# JODATUT2019 Datan kerääminen

Nimi: Petri Salminen
Opnum: 243840
Email: petri.salminen@tuni.fi

Aloin miettimään ongelmalähtöisesti harjoitustyöni aihetta. Päädyin siihen, että haluaisin automatisoida "outojen" ja "epäilyttävien" pyyntöjen löytämisen verkkopalvelimen lokidatasta. Saatan pystyä hyödyntämään tätä töissäni, joten turhaa työtä tuskin tulee tehtyä kumminkaan.

Näen kuitenkin liian monimutkaiseksi (lakisyistä) hyödyntää oman yritykseni dataa, joten piti lähteä etsimään dataa jostain muualta. Kagglesta onneksi löysin shawon10/web-log-dataset (https://www.kaggle.com/shawon10/web-log-dataset), joka on ilmeisesti aika pieni, mutta haluan vähintäänkin yrittää saada siitä jotain irti.

Asennetaan datasetti hyödyntäen kaggle APIa (https://github.com/Kaggle/kaggle-api). Asetuksista on ensin luotava token, jotta APIa voi hyödyntää. Se pitää laittaa (esim. minun linux-asennuksessani) polkuun `~/.kaggle/kaggle.json`. Oikeuksiksi on hyvä asettaa 600, jotta muut käyttäjät eivät pääse lukemaan tiedostoa.

Nyt voimme ladata datasetin kaggle API:n kautta:

In [3]:
```
!kaggle datasets download -p data --unzip shawon10/web-log-dataset
```

```
Downloading web-log-dataset.zip to data
  0%|                                               | 0.00/144k [00:00<?, ?B/s
]
100%|████████████████████████████████████████| 144k/144k [00:00<00:00, 1.90MB/s
]
```

Seuraavaksi varmistamme, että data on todellakin siellä, minne halusimme:

In [4]:
```
!ls ./data
```

```
weblog.csv  webLog.csv
```

Ilmeisesti datasetissä on kaksi merkillisesti nimettyä tiedostoa: `weblog.csv` ja `webLog.csv`. Tutkitaan hieman alustavasti, mikä niissä on homman nimi hyödyntämällä linuxista löytyviä komentoja.

In [18]:
```
!head -n3 data/*csv
```

```
==> data/webLog.csv <==



==> data/weblog.csv <==
```

Ilmeisesti ylläoleva ei näy pdf:ssä oikein. Alla tuloste sellaisenaan. Huoh

In [19]:
```
print('''==> data/webLog.csv <==
"10.128.2.1","[29/Nov/2017:06:58:55","GET /login.php HTTP/1.1","200"
"10.128.2.1","[29/Nov/2017:06:59:02","POST /process.php HTTP/1.1","302"
"10.128.2.1","[29/Nov/2017:06:59:03","GET /home.php HTTP/1.1","200"

==> data/weblog.csv <==
IP,Time,URL,Staus
10.128.2.1,[29/Nov/2017:06:58:55,GET /login.php HTTP/1.1,200
10.128.2.1,[29/Nov/2017:06:59:02,POST /process.php HTTP/1.1,302
''')
```

```
==> data/webLog.csv <==
"10.128.2.1","[29/Nov/2017:06:58:55","GET /login.php HTTP/1.1","200"
"10.128.2.1","[29/Nov/2017:06:59:02","POST /process.php HTTP/1.1","302"
"10.128.2.1","[29/Nov/2017:06:59:03","GET /home.php HTTP/1.1","200"

==> data/weblog.csv <==
IP,Time,URL,Staus
10.128.2.1,[29/Nov/2017:06:58:55,GET /login.php HTTP/1.1,200
10.128.2.1,[29/Nov/2017:06:59:02,POST /process.php HTTP/1.1,302
```

Haa! Ilmeisesti `webLog.csv` -tiedostossa merkkijonot ovat heittomerkkien sisällä ja `weblog.csv` -tiedostossa on myös header-rivi.

## Hyödylliset linkit

- https://github.com/Kaggle/kaggle-api (https://github.com/Kaggle/kaggle-api)

## Erityisen helpot tai vaikeat asiat

Erityisen helppoa itselleni oli datan tutkiminen komentoriviltä (tai tässä tilanteessa jupyterin välityksellä). Oli mukava huomata, että !-merkillä pystyy ajamaan komentoja kuin suoraan komentoriviltä.

Hankalin asia oli saada kaggle API toimimaan oikein. Ensin asensin sen väärään ympäristöön (venv) ja muuta hassua, mutta siitä selvittiin.

Datan löytäminen olisi saattanut olla vaikeaa, mutta minulla kävi tuuri, kun datasetti löytyi nopeasti.