

Assignment 3

*Lecturer: Reza Shokri**Student: Name Number*

Instructions

- All submissions should be in ZIP and should be made via LumiNUS. The report should be in PDF. You are strongly encouraged to use LaTeX (use the provided template).
- We provide the submission template. Your submission should be in the same folder structure, replacing A01xxxx by your student number.
- Your file name should be your student number (e.g., A01xxx.zip)
- Use font size 12.
- We will ignore all the submissions that violate any of the above.
- You have 2 weeks to submit your solutions.
- Report page limit: ≤ 2
- **You are not allowed to share the content of the assignment with others (e.g., release them on the Internet).**
- **No collaboration** is permitted on this assignment. Any cheating (e.g., copying from any source, or permitting your work to be copied) will result in a failing grade. The assignments are provided to help you learning by solving problems. What's the point of obtaining the solutions from elsewhere? Your grade won't help you securing your code.

Traffic Analysis

The objective of this assignment is to show the vulnerability of the HTTPS traffic to traffic analysis attacks. Prior to the attack, in the profiling phase, the attacker connects to a set of URLs and captures the communicated packets between his machine and the server hosting the URL. The attacker performs the profiling attack multiple times, in order to obtain many samples of the HTTPS traffic to target webpages, for generating a fingerprint for each webpage. In the attack phase, where the attacker observes the connection of the victim to an anonymous webpage (through a proxy server, or anonymity network such as Tor), he tries to match his observation with the fingerprints.

Profiling phase

You receive the processed captured packets of 35 URLs, to build the fingerprints. The packet traces for each URL are saved in a file, with the same and persistent file ID across the 8 measurements. The packet trace files contain the time of arrival for each packet, its size in Bytes, and direction (in: towards the client, out: towards the server). You can obtain the profiles in `traces.zip`. (You can also find the profiles in the virtual machine we provide.) Figure 1 is an example of a trace file.

Attack phase

You will receive the captured packets, corresponding to the attacker's observation of anonymous webpages. The set of anonymous URLs is the same as the profiled URLs, but of course, their identities are randomly permuted. This is known as the closed-world setting. Your algorithm will be tested by the observations of two different attackers `observation1` and `observation2`. The observations from one attacker are in the same folder which contains 35 files, as shown in Figure 3.

```

00:00:00.000000 0 in
00:00:00.000056 0 out
00:00:00.000897 201 out
00:00:00.134478 0 in
00:00:00.134523 0 out
00:00:00.135007 201 out
00:00:00.243595 0 in
00:00:00.262097 1448 in
00:00:00.262118 0 out
...

```

Figure 1: *An example of a trace file.*

```

observation1/
    1-anon
    2-anon
    3-anon
    4-anon
    ...
    35-anon

```

Figure 2: *An example of attacker's observation folder.*

The expected solution of your algorithm is 1 file for 2 columns (one for each attack observation), with 35 lines (for each URL in the profiles, where the line number represents the URL id) and 1 column (for this attacker observation), separated with a space character. In each column, please provide the anonymous id of a URL corresponding to its id in the profiles. Please do NOT put any extra character/line, as your solution file will be corrected automatically. Figure 3 shows an example of a solution, where the first line shows the URL with anonymous id 2 in `observation1`

and anonymous id 34 in `observation2` are associated with the profiles with id 1:

```
2 34
7 1
12 8
9 11
30 2
15 28
...
```

Figure 3: *An example of a solution file.*

Code requirement

Please provide a `test.sh` bash script which takes the attacker's observations folders (`observation1`, `observation2`) as the input, and produces the result file. We will run your `test.sh` by command:

```
$ ./test observation1 observation2
```

Your bash script should automatically produce `result.txt` as figure 3. The assignment is considered as successfully submitted only if the `test.sh` can produce the result file with the correct format.

You can only use Python and the libraries we provided (See details in the virtual machine on MS Teams: File – > Assignment – > Assignment-3.ova). If you need other libraries, please contact TAs to add the library and update the virtual machine image accordingly. The username for the virtual machine is `student` and password is 123456.

NOTE: Your program will be tested on the virtual machine we provide.

All code submissions that fail to run on this virtual machine will not be graded.

Submission

Please submit your code files and bash script along with 2 page summary of your attack algorithm (A01xxxx.pdf). Your submission folder should be in the same folder structure as we provided.