CS426 Computer Security: Lab 2

Dave (Jing) Tian

March 2, 2021

1 Deadline

Tue Mar 9, AoE. (You know the late penalty.)

2 Goal

Padding oracle attack.

3 Description

In this lab, you will find a file named *oracle.py*. This file is the padding oracle that we will query to launch the padding oracle attack. For instance:

```
$./oracle.py -e 1111222233334444daveti
Ciphertxt: b'
    e2f609b0002fd83715d30f1aab696ea22364b67a97db1f4e96f14c83b48e6195
    ', iv: b'1234567812345678'

$./oracle.py -d
    e2f609b0002fd83715d30f1aab696ea22364b67a97db1f4e96f14c83b48e6195

Plaintxt: b'1111222233334444daveti'
```

Go ahead to dig into the source file and see how PKCS#7 is implemented and how to play with the oracle in general.

Once you are familiar with the oracle, you need to implement the padding oracle attack in a new file called *attack.py*, which should support at least "-a"

option followed by a given ciphertxt. For instance, if you call *attack.py* to break the ciphertxt above, it should look like below:

```
$./attack.py -a
    e2f609b0002fd83715d30f1aab696ea22364b67a97db1f4e96f14c83b48e6195

6461766574690a0a0a0a0a0a0a0a0a0a
```

The output "6461766574690a0a0a0a0a0a0a0a0a0a" is in hex string format, which is essentially "daveti\n\n\n..." in ASCII format. Internally, your *attack.py* should "query" the padding oracle *oracle.py* to break the ciphertxt. Since we are running all the scripts locally, the "query" can be emulated by calling *oracle.py* within *attack.py*, e.g., using subprocess in Python.

You will also find a file named *test.sh*, where we provide the basic automation for grading. If you are able to run this script using your own *attack.py* without any troubles, it is likely that the final grading will be fine too. If you run the script with a reasonable *attack.py* implementation, the output should look like below:

```
$./test.sh
Success.
Success.
```

Note that we will use our own *oracle.py* and *test.sh* in the final grading. They look like the files provided a lot with minor changes, e.g., secret key, IV, ciphertxt, etc. This means any changes locally within these files will not work in our grading environment.

4 Deliverable

attack.py file supporting at least "-a" option followed by a ciphertxt. The output of the file should be the recovered plaintxt in hex string format, nothing else. Please check the example above.

5 Grading

You will need to crack all the ciphertxt provided in grading to get a 100 points. Any failure will lead to zero points.

6 Others

I recommend using Python3 to reduce the hassle of import dependency in the assignment. Even though the padding oracle accepts IV as an optional argument, which helps break the first cipher block if needed, we will not consider this scenario in this assignment.