APPLIED CRYPTOGRAPHY

NAME: VISHWAS M

SRN: PES2UG20CS390

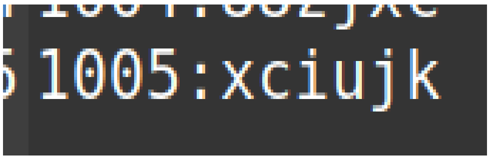
SEC: F

LAB: Hash Length Extension Lab

Task1: Send Request to List Files

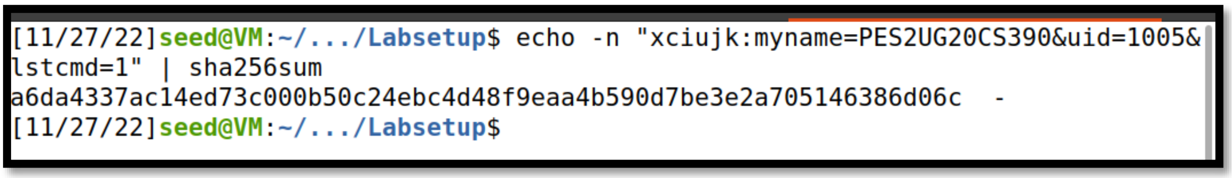
1)

Step1: Finding the uid



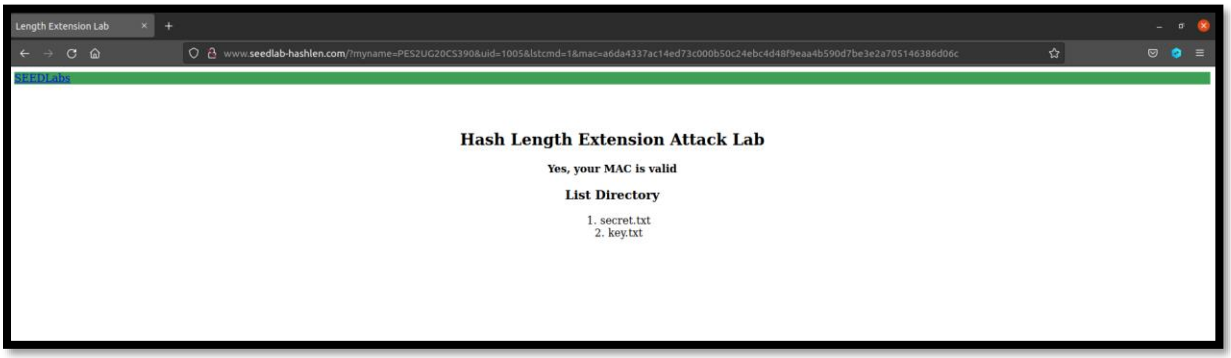
```
1005:xciujk
```

Step2: Calculating MAC



```
[11/27/22]seed@VM:~/.../Labsetup$ echo -n "xciujk:myname=PES2UG20CS390&uid=1005&
lstcmd=1" | sha256sum
a6da4337ac14ed73c000b50c24ebc4d48f9eaa4b590d7be3e2a705146386d06c  -
[11/27/22]seed@VM:~/.../Labsetup$
```

Step3: Sending the Request



```
www.seedlab-hashlen.com/?myname=PES2UG20CS390&uid=1005&lstcmd=1&mac=a6da4337ac14ed73c000b50c24ebc4d48f9eaa4b590d7be3e2a705146386d06c
```

SEEDLabs

Hash Length Extension Attack Lab

Yes, your MAC is valid

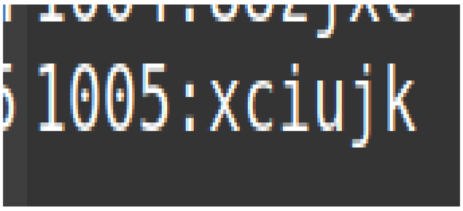
List Directory

1. secret.txt
2. key.txt

We are just finding the directory list from the above steps.

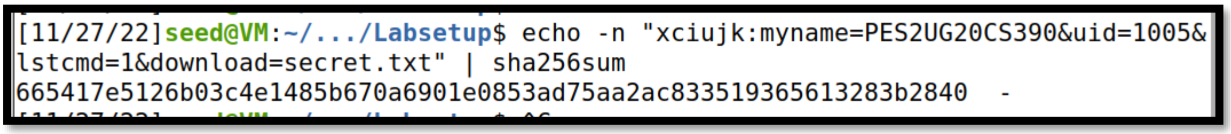
2)

Step1: Finding uid



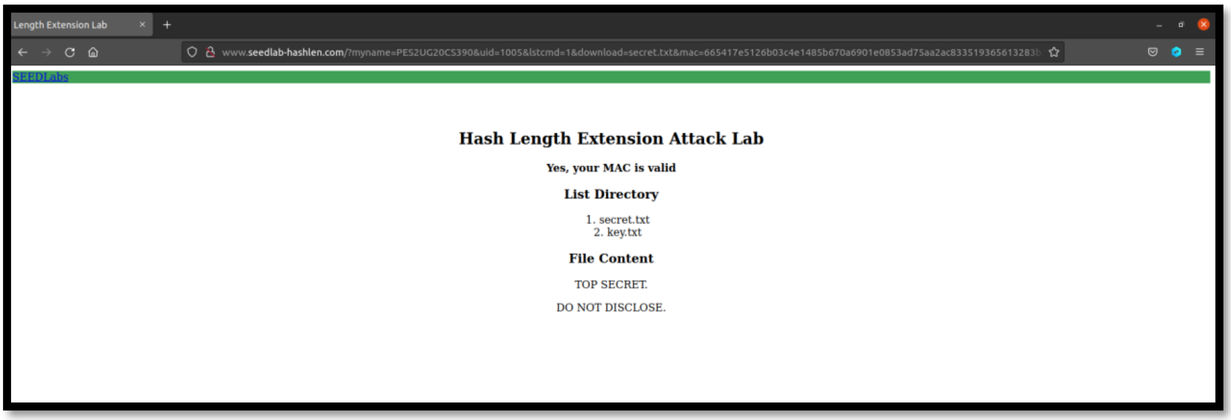
```
1005:xciujk
```

Step2: Calculating MAC



```
[11/27/22]seed@VM:~/.../Labsetup$ echo -n "xciujk:myname=PES2UG20CS390&uid=1005&
lstcmd=1&download=secret.txt" | sha256sum
665417e5126b03c4e1485b670a6901e0853ad75aa2ac833519365613283b2840  -
```

Step3: Sending the Request



www.seedlab-hashlen.com/?myname=PES2UG20CS390&uid=1005&lstcmd=1&download=secret.txt&mac=665417e5126b03c4e1485b670a6901e0853ad75aa2ac833519365613283b

SEEDLabs

Hash Length Extension Attack Lab

Yes, your MAC is valid

List Directory

1. secret.txt
2. key.txt

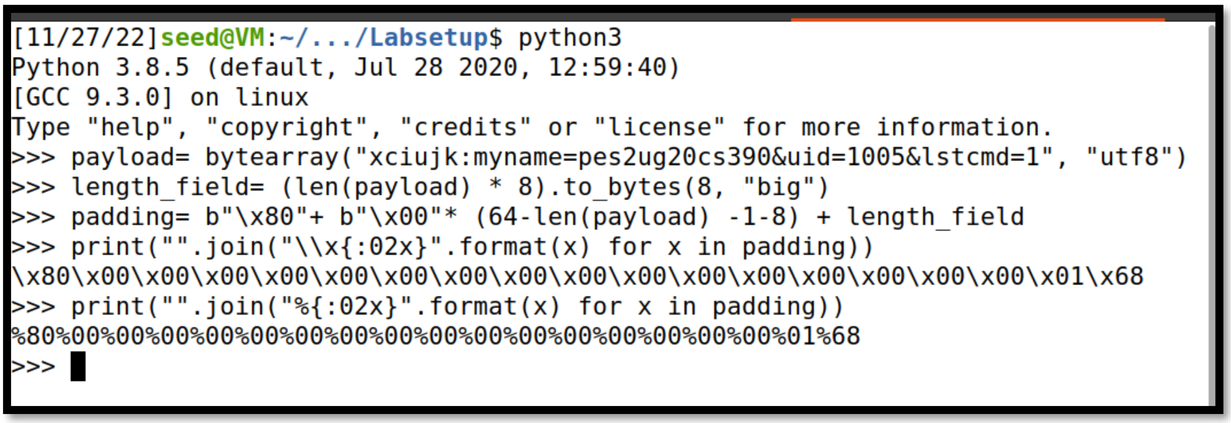
File Content

TOP SECRET.

DO NOT DISCLOSE.

We are finding the contents from secret.text which is displayed in the above screenshots.

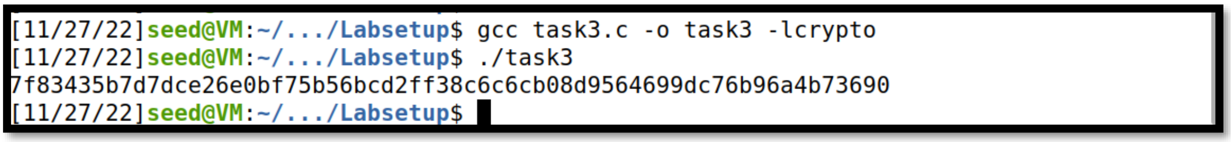
Task 2: Create Padding



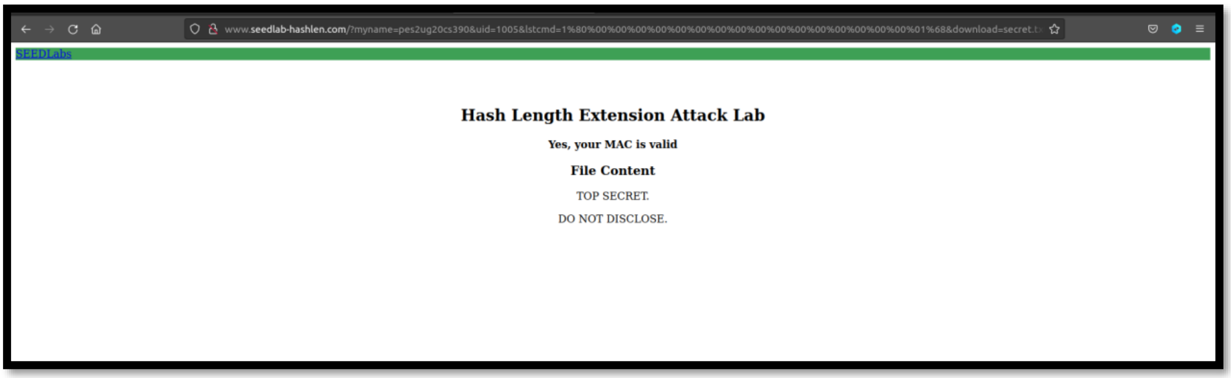
```
[11/27/22]seed@VM:~/.../Labsetup$ python3
Python 3.8.5 (default, Jul 28 2020, 12:59:40)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>> payload= bytearray("xciujk:myname=pes2ug20cs390&uid=1005&lstcmd=1", "utf8")
>>> length_field= (len(payload) * 8).to_bytes(8, "big")
>>> padding= b"\x80"+ b"\x00"* (64-len(payload) -1-8) + length_field
>>> print("".join("\\x{:02x}".format(x) for x in padding))
\x80\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x01\x68
>>> print("".join("%{:02x}".format(x) for x in padding))
%80%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%01%68
>>> 
```

We can observe the URL and hash padding.

Task 3: The Length Extension Attack



```
[11/27/22]seed@VM:~/.../Labsetup$ gcc task3.c -o task3 -lcrypto
[11/27/22]seed@VM:~/.../Labsetup$ ./task3
7f83435b7d7dce26e0bf75b56bcd2ff38c6c6cb08d9564699dc76b96a4b73690
[11/27/22]seed@VM:~/.../Labsetup$ 
```



www.seedlab-hashlen.com/?myname=pes2ug20cs390&uid=1005&lstcmd=1%80%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%01%68&download=secret.tx

SEEDLabs

Hash Length Extension Attack Lab

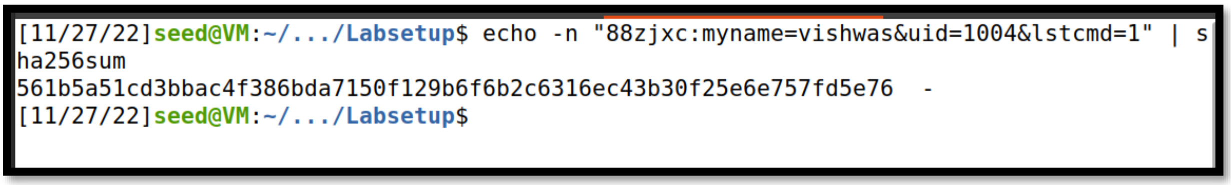
Yes, your MAC is valid

File Content

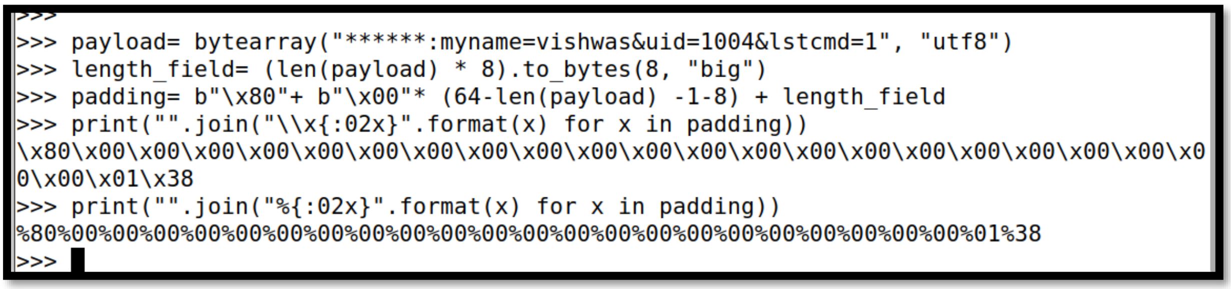
TOP SECRET.

DO NOT DISCLOSE.

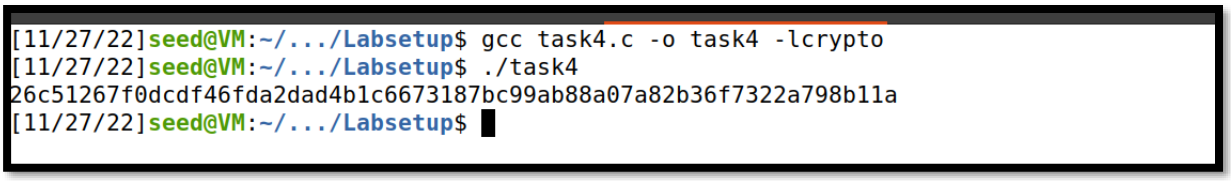
Without key:



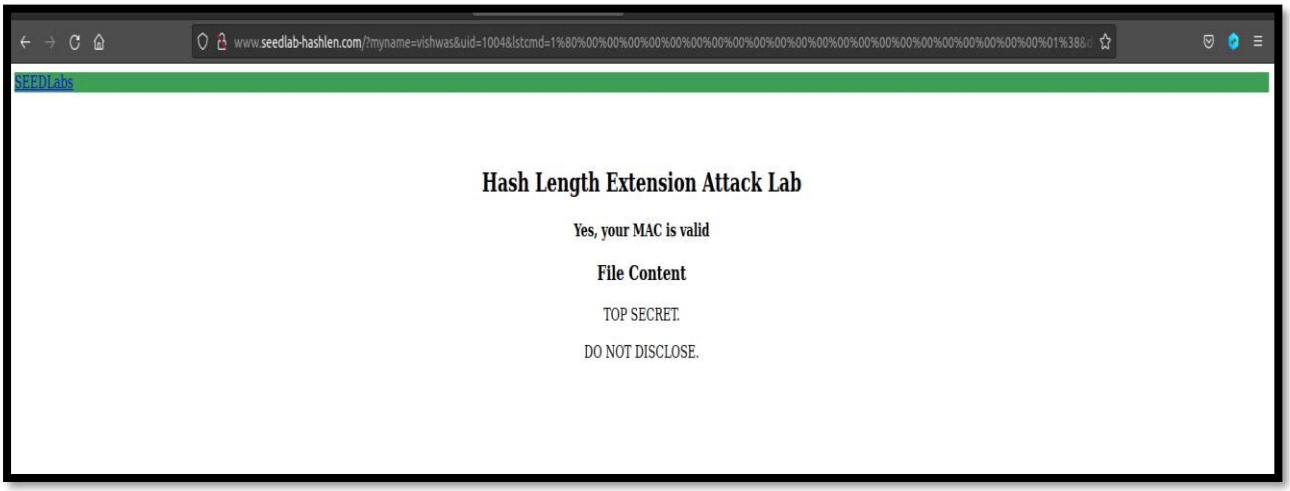
```
[11/27/22]seed@VM:~/.../Labsetup$ echo -n "88zjxc:myname=vishwas&uid=1004&lstcmd=1" | s
ha256sum
561b5a51cd3bbac4f386bda7150f129b6f6b2c6316ec43b30f25e6e757fd5e76  -
[11/27/22]seed@VM:~/.../Labsetup$
```



```
>>>
>>> payload= bytearray("******:myname=vishwas&uid=1004&lstcmd=1", "utf8")
>>> length_field= (len(payload) * 8).to_bytes(8, "big")
>>> padding= b"\x80"+ b"\x00"* (64-len(payload) -1-8) + length_field
>>> print("".join("\\x{:02x}".format(x) for x in padding))
\x80\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x0
0\x00\x01\x38
>>> print("".join("%{:02x}".format(x) for x in padding))
%80%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%01%38
>>> 
```



```
[11/27/22]seed@VM:~/.../Labsetup$ gcc task4.c -o task4 -lcrypto
[11/27/22]seed@VM:~/.../Labsetup$ ./task4
26c51267f0dcdf46fda2dad4b1c6673187bc99ab88a07a82b36f7322a798b11a
[11/27/22]seed@VM:~/.../Labsetup$ 
```



www.seedlab-hashlen.com/?myname=vishwas&uid=1004&lstcmd=1%80%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%01%38&d

SEEDLabs

Hash Length Extension Attack Lab

Yes, your MAC is valid

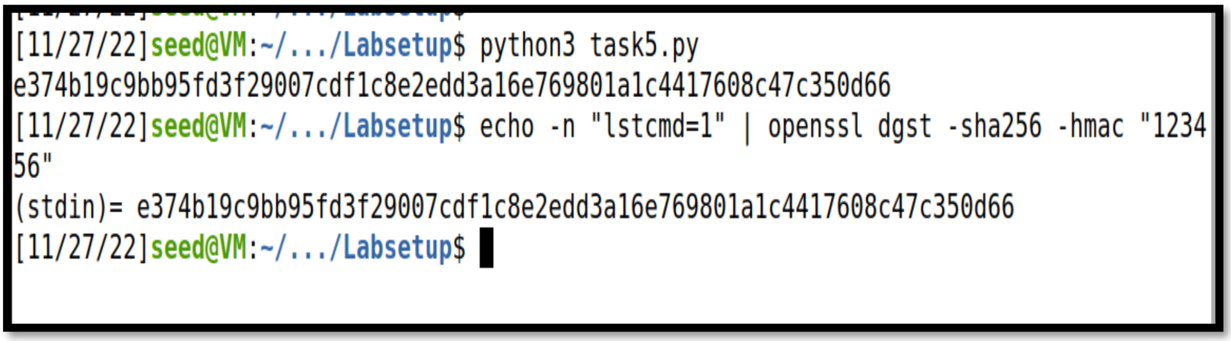
File Content

TOP SECRET.

DO NOT DISCLOSE.

The attack is successful and we can see the contents of the file secret.txt even though it is not a valid MAC address. We do both the attacks with the key and without it.

Task4:



```
[11/27/22]seed@VM:~/.../Labsetup$ python3 task5.py
e374b19c9bb95fd3f29007cdf1c8e2edd3a16e769801a1c4417608c47c350d66
[11/27/22]seed@VM:~/.../Labsetup$ echo -n "lstcmd=1" | openssl dgst -sha256 -hmac "1234
56"
(stdin)= e374b19c9bb95fd3f29007cdf1c8e2edd3a16e769801a1c4417608c47c350d66
[11/27/22]seed@VM:~/.../Labsetup$ 
```

We use hmac to avoid the length extension attack as the mac generated will not be same for the extended length and the original address without padding and will thus fail to authenticate the address.