In [ ]:
```
# Reference use
# Amos, D. (2020). A Practical Introduction to Web Scraping in Python. https://realpython.com/python-we
b-scraping-practical-introduction/
```

In [8]:
```
import sys, urllib.request, re
from bs4 import BeautifulSoup
from urllib.request import urlopen
```

In [2]:
```
scoreCard = {'vulnerability': 3, 'linux': 20, 'CVE':3, 'malware':5}
threshold = 50
```

In [3]:
```
def scoreHackerNews(homePageAbstract):
        score = 0
        abstractDict = {}
        for line in homePageAbstract.split():
            if line in abstractDict.keys():
                abstractDict[line] += 1
            else:
                abstractDict[line] = 1

        for val in scoreCard.keys():
            pattern = '.*(' + val + '|' + val.upper() + '|' + val.capitalize() + '|' + val.lower() +
').*'
            for key in abstractDict.keys():
                if re.search(pattern, key):
                    score += scoreCard[val] * abstractDict[key]
        return score
```

In [4]:
```
days = [31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31]
def pad(string):
    while len(string) < 2:
        return "0" + string
    return string
def generatePrevDay(year, month, day):
    if (int(day)) > 0:
        day = str(int(day)-1)
    elif month == "00":
        year = str(int(year)-1)
        month = "12"
        day = str(days[11])
    else:
        month = (str(int(month)-1))
        day = str(days[int(month)])
    day = pad(day)
    month = pad(month)
    year = pad(year)
    return year, month, day
```

In [5]:
```
import datetime
def generateDateRange(prevDaysAmount):#, year, month, day):
    today = datetime.datetime.now().strftime("%y-%m-%d").split("-")
    year = today[0]
    month = today[1]
    day = today[2]

    dates = [today[0]+"-"+today[1]+"-"+today[2]]

    for i in range(prevDaysAmount+1):
        try:
            year, month, day = generatePrevDay(year, month, day)
            dates.append(year+"-"+month+"-"+day)
        except:
            print("Failed date!")
    return dates
```

In [6]:
```
def HackerNewsParser(days, scoreCard, threshold):
    urlValidator = '^https://thehackernews.com/.*'
    links = []
    for date in generateDateRange(5):
        template = "https://thehackernews.com/search?updated-max={}T07:04:00-08:00&max-results=1500&sta
rt=1&by-date=false"
        baseUrl = template.format(date)
        page = urlopen(baseUrl)
        html = page.read().decode("utf-8")
        soup = BeautifulSoup(html, "html.parser")
        for link in soup.find_all("a"):
            if len(link.text) > 50 and re.search(urlValidator, link["href"]):
                testLink = link["href"]
                testPage = urlopen(testLink)
                testHtml = testPage.read().decode("utf-8")
                testSoup = BeautifulSoup(testHtml, "html.parser")
                testScore = 0
                for words in testSoup.find_all("p"):
                    for line in words:
                        try:
                            testScore += scoreHackerNews(line)
                        except:
                            continue
                if testScore >= 25 and testLink not in links:
                    links.append(testLink)
    return links
```

In [7]:
```
print(HackerNewsParser(10, scoreCard, threshold))
```

```
['https://thehackernews.com/2021/03/new-zoom-screen-sharing-bug-lets-other.html']
```