Youtube Fake View Script

Network Security - Carnegie Mellon University

Team: Buckstar

All code and documents are for academic purpose only.

A. Preface: How Youtube View Counts

Refer: Quora-How does YouTube calculate its views?

- Youtube has distributed log servers, the view count record is stored in one centralized server.
- When one Youtube view count goes over 300, the views will be verified (usually, it get stucked for few hours or 1 day). More, when the number 300 was decided by the designers, the programmer made the elementary careless mistake of using the age old ViewCount <= 300 instead of the ViewCount < 300 in their logic
- If a video is viewed in its entirety by someone who clicked on it, it is counted as one view.
- YouTube also considers views from the same IP in breaks of 6 to 8 hours. So one person viewing the same video repeatedly would only generate 3 to 5 views a day, after views cross 300.

B. Analysis Youtube Player Mechanism

1. HTML

In every Youtube page, it loads blank HTML for the player at first.

```
<div id="player-api" class="player-width player-height off-screen-target player-api"
    tabIndex="-1">
</div>
```

2. Youtube ytplayer Javascript Code

Then, the Javascript code will load the Youtube player (Adobe Flash Player).

```
<script>
var ytplayer = ytplayer || {};
ytplayer.config = {"min_version":"8.0.0",
                   "url":"https:\/\/s.ytimg.com\/yts\/swfbin\/player-vflenMUmo\/watch_as3.swf"
...
</script>
```

3. Loads Adobe Flash (SWF Player)

Here is how it looks like after the video is loaded. (The <embed> element is supported in all major browsers.)

```
<embed name="movie_player" id="movie_player" width="100%" height="100%" tabindex="0"
    type="application/x-shockwave-flash"
    src="https://s.ytimg.com/yts/swfbin/player-vflenMUmo/watch_as3.swf"
    allowscriptaccess="always" bgcolor="#000000" allowfullscreen="true" ...>
```

C. Script

In ordering to generate Youtube views, I've tried several approaches including using PhantomJS and SeleniumHQ. And, PhamtonJS won't work for increasing Youtube views since it's not supporting Flash Player; however, SeleniumHQ works nicely and is tested on Ubuntu Server.

1. PhamtonJS (Failed)

Run:

```
make run_phamtonjs
```

And here are the details about the files:

phantomjs/fake__click__no__js.js: opens a Youtube page without executing its javascript.

phantomjs/fake__click.js: opens a Youtube page then executes its javascript. **However, by looking to its screenshot, we can learn that phamtonjs is not supporting Adobe Flash Player.** Refer to: phantomjs doesn't support flash player.

2. Selenium HQ (Success)

Run:

```
cd seleniumhq && ./run.sh
```

opens Youtube page using Firefox driver. (**X11 is required**)

To start browser headlessly, we should apply Xvfb(virtual framebuffer X server for X Version 11) as background process. Run:

```
cd seleniumhq && make run_headless_server
```

Or, just run following command at root directory:

```
make run_selenium
```

D. Faking User Agent

Give different parameters for fake_click.py to specify User Agent or Target, try:

```
> ./fake_click.py -h
usage: Visit one website using Selenium [-h] [-u [USER_AGENT]] [-t [TARGET]]
                                        [-f [FILE]]

optional arguments:
  -h, --help            show this help message and exit
  -u [USER_AGENT], --user-agent [USER_AGENT]
                        specify a user agent for HTTP header
  -t [TARGET], --target [TARGET]
                        target URL
  -f [FILE], --file [FILE]
                        user agent list
```

for example:

```
> ./seleniumhq/fake_click.py -u "my custom user agent code" -t
    "https://www.youtube.com/watch?v=6_XjCeT6V1k"
```

Test

You can test the code by referring to our own local server by running:

```
cd seleniumhq && ./test_listener.py
```

then:

```
cd seleniumhq && ./fake_click.py
```

you should get following in your server console:

```
Connected by ('127.0.0.1', 51108)
GET / HTTP/1.1
Host: localhost:50007
User-Agent: some UA string
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

E. Experiments

1. Test a list of user agents from one IP (about 526 user agents)

- Command:
  cd seleniumhq && python ./fake_user.py -f user_agents.txt -t "http://www.youtube.com/...
- Result: We found that Youtube gives out different websites based on different user agents. Some of them get normal desktop versions, some get mobile versions.
- Note: output screenshots is stored under *seleniumhq/screenshots/*

2. Test a list of *Black Berry* user agents from one IP (about 117 user agents)

- Command:
  cd seleniumhq && python ./fake_user.py -f user_agents_blackberry.txt -t "http://www.youtube.com/...
- Result: All of the sites Youtube returned for these user agents were pure HTML files with a link to RTSP stream protocol.
- Note: output screenshots is stored under *seleniumhq/screenshots_blackberry/*

F. Future Work

- Look into RTSP protocol.
- Record traffic packets and see if they can be reproduced.

G. Code

All the code mentioned in this document is on this GitHub Repo.