UFCFEL-15-3 Security Data Analytics and Visualisation

Portfolio Assignment 3: Large-Scale Data Exploration for Insider Threat Detection (2022)

The completion of this worksheet is worth a **maximum of 45 marks** towards your portfolio assignment for the UFCFEL-15-3 Security Data Analytics and Visualisation (SDAV) module.

Brief

In this task, you have been asked to investigate a potential security threat within an organisation. Building on your previous worksheet expertise, you will need to apply your skills and knowledge of data analytics and visualisation to examine and explore the datasets methodically to uncover which employee is acting as a threat and why. The company have provided you with activity logs for various user interactions for the past 6 months, resulting in a lot of data that they need your expertise for to decipher. They want to have a report that details the investigation that you have carried out, details of the suspected individual, and a clear rationale as to why this suspect is flagged. You will need to document your investigation, giving clear justification for your process using Markdown annotation within your notebook. You will need to provide a clear rationale for why you suspect a given individual to be acting as a threat, based on the pattern of activity that you identify.

This coursework is specifically designed to challenge your critical thinking and creativity, and is designed as an open problem. Examine the data and try to think how an individual user may appear as an anomaly against the remainder of the data. This could be an anomaly compared to a group of users, or an anomaly as compared over time.

Assessment and Marking

Marks will be allocated within the following criteria:

- Identification and justification of the suspicious behaviour (15)
- Analytical process and reasoning to deduce the suspicious behaviour (15)
- Use of informative visualisation and data exploration techniques (10)
- Clarity and professional presentation (5)

To achieve the higher end of the grade scale, you need to demonstrate creativity in how you approach the problem of identifying malicious behaviours, and ensure that you have accounted for multiple anomalies across the set of data available.

This assignment should be submitted as as PDF to your Blackboard portfolio submission as per the instructions in the assignment specification available on Blackboard. A copy of your work should also be provided via a UWE Gitlab repository, with an accessible link provided with your portfolio.

Contact

Questions about this assignment should be directed to your module leader (Phil.Legg@uwe.ac.uk). You can use the Blackboard Q&A feature to ask questions related to this module and this assignment, as well as the on-site teaching sessions.

Load in the data

In [6]:

```
# DO NOT MODIFY THIS CELL - this cell is splitting the data to provide a suitable subset of data to work with for this task.
# If you change this cell your output will differ from that expected and could impact your mark.

import random
import string
import pandas as pd
import matplotlib.pyplot as plt
import seaborn
import datetime

dataset_list = ['onlinebargains']
DATASET = dataset_list[0]

def load_data(DATASET):
    if DATASET in dataset_list:
        email_data = pd.read_csv('./T3_data/' + DATASET + '/email_data.csv', parse_dates=True, index_col=0)
        file_data = pd.read_csv('./T3_data/' + DATASET + '/file_data.csv', parse_dates=True, index_col=0)
```

```
    web_data = pd.read_csv('./T3_data/' + DATASET + '/web_data.csv', parse_dates=True, index_col=0)
    login_data = pd.read_csv('./T3_data/' + DATASET + '/login_data.csv', parse_dates=True, index_col=0)
    usb_data = pd.read_csv('./T3_data/' + DATASET + '/usb_data.csv', parse_dates=True, index_col=0)
    employee_data = pd.read_csv('./T3_data/' + DATASET + '/employee_data.csv', index_col=0)

    email_data['datetime'] = pd.to_datetime(email_data['datetime'])
    file_data['datetime'] = pd.to_datetime(file_data['datetime'])
    web_data['datetime'] = pd.to_datetime(web_data['datetime'])
    login_data['datetime'] = pd.to_datetime(login_data['datetime'])
    usb_data['datetime'] = pd.to_datetime(usb_data['datetime'])
  else:
    print ("DATASET variable not defined")
    return
  return employee_data, login_data, usb_data, web_data, file_data, email_data

employee_data, login_data, usb_data, web_data, file_data, email_data = load_data(DATASET)
employee_data
```

Out [6]:

The cell above is creating a set of DataFrames to work with. The set of tables are named as follows:

- employee_data
- login_data
- usb_data
- web_data
- file_data
- email_data

1. Begin investigation

First, I started my investigation by trying to find a common data between files to find the relation. Then, I studied each file individually to find the vulnerability that may cause security issue, for example, trying to find the users who logged in, but not logged out in the same day. Also the users who send multiple emails during the month, the files accessed for each user compared to his role, etc. I ended up with the following scenario:

1) Create a function that is checking user's log out time and compare it with the last time of accessing files, the usage of USB, the last email sent, and last website accessed during the day. In case theses times are larger than the log out datetime, it means somebody is accessing his PC. After implementing this code, I didn't find any of this happen. Below the code using this:

```
def load_data(DATASET):
    if DATASET in dataset_list:
        email_data = pd.read_csv('./T3_data/' + DATASET + '/email_data.csv', parse_dates=True, index_col=0)
        file_data = pd.read_csv('./T3_data/' + DATASET + '/file_data.csv', parse_dates=True, index_col=0)
        web_data = pd.read_csv('./T3_data/' + DATASET + '/web_data.csv', parse_dates=True, index_col=0)
        login_data = pd.read_csv('./T3_data/' + DATASET + '/login_data.csv', parse_dates=True, index_col=0)
        usb_data = pd.read_csv('./T3_data/' + DATASET + '/usb_data.csv', parse_dates=True, index_col=0)
        employee_data = pd.read_csv('./T3_data/' + DATASET + '/employee_data.csv', index_col=0)

        email_data['datetime'] = pd.to_datetime(email_data['datetime'])
        file_data['datetime'] = pd.to_datetime(file_data['datetime'])
        web_data['datetime'] = pd.to_datetime(web_data['datetime'])
        login_data['datetime'] = pd.to_datetime(login_data['datetime'])
        usb_data['datetime'] = pd.to_datetime(usb_data['datetime'])
    else:
        print ("DATASET variable not defined")
        return
    return employee_data, login_data, usb_data, web_data, file_data, email_data

employee_data, login_data, usb_data, web_data, file_data, email_data = load_data(DATASET)

def check_data(data,operation):
    suspecious=[]
    logoff=check_login(login_data)
    for i in range(1,12):
        month=str(i)
        if i< 10 :
            month='0'+str(i)
        logoff=logoff[logoff['month']==month]
        print('Login for Month ('+str(month)+'): '+str(len(logoff)))
        print('Kindly Wait.. Extracting data')
        datalist = pd.DataFrame({"user": data['user'].tolist(),
                         "date": data['datetime'].dt.date,
                         "time": data['datetime'].dt.strftime("%H:%M:%S"),
                         "datetime":  data['datetime'].tolist(),
                         "month": data["datetime"].dt.strftime('%m')})
        logoff_data=np.array(logoff)
        datalist=datalist[datalist['month']== month]
        print('Length of '+operation+' of Month ('+str(month)+'): '+str(len(datalist)))
        print('Kindly Wait.. Comparing data')
        for row in logoff_data:
            data_new=datalist[(datalist['user']== row[0]) &
(pd.to_datetime(datalist['date'])==  pd.to_datetime(row[1]))]
            max_date=np.max(pd.to_datetime(data_new['datetime']))
            maxRow=data_new[data_new['datetime']==max_date]
```

```
            data_arr=np.array(maxRow)
            for item in data_arr:
                if(pd.to_datetime(item[3]) >= pd.to_datetime(row[4])):
                    suspecious.append(row)
        if len(suspecious)>0:
            print("Suspecious users for Month ("+month+") are:")
            [print(p[0]) for p in suspecious]
        else:
            print("No Suspecious found on month ("+month+")")

def check_login(login_data):
    subset = pd.DataFrame({"user": login_data['user'].tolist(),
                           "date": login_data['datetime'].dt.date,
                           "time": login_data['datetime'].dt.strftime("%H:%M:%S"),
                           "action": login_data['action'].tolist(),
                           "datetime":  login_data['datetime'].tolist(),
                           "month": login_data["datetime"].dt.strftime('%m')})
    logoff_data=subset[subset["action"]=="logoff"]
    return logoff_data
```

2) Create a function that check the maximum function done by the user in a month.

```
def check_maximum(data,operation,user,isInvest):
    print("")
    print("-------------------------------")
    print("Start "+operation+" Investigation")
    print("-------------------------------")
    print("Please Wait...")
    print("")
    data["month"] = data["datetime"].dt.strftime('%m')
    subset = pd.DataFrame({"user": data[user].tolist(),
                           "month": data['month'].tolist(),
                           "summation": 0})
    df_data = subset.groupby(['user','month'],as_index=False).agg({'summation':'count'})
    if(isInvest==False):
        print("Distinct Data")
        print("-------------")
        print(df_data)

    distinct_data = df_data['user'].tolist()
    distinct_data = list(set(distinct_data))
    avgList=[]
    maxList=[]
```

```
maximum_no=0
maxAverage_no=0
maximum_data=""
maxAverage_data=""

for i in distinct_data:
  data_month=df_data[df_data['user'] == i]
  meanValue=np.floor(mean(data_month['summation']))
  maxValue=np.max(data_month['summation'])
  maxRow=np.array(data_month[data_month['summation']==maxValue])
  maxMonth=maxRow[0,1]

  maxList.append([i,maxMonth, maxValue])
  avgList.append([i, meanValue])
  plt.plot(data_month.month,data_month.summation)
  if maxValue > maximum_no :
    maximum_no=maxValue
    maximum_data=maxRow[0,0]

  if meanValue > maxAverage_no :
    maxAverage_no=meanValue
    maxAverage_data=maxRow[0,0]

maxList_new = pd.DataFrame(maxList, columns=['User', 'Month', 'Max Value'])
avgList_new = pd.DataFrame(avgList, columns=['User', 'Average Value'])
if(isInvest==False):
   print("Maximum Months Per User")
   print("----------------------")
   print(maxList_new)
   print("")
   print("Average Per User")
   print("----------------------")
   print(avgList_new)
   print("")
   plt.show()
print("Suspecious Users of "+operation)
print("------------------------------")
print("1- Maximum value="+str(maximum_no))
print("   User Name: "+str(maximum_data))
print("2- Maxmimum Average value="+ str(maxAverage_no))
print("   User Name: "+str(maxAverage_data))
print("")
```

- Check the maximum number of emails in a month and get the average number of sent emails by the user

```
check_maximum(email_data,"Email Sent",'sender',True)
```

- Check the user who access the maximum number of files in a month and get the average number of accessed files

```
check_maximum(file_data,"File Access",'user',True)
```

- Check the user who is most use USB in a month and get the average number of use USB

```
check_maximum(usb_data,"USB Usage",'user',True)
```

- Check the user who access the maximum number of websites in a month and get the average number of accessed websites

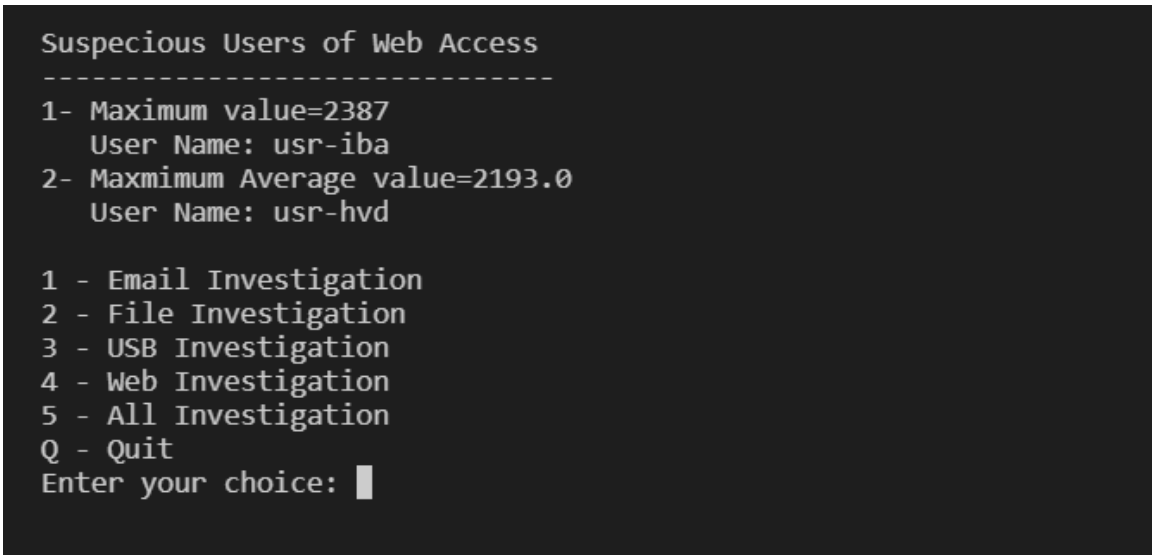
```
check_maximum(web_data,"Web Access",'user',True)
```

Conclusion - Summary of Findings

The first function is not fetching any suspect, it seems nobody is accessing another body's PC. The other function is using assumptions of the users who are the most accessing files, sending emails, using USB, or accessing websites are suspected.

```
1 - Email Investigation
2 - File Investigation
3 - USB Investigation
4 - Web Investigation
5 - All Investigation
Q - Quit
Enter your choice:
```

```
--------------------------------
Start Email Sent Investigation
--------------------------------
Please Wait...

Suspecious Users of Email Sent
--------------------------------
1- Maximum value=3188
User Name: usr-lfl@onlinebargains.com
2- Maxmimum Average value=2919.0
User Name: usr-lfl@onlinebargains.com


--------------------------------
Start File Access Investigation
--------------------------------
Please Wait...

Suspecious Users of File Access
--------------------------------
1- Maximum value=3225
User Name: usr-gsw
2- Maxmimum Average value=2925.0
User Name: usr-fbi


--------------------------------
Start USB Usage Investigation
--------------------------------
Please Wait...

Suspecious Users of USB Usage
--------------------------------
1- Maximum value=412
User Name: usr-jok
2- Maxmimum Average value=349.0
User Name: usr-okf


--------------------------------
Start Web Access Investigation
--------------------------------
Please Wait...
```


Suspecious Users of Web Access
1- Maximum value=2387
User Name: usr-iba
2- Maxmimum Average value=2193.0
User Name: usr-hvd
1 - Email Investigation
2 - File Investigation
3 - USB Investigation
4 - Web Investigation
5 - All Investigation
Q - Quit
Enter your choice: