哈尔滨工业大学（深圳）

《密码学基础》实验报告

Hash 长度扩展攻击实验

学　院: 计算机科学与技术

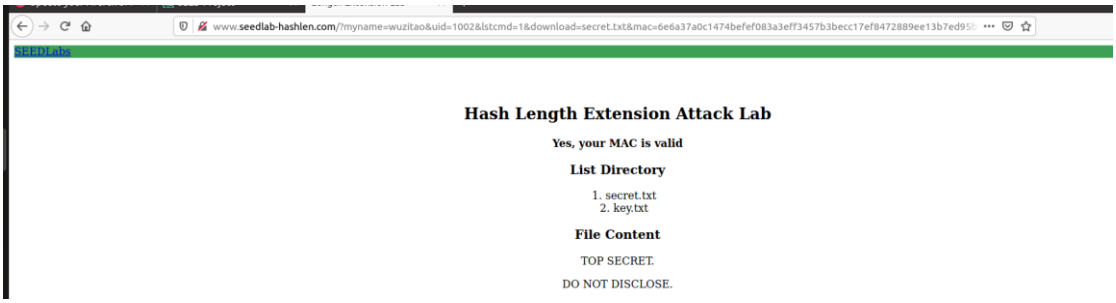
姓　名: 吴梓滔

学　号: 220110430

专　业: 计算机科学与技术

日　期: 2024-10-23

1、 请发送一个`download`命令到服务区，myname 的信息修改为你自己的姓名拼音，并且记录你得到的响应内容（截图显示）。



消息 myname=wuzitao&uid=1002&lstcmd=1&download=secret.txt 创建的 mac 为 6e6a37a0c1474befef083a3eff3457b3becc17ef8472889ee13b7ed95b15afc0。发送命令后，得到如图所示的相应内容。

2、 为消息 <key>:myname=<name>&uid=<uid>&lstcmd=1 创建对应 padding，其中`<key>`和`<uid>`的实际内容应该从`LabHome/key.txt`文件中得到，myname 依然用你自己的姓名。

结果类似这样，红色部分可以参考代码换成 AAAAAA，不影响填充的内容：

```
123456:myname=SEEDManual&uid=1001&lstcmd=1
%80%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%01%50
```

```
[10/21/24]seed@VM:~/Crypto_Hash_Extension$ python3 ./compute_padding.py
AAAAAA:myname=wuzitao&uid=1002&lstcmd=1%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
```

通过 compute_padding.py 为消息 myname=wuzitao&uid=1002&lstcmd=1 生成了 padding 值，并且 padding 值和 key 无关。Padding 为：

%80%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%00%01%38

3、 为下面的请求生成一个有效的 MAC，其中`<key>`和`<uid>`的实际内容应该从`LabHome/key.txt`文件中得到，name 就是自己的姓名拼音。

http://www.seedlab-hashlen.com/?myname=<name>&uid=<uid>&lstcmd=1&mac=<mac>

```
[10/21/24]seed@VM:~/Crypto_Hash_Extension$ echo -n "983abe:myname=wuzitao&uid=1002&lstcmd=1" | sha256sum
eaedc6bd9953f237f7c9565bf4c91138360e0edd44d4f3b176df21b0d9ab1086  -
```

为消息 myname=wuzitao&uid=1002&lstcmd=1 生成了一个有效 mac 值

eaedc6bd9953f237f7c9565bf4c91138360e0edd44d4f3b176df21b0d9ab1086

4、 发送构造好的新请求到服务器，padding 是上面获取到的信息，记录收到的服务器响应并截图。

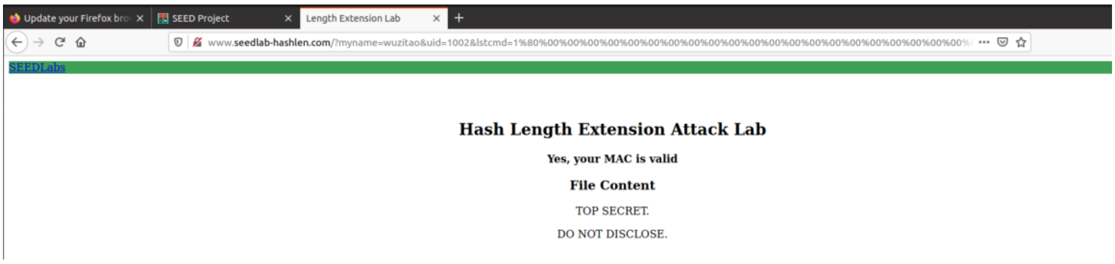
http://www.seedlab-hashlen.com/?myname=<name>&uid=<uid>&lstcmd=1<padding>&download=secret.txt&mac=<new-mac>

```
[10/21/24]seed@VM:~/Crypto_Hash_Extension$ gcc url_length_extension.c -lcrypto
[10/21/24]seed@VM:~/Crypto_Hash_Extension$ a.out
8faa537054de02e34895b83af39f21ce5a7ec0c09195c259b172f745d01115bf
```

将第 3 步中构建的 mac 值设置到 url_length_extension.c 中，运行生成一个新的 mac 值。用该 mac 值和 padding 构造出一个新的 url 请求如下：

http://www.seedlab-hashlen.com/?myname=wuzitao&uid=1002&lstcmd=1%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&download=secret.txt&mac=8faa537054de02e34895b83af39f21ce5a7ec0c09195c259b172f745d01115bf

这个包含对 secret.txt 的下载命令。发送到服务区后，成功获取到 secret.txt 的内容。



5、 用 HMAC 算法修改代码后，记录使用长度攻击的结果，根据收到的服务器响应进行截图。

```
[10/21/24]seed@VM:~/Crypto_Hash_Extension$ python3 cal_hmac.py
55e0ebe300415b12cd89b5f3bc8e849051937cb58e93c42ffbcbc795d5bd7502
[10/21/24]seed@VM:~/Crypto_Hash_Extension$ vi cal_hmac.py
[10/21/24]seed@VM:~/Crypto_Hash_Extension$ python3 cal_hmac.py
78d9500e4f99076db5ef2a749b3fb7a1f39a634f943f920882649f2fe85fa64c
```

将服务器端验证和用户生成 mac 的代码均改为 HMAC 后，重复上述 4 个步骤。

（1）生成有效 mac 下载 secret.txt

myname=wuzitao&uid=1002&lstcmd=1&download=secret.txt 的 mac 值为

78d9500e4f99076db5ef2a749b3fb7a1f39a634f943f920882649f2fe85fa64c

如图所示，能够访问到 secret.txt 的内容。

(2)生成另一个消息的 padding 和 mac

为消息 myname=wuzitao&uid=1002&lstcmd=1 生成的 mac 值为

55e0ebe300415b12cd89b5f3bc8e849051937cb58e93c42ffbcbc795d5bd750

2，生成的 padding

为%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。

```
// The MAC for the valid URL
int a[8] = {  0x55e0ebe3, 0x00415b12, 0xcd89b5f3, 0xbc8e8490,
              0x51937cb5, 0x8e93c42f, 0xfbcbc795, 0xd5bd7502 };
```

```
[10/21/24]seed@VM:~/Crypto_Hash_Extension$ gcc url_length_extension.c -lcrypto
[10/21/24]seed@VM:~/Crypto_Hash_Extension$ a.out
17cc5647cff5d21ca34d6eb8130d1beefba861adb62a2c177e462b2dd124f660
```

如图，使用填充前已经获得的 mac 生成新 mac。新 mac 为：

17cc5647cff5d21ca34d6eb8130d1beefba861adb62a2c177e462b2dd124f66

0

（3）构造新请求尝试攻击

根据上面生成的新 mac 和 padding，构造出 url 如下：

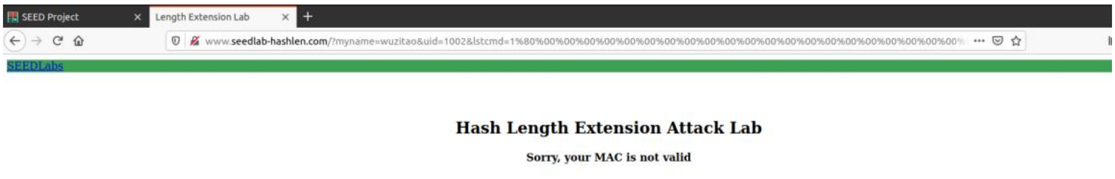
http://www.seedlab-hashlen.com/?myname=wuzitao&uid=1002&lstcmd=1%

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&download=secret.txt&mac=17cc5647cff5d21ca34d6eb8130d

1beefba861adb62a2c177e462b2dd124f660

发送该请求得到下图的响应。



如图，访问失败，说明更换成 HMAC 算法后，长度扩展攻击失败。