Lab #2 - MAC forgery attacks

Léo Gabaix

Francesco Mosanghini

4.1 CBC-MAC concatenation attack

We developed a python script to recreate this attack.

Here are the steps to run the code:

```
# create virtual environment
$ python3 -m venv lab_env

# activate virtual environment
$ source lab_env/bin/activate

# install requirements
$ pip install -r requirements.txt

# run the script
$ python3 cbc_mac.py
Message 1: What about joining me tomorrow for dinner?
Message 2: Oops, Sorry, I just remember that I have a meeting very soon in the mornin
g.
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
hex string is too short, padding with zero bytes to length
[PRINT]: Tag2:
00000000: 6c96 f6a4 8e5d ce96 af18 9f84 7824 70ea  l....]......x$p.

[PRINT]: Forged:
00000000: 6c96 f6a4 8e5d ce96 af18 9f84 7824 70ea  l....]......x$p.

# exit virtual environment
$ deactivate

# delete virtual environment
$ rm -r lab_env/
```

4.2 One-pass HMAC length extension attack

To easily manipulate the Hash function context object we developed the second attack using C.

Here are the steps to run the code:

```
# compile code
$ gcc lengthExtension.c -o OnePassHMAC -lcrypto

# run binary file
$ ./OnePassHMAC
Forged message: What about joining me tomorrow for dinner?Appended data!
Forged MAC: 907291629dca3fd51720137559d047f6
Valid. MAC: 907291629dca3fd51720137559d047f6
```