패킷 입출력 파형을 통한 웹사이트 핑거프린팅

https://youtu.be/fygEhMhC4lA

서론

- 최근 다양한 종류의 웹사이트 핑거프린팅이 발달

- 맞춤 광고나 해킹 등에 사용

서론

- 기존에는 브라우저 쿠키 등을 통해 진행
- 최근에는 브라우저 암호화가 강화되어 다른 방법이 필요
- 본 논문에서는 짧은 시간 내에 캡처된 사용자 패킷으로부터의 파형을 통해 진행

관련 연구

- T Wang, I Goldberg, “Improved website fingerprinting on Tor”, WPES’13, 2013

- pcap 파일을 특정 형태의 문자열로 변환한 뒤 유사도를 통해 핑거프린팅 진행

관련 연구

- Meng Shen, Yiting Liu, Siqi Chen, Liehuang Zhu, Yuchao Zhang, “Webpage Fingerprinting Using Only Packet Length Information”, ICC 2019, 2019

- 패킷 길이 정보만을 이용하여 핑거프린팅 진행

데이터 수집

- OS: Ubuntu 18.04 LTS
- Capture Program: tcpdump 4.9.3 (libpcap 1.8.1)
- 수집 자동화를 위해 파이썬 프로그램 작성
- Python 3.6.9
- Selenium: 3.141.1
- Chrome: 86.0.4240.22 (secret mode)

데이터 수집

- 코드 상에서 tcpdump를 이용하는 sh파일을 실행시켜서 수집 진행
- 셀레니움을 통해 브라우저를 구동시키고, 추가로 터미널을 열어 수집 프로그램 실행

```python
#!/usr/bin/python

from time import sleep
from selenium import webdriver
import os
import json
import util

if __name__ == '__main__':

    with open('config.json') as json_file:
        json_data = json.load(json_file)

        for name in json_data['pcaps']:
            #if os.path.isdir('./pcaps/' + name): continue

            print('[[ ' + name + ' ]]')
            for i in range(util.number_of_collect):
                print( str(i+1) + '/' + str(util.number_of_collect) + '...' )

                browser = util.get_browser()

                os.system("sudo gnome-terminal -- /bin/sh -c 'python3 macro.py " + name + "'")
                browser.get('http://' + name)
                sleep(util.work_time)
                browser.quit()

            os.system('sudo chmod -R 777 pcaps/' + util.getReplaced(name))
            print('collecting done\n')
```

메인 프로그램

```python
#!/usr/bin/python

from time import sleep
from selenium import webdriver
from multiprocessing import Process
import os
import sys
import util

if __name__ == '__main__':

    os.system('sudo ./capture.sh ' + util.getReplaced(sys.argv[1]) + ' &')
    sleep(util.work_time)
```

수집 프로그램

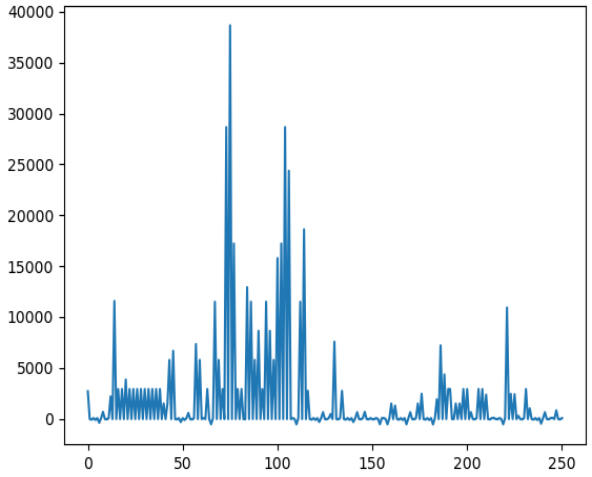
데이터 수집

- 구글 서비스 6개를 이용하여 진행

- 구글 검색
- 구글 지도
- 구글 Meet
- 구글 뉴스
- 구글 플레이
- 유튜브

데이터 수집

- 각 사이트를 5초씩 100번 접속하여 사이트별 100개, 전체 600개의 pcap 파일 수집
- 파형을 위해 pcap 파일을 그래프로 표현
- x축: 전체 입출력 길이를 0부터 나타낸 것
- y축: 각 구간에 대한 입출력 길이 (입력: 양수, 출력: 음수)



google.com의 파형

데이터 수집

- 테스트 데이터도 같은 방식으로 수집

- 사이트별 10개씩 전체 60개 수집

학습

- Tensorflow 2.2.0
- ReLU

epochs	loss	정확도 (x100)
0	1.7921119928359985	0.1666666716337204
5	0.9970917701721191	0.5400000214576721
10	0.20056971907615662	0.9044444561004639
15	0.09928524494171143	0.9288889169692993
20	0.12476327270269394	0.9644444584846497
25	0.09282336384057999	0.9777777791023254
30	0.10417622327804565	0.9844444394111633
35	0.13557982444763184	0.9733333587646484
40	0.1030362918972969	0.9688888788223267
45	0.1155698150396347	0.9755555391311646

에포크별 로스와 정확도

테스트 및 결과

- 테스트 결과 평균 66.7%의 정확성 기록
- 구글 검색과 구글 뉴스를 제외하면 높은 수준의 정탐율 기록

웹사이트	정확성
google.com	0/10 (0%)
maps.google.com	5/10 (50%)
meet.google.com	10/10 (100%)
news.google.com	9/10 (90%)
play.google.com	10/10 (100%)
youtube.com	6/10 (60%)

테스트 결과

웹사이트	탐지 (정탐률)
google.com	4 (0/4, 0%)
maps.google.com	5 (5/5, 100%)
meet.google.com	10 (10/10, 100%)
news.google.com	24 (10/24, 41.7%)
play.google.com	11 (10/11, 90%)
youtube.com	6 (6/6, 100%)

탐지 결과

결론 및 향후 연구

- 본 논문에서의 방식을 다양한 서비스에 이용하는 것을 기대
- 파형이 비슷한 웹사이트들에 대한 추가적인 분석
- 데이터 수집 조건의 다변화

참고문헌

- A Hintz, “Fingerprinting websites using traffic analysis”, International Workshop on Privacy Enhancing Technologies, 2002
- T Wang, I Goldberg, “Improved website fingerprinting on Tor”, WPES’13, 2013
- Meng Shen, Yiting Liu, Siqi Chen, Liehuang Zhu, Yuchao Zhang, “Webpage Fingerprinting Using Only Packet Length Information”, ICC 2019, 2019

Q & A