Message: 12:04:52.082: 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"
Could not find file "/home/seed/.config/bless/history.xml"
^C
```

Bless task4:



Bless suffix:





Observation:

Since we find a way to generate the same md5 hash value we can get the malicious software authorized.