ncsc.cyberforce.no

Nasjonalt cybersikkerhetssenter (NCSC)

DatEng

- [101] Statistikk UserAgents #1
- [102] Statistikk UserAgents #2
- [103] Prosentfordeling - netflow #1
- [104] parsing error
- [105] Visualisering #1

THHH

- [121] Trusseljakt på Windows
- [122] Kodeknekkeren
- [123] Trusseljakt i Linux
- [124] Domain dominance
- [125] Beep Beeeeep Beep Beeeeeep

TakAna

- [151] Trusseletterretning i det digitale rom
- [152] Sårbarhetsoversikt #1
- [153] Sårbarhetsoversikt #2
- [154] Rammeverk
- [155] Viktig faktor i en cyberopasjon

In [ ]:
```
import hashlib
def gen_resp(string:str=""):
    en=string.encode()
    hex_result = hashlib.md5(en)
    print("SNIFF{" + hex_result.hexdigest() + "}")
```

101 - Statistikk UserAgents #1

- file: "uaiai.txt"

Kan du bidra med å finne antall unike UserAgents i dette datasettet, i perioden fra og med søndag, 29. august, 2021 7:27:03 PM (UTC) og til og med fredag, 29. oktober, 2021 9:48:39 AM (UTC)?

In [ ]:
```
from datetime import datetime
import time

ts_start    = time.mktime( datetime(2021,8,29,17,27,3).timetuple() )
ts_end      = time.mktime( datetime(2021,10,29,7,48,39).timetuple() )


with open("files/uaiai.txt", "r") as datafile:
```

```
    content = datafile.read().strip().split("\n")
    data = []
    for line in content:
        line_data = line.split("|")
        ts = int(line_data[0])
        if line_data[2] == "":  #Skip blanks
            continue
        if ts_start <=  ts and ts_end >= ts:
            data.append( {"ts": datetime.fromtimestamp(ts), "md5": line_data[1], "ua": line_data

unique = [el["ua"] for el in data]

unique = set(unique)
print( len(unique))
gen_resp( str(len(unique)) )
```

102 - Statistikk UserAgents #2

- file: "uaiai.txt"

Kan du finne ut hva som er den mest bruke UserAgenten i hele datasettet (- samme datasett som i oppgave #1)?

In [ ]:
```
uas = {}

with open("files/uaiai.txt", "r") as datafile:
    content = datafile.read().strip().split("\n")
    data = []
    for line in content:
        line_data = line.split("|")
        if line_data[2] == "":  #Skip blanks
            continue
        ua = line_data[2]
        if ua not in uas:
            uas.update({ua:1})
        else:
            uas[ua] += 1

cur = 0
winner = None
for ua, el in uas.items():
    if cur <  el:
        print( el, ua )
        winner = ua
        cur = el

gen_resp( winner )
```

103 - Prosentfordeling - netflow #1

- file: "theflow.csv"

1. Basert på følgende datasett, kan du finne hvor mange unike IP-adresser som eksisterer?
2. Datamodellen er bygget opp med følgende struktur:
   STARTTIME;SOURCE;DEST;PROTO;PACKETS;BYTES;DURATION
3. Eksempel på en linje: 2019-09-03T05:00:03;8.8.8.8:53;10.23.35.133:64352;udp;1;78;00:00:00

In [ ]:
```
IPS = []

with open("files/theflow.csv", "r") as datafile:
    content = datafile.read().strip().split("\n")
    for line in content[1:]:
        data = line.split(";")

        src_ip = data[1][ : data[1].find(":") ] if ":" in data[1] else data[1]
        if src_ip not in IPS:
            IPS.append( src_ip )

        dst_ip = data[2][ : data[2].find(":") ] if ":" in data[2] else data[2]
        if dst_ip not in IPS:
            IPS.append( dst_ip )

print( len(IPS) )
```

104 - parsing error

- files: "error-on-lines.csv"

1. Hvilke linjer har en feil som gjør at det ville vært problematisk å parse disse dataene? Headeren telles som linje 1.

   Avgi svaret som en kolonseparert liste uten mellomrom i stigende rekkefølge, eksempelvis: "13:37" - uten fnutter."

In [ ]:
```
from datetime import datetime

error_lines = []

def line_2_csv( line_str, sep=",", textmarker='"' ):
    columns = []

    text_index = 0
    while text_index < len(line_str):
        scan_2_marker = False
        if line_str[text_index] == textmarker:
            scan_2_marker = True
            count = line_str[text_index+1:].find(textmarker)
        else:
            count = line_str[text_index:].find(sep)
        if count == -1:
            count = len(line_str) - text_index

        if scan_2_marker:
            columns.append( line_str[text_index+1:text_index + count + 1] )
            text_index += count + 2
        else:
            columns.append( line_str[text_index:text_index+count] )
            text_index += count

        count = line_str[text_index:].find( sep )
        if count != -1:
            text_index += 1 + count #Skip separator
    return columns


csv = []
```

```
with open("files/error-on-lines.csv", "r") as datafile:
    content = datafile.read().strip().split("\n")
    for line_number, line_str in enumerate(content):
        line_str = line_str.strip()

        csv_line = line_2_csv( line_str )

        if len( csv_line ) != 4:
            error_lines.append( str(line_number+1) )
            continue
        if line_number == 0:
            continue

        csv_line[0] = datetime.strptime( csv_line[0], "%Y-%m-%dT%H:%M:%S.%f%z" )
        csv.append( csv_line )

print( ":".join(error_lines) )
```

105 - Visualisering #1

- file: "cyber.numbers"

1. Hva skal følgende tallrepresentasjon trolig fremstille?

In [ ]:
```
from matplotlib.pyplot import plot

content = [
    [[[50,61], [41,65], [30,65], [19,62], [12,57], [7,51], [6,43], [6,34], [5,27], [7,18], [10,1
    [[99,64], [88,64], [88,41], [67,1], [80,1], [93,30], [108,1], [120,1], [99,41], [99,64]]],
    [
        [[137,65], [137,1], [160,1], [168,2], [173,4], [175,6], [177,10], [178,14], [178,18], [1
        [[149,28], [149,11], [158,11], [163,13], [165,16], [165,20], [164,24], [161,26], [159,28
        [[149,53], [149,36], [158,36], [163,38], [165,41], [165,45], [164,49], [161,51], [159,53
    ],
    [[[205,64], [205,1], [244,1], [244,10], [217,10], [217,27], [243,27], [243,36], [217,36], [2
    [
        [[266,64], [266,1], [288,1], [296,2], [302,4], [305,7], [307,11], [309,18], [308,23], [3
        [[280,34], [280,11], [288,11], [294,12], [296,14], [298,16], [299,19], [299,23], [297,27
    ],
    [[[330,62], [330,51], [343,54], [353,54], [360,52], [362,50], [363,46], [362,43], [358,40],
    [[[395,65], [395,1], [407,1], [407,26], [430,26], [430,1], [441,1], [441,65], [430,65], [430
    [[[465,64], [465,1], [504,1], [504,10], [477,10], [477,27], [503,27], [503,36], [477,36], [4
    [[[570,65], [530,65], [530,1], [542,1], [542,54], [570,54], [570,65]]],
    [[[635,65], [595,65], [595,1], [607,1], [607,54], [635,54], [635,65]]]
]

for line in content:
    for dataset in line:
        x_arr = [x for x,y in dataset]
        y_arr = [y for x,y in dataset]
        plot(x_arr, y_arr)
```