Visualisation for Security of Websites against Data Breaches

1. Aarush Bhat
19BCE0564

2. Jyotir Aditya
19BCE084

Submitted to
Prof. Meenakshi S P for Data Visualisation - CSE3020

Abstract— The idea behind this project is to make people aware of data breaches and the safety of the websites they are using on a daily basis. Data has become the new gold in the 21st century and hackers around the world are targeting people who are less aware of their security features. This visualization informs our audience about the importance of security and shows an intuitive visualization that is easy to understand and comprehend for the average user of the internet.

This helps the average internet user to analyse when, how and where their precious data has been breached by malicious hackers. It offers a very customizable visualization which user can select and change based on their requirements

I. INTRODUCTION

The app offers an intuitive UI for easy Visualizations of Data breaches that have occurred from 2004 to early 2021, selectable from and to whichever year the users want to see. It characterizes the data breaches based on 3 filters, Method used to breach the data, Sensitivity of the data breached, and Sector of the organization whose data has been breached.

The filters help the user to manipulate the visualization based on what data they need, the filters can be used as follows:

1. Method: Method of breach consists of 6 attributes, Hacked, Oops, Poor Security, Lost Device, Inside Job.
2. Data Sensitivity: Levels of data sensitivity we categories in 5 attributes, Just email address/Online information, SSN/Personal details, Credit card information, Health & other personal records, Full details.
3. Sector: Sectors are categories into the following, web, retail, transport, government, telecoms, app, healthcare, tech, financial, legal, gaming, media, academic, energy, military.

The visualizations can be viewed between any year between the stated years based on users requirements using an intuitive slider.

The app covers most prospects that can be visualized using the dataset and is consumer-ready. The app also provides a guide on how to keep your privacy intact and safe on the internet.

A. Problem Statement

Stats for data breaches on websites:

1. Confirmed data breaches in the healthcare industry increased by 28% in 2020
2. Scams increased by 400% over the month of March, making COVID-19 the largest-ever security threat
3. The average cost per lost or stolen record in a data breach is $150.
4. In 2020.the country with the highest average total cost of a data breach was the United States at $6.64 million.

Due to the above stats, a platform must be there where one can easily verify the daily websites they are using and be safe in this ocean of scammers

B. Objective

WHAT IS A DATA BREACH?

A data breach exposes confidential, sensitive, or protected information to an unauthorized person. The files in a data breach are viewed and/or shared without permission.

Anyone can be at risk of a data breach — from individuals to high-level enterprises and governments. More importantly, anyone can put others at risk if they are not protected.

In general, data breaches happen due to weaknesses in:

1. Technology
2. User behaviour

As our computers and mobile devices get more connective features, there are more places for data to slip through. New technologies are being created faster than we can protect them.

Devices in the IoT sector are proof that we are increasingly valuing convenience over security. Many "smart home" products have gaping flaws, like lack of encryption, and hackers are taking advantage. Since new digital products, services, and tools are being used with minimal security testing, we'll continue to see this problem grow.

However, even if the backend technology was set up perfectly, some users will likely still have poor digital habits. All it takes is one person to compromise a website or network.

Without comprehensive security at both the user and enterprise levels, you are almost guaranteed to be at risk.

Protecting yourself and others starts with understanding how a data breach occurs.

II. Features

1. Visualisation of data breaches from 2004-2021
2. Idioms to compare which sector is prone to most hacking
    a.
    b. Categories of sectors:
    c. Web
    d. Retail
    e. Transport
    f. Government
    g. Telecoms
    h. App
    i. Healthcare
    j. Tech
    k. Financial
    l. Legal
    m. Gaming
    n. Media
    o. Academic
    p. Energy
    q. Military.
    r.
3. Know if your data has been breached
4. Which sites are safest to visit
5. How to protect oneself from being hacked
6. Safe password techniques
7. Visualisation of the reason why companies are breached

III. Data collection

Data has been collected from various resources like Kaggle and other websites like haveibeenpawned.com etc using their live API. This is combined with various attributes Company Name, alternative name, records lost, YEAR, date, story, SECTOR, METHOD, interesting story, DATA SENSITIVITY, displayed records etc . and was cleaned manually so as to implement the visualization techniques.

Link for Both clean and unclean .csv dataset

IV. Tasks

Visualisation*:* The visualisation techniques were decided and Plotly was used to implementing the actual visualisation

UI design: The design to the Frontend and the app was made in Figma with precise dimensions and transitions

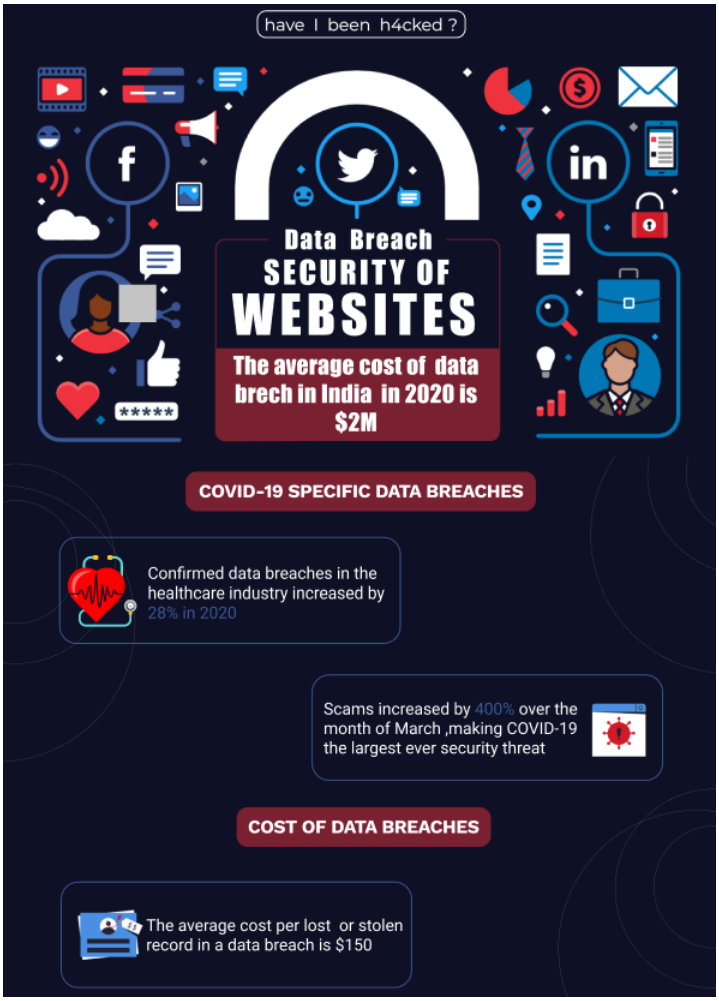
Frontend: The designed UI was implemented with HTML and CSS so to make it a consumer product where users can interact with the website.

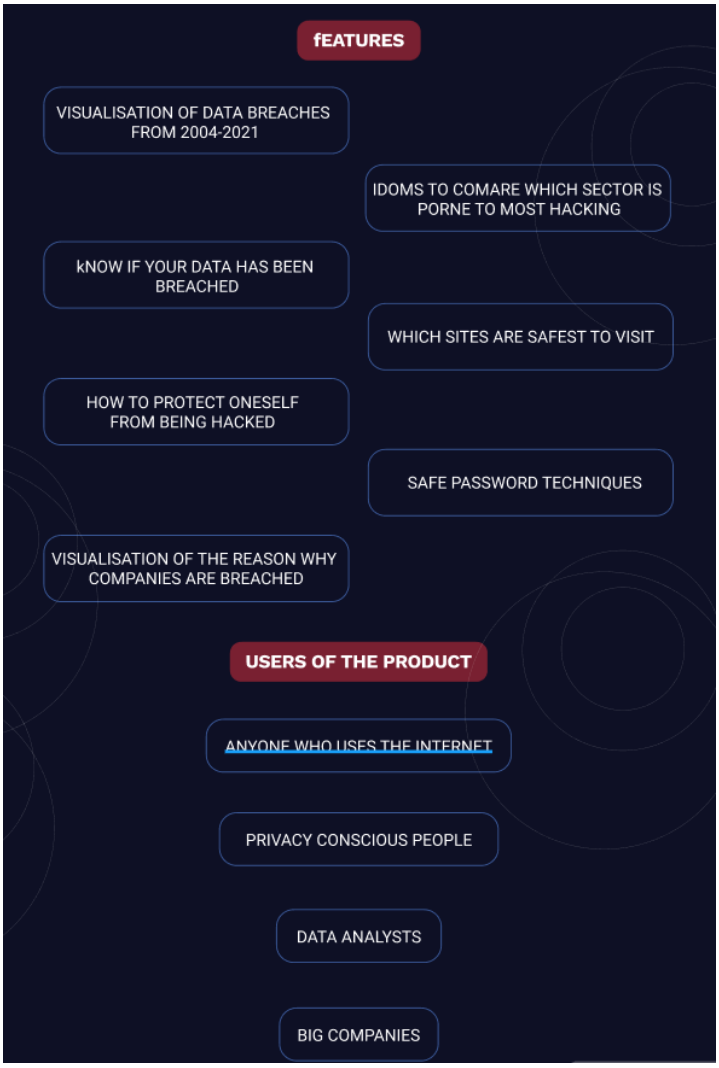
Backend: Flask app was made to implement the visualisation and to connect it with the API, dataset and frontend

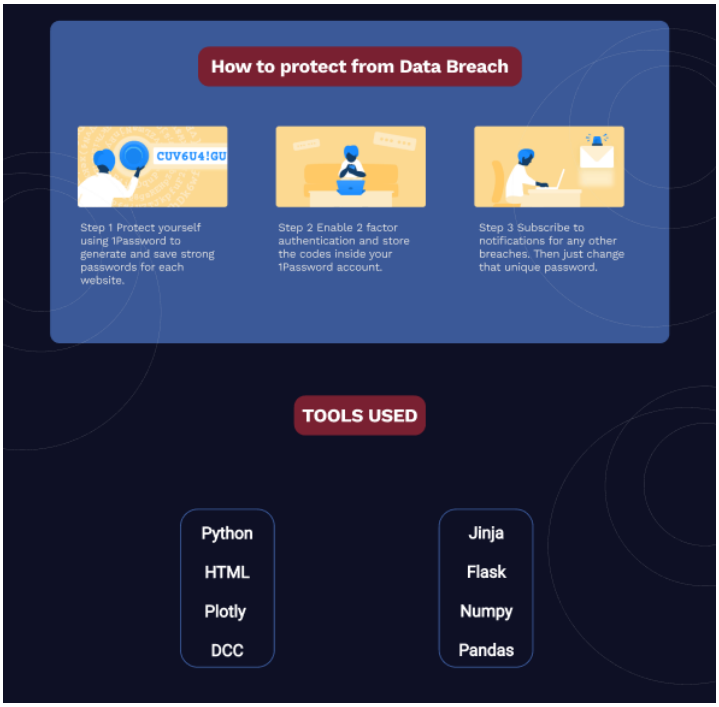
V. Implementation

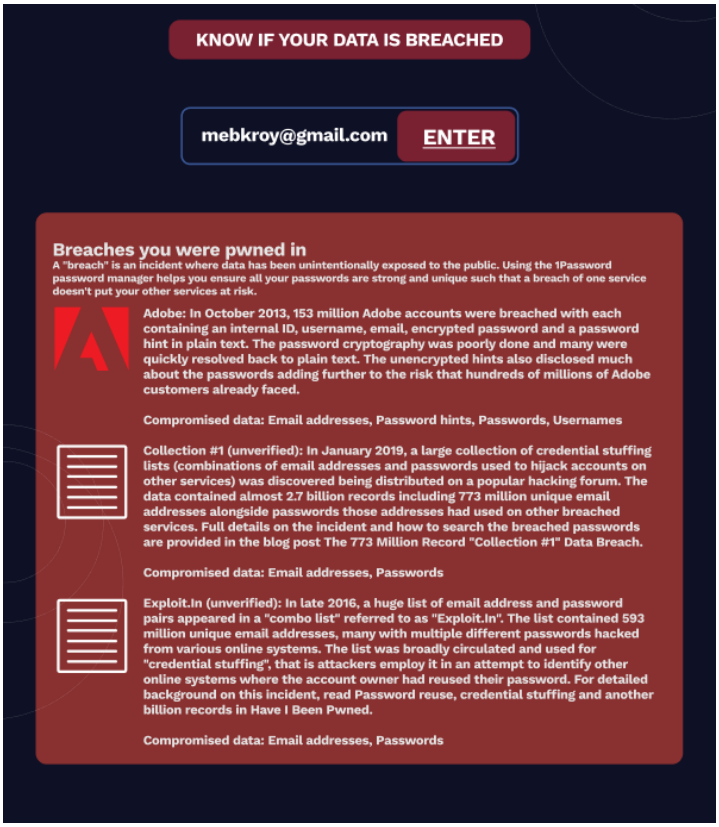
A. Landing Page

The App offers a landing page that consists of information and resources on data security and breaches.









B. Visualization Portal

The basic visualization portal is as follows:

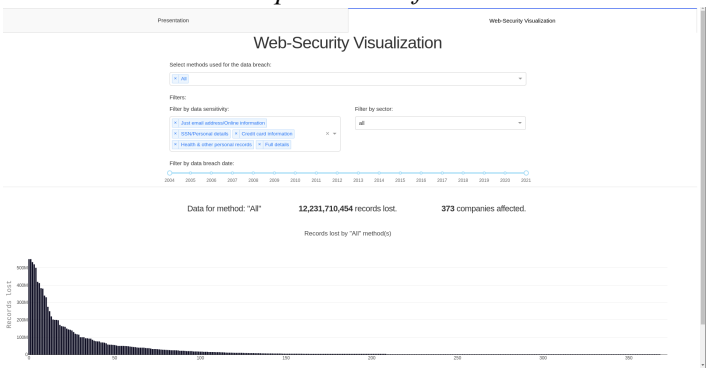


Figure: contains visualization will no filter selected

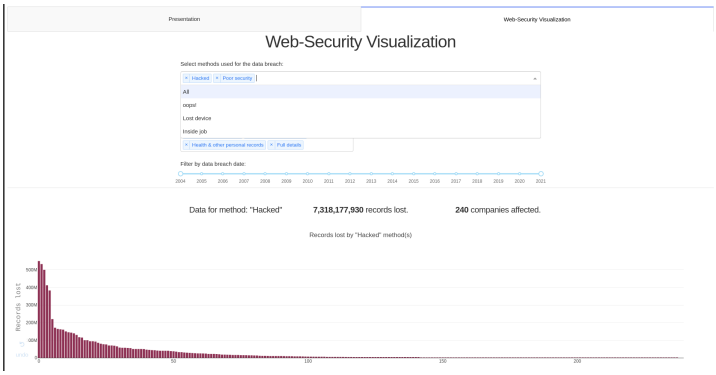


Figure: contains visualization with hacked and poor security filter selected for method.

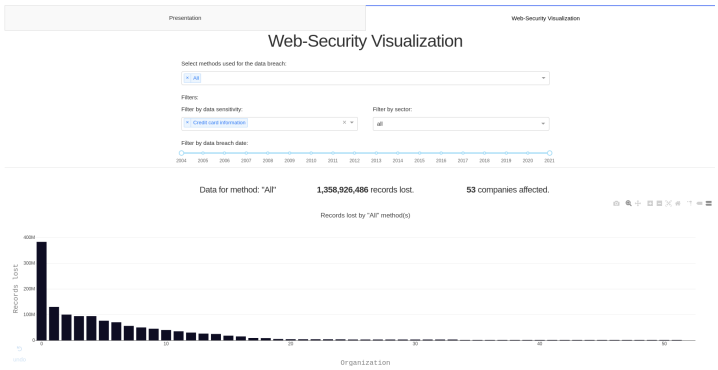


Figure: contains visualization with all the breaches in which credit card info was breached

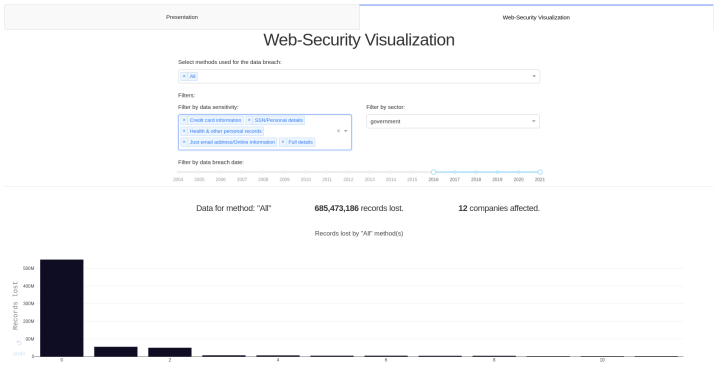


Figure: contains visualization with breaches between 2016 and 2021 from govt. institutions

C. Full Portal

The portal can be visited and used on Heroku at http://links.r-ush.co/dv-data-breaches-viz.

VI. Tools

The software, as well as hardware tools required for this project, are:

1) Python
2) HTML
3) CSS
4) Plotly
5) DCC
6) Jinja
7) Flask
8) NumPy
9) pandas

VII. Codes

A. *app.py*

Python Code:

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import dash
import dash_renderer
from dash.dependencies import Input, Output, State
import dash_core_components as dcc
import dash_html_components as html
import pandas as pd

external_stylesheets = ['https://codepen.io/chriddyp/pen/bWLwgP.css']
# default styles for Dash apps

app = dash.Dash(__name__, external_stylesheets=external_stylesheets)
app.title = 'Viz for data breaches'
app.scripts.config.serve_locally = True
server = app.server



### Load data ###
data = pd.read_csv('./data/breaches_clean.csv', thousands= ',')
data['SECTOR'] = data['SECTOR'].apply(lambda sector: sector.split(', '))
data.sort_values(by='records lost', inplace=True, ascending=False)

methods_list_names = ['All', 'Hacked', 'oops!', 'Poor security', 'Lost device', 'Inside job']
methods_list_values = ['all', 'hacked', 'oops!', 'poor security', 'lost device ', 'inside job']
methods_dict = { value: name for name, value in zip(methods_list_names, methods_list_values) }
method_options = [ { 'label': label, 'value': value} for label, value in zip(methods_list_names, methods_list_values) ]
method_color = {
    'all': '#0C0F25',
    'hacked' : '#903553',
    'oops!' : '#97BD5B',
    'poor security' : '#562972',
    'lost device ': '#EFAD24',
    'inside job': '#FFBAD2',
}

sector_list = ['web', 'retail', 'transport', 'government', 'telecoms',
        'app', 'healthcare', 'tech', 'financial', 'legal', 'gaming', 'media',
        'academic', 'energy', 'military']
sector_list.sort()
sector_list.insert(0, 'all')
sector_options = [ { 'label': sector, 'value': sector} for sector in sector_list ]



sensitivity_options = [
    {'value': 1, 'label': 'Just email address/Online information' },
    {'value': 2, 'label': 'SSN/Personal details' },
    {'value': 3, 'label': 'Credit card information' },
    {'value': 4, 'label': 'Health & other personal records' },
    {'value': 5, 'label': 'Full details' },
]

app.layout = html.Div(
    [
        dcc.Tabs([
        dcc.Tab(label='Presentation', children=[
            html.Div(
                [

html.Img(src=app.get_asset_url('landingPage.svg'),style = {'textAlign': 'center','max-width':'1400px'},)
                ],
                style = {'textAlign': 'center','background': '#0e1025', "color":"#fff"},
```

```
            )
        ]),
        dcc.Tab(label='Web-Security
Visualization', children=[
        html.Div(
            [
                html.H1(
                    'Web-Security
Visualization',
                    className='twelve columns',
                    id='title'
                ),
            ],
            style = {'textAlign': 'center'},
            className='row'
        ),
        html.Div(
            [
                html.P('Select methods used for
the data breach:'),
                dcc.Dropdown(
                    id='methods-selector',
                    options=method_options,
                    multi=True,
                    value=['all'],
                    placeholder='Select
method ... ',
                    clearable=False
                ),
            ],
            className='container'
        ),

        html.Div(
            [
                html.P('Filters:'),
                html.Div(
                    [
                        html.Div(
                            [
                                html.Div(
                                    [

html.P('Filter by data sensitivity:'),

dcc.Dropdown(

id='data-sensitivity-select',

options=sensitivity_options,

multi=True,

value=[1,2,3,4,5],

placeholder='Select sensitivity options ... ',
                                            ),
                                        ],

className='six columns'
                                    ),
                                    html.Div(
                                        [

html.P('Filter by sector:'),

dcc.Dropdown(

id='sector-select',

options=sector_options,

multi=False,

value='all',

placeholder='Select sector ... ',

clearable=False
                                            ),
                                        ],

className='six columns'
                                    ),
                                ],
                                className='row'
                            ),
                            html.Div(
                                [
                                    html.P('Filter
by data breach date:'),
```

```
dcc.RangeSlider(

id='year-slider',
                                    min=2004,
                                    max=2021,

value=[2004, 2021],
                                    marks={i: i
for i in range(2004, 2021+1)},
                                ),
                            ],
                            className='row',

style={'margin-top': '20'}
                        ),
                    ],
                )
            ],
            className='container',
            style={'margin-top': '20'},
        ),

        html.Hr(),

        html.Div(id='graphs')])

      ])
    ]
)



def get_data_for_method (method: str,
local_data: pd.DataFrame):
    if method == 'all':
        return local_data
    else:
        if method in methods_list_values:
            return
local_data[local_data['METHOD'] == method]
        else:
            raise Exception('method not found')
```

```
def method_header(method_name: str,
records_lost_no: int, companies_affected_no:
int):
    return html.Div(
            [
                html.H5(
                    f'Data for method:
"{method_name}"',
                    className='four columns',
                    style={'text-align':
'center'}
                ),
                html.H5(

[html.Strong("{:,}".format(int(records_lost_no)
)),' records lost.'],
                    className='four columns',
                    style={'text-align':
'center'}
                ),
                html.H5(

[html.Strong(companies_affected_no),' companies
affected.'],

                    className='four columns',
                    style={'text-align':
'right'}
                ),
            ],
            className='row container'
        )


@app.callback(
            Output('graphs', 'children'),
            [
                Input('methods-selector',
'value'),
                Input('year-slider', 'value'),

Input('data-sensitivity-select', 'value'),
                Input('sector-select',
'value'),
            ]
    )
def make_main_figure(methods, years,
```

```
selected_sensitivities, selected_sector):

    local_data = data.copy(deep=True)

    print(f'Years selected: {years}')
    list_years_set = list(range(years[0],
years[1] + 1))
    local_data =
local_data[local_data['YEAR'].isin(list_years_s
et)]

    print(f'Data sensitivity selected:
{selected_sensitivities}')
    if selected_sensitivities =/=
sensitivity_options:
        local_data = local_data [
local_data['DATA SENSITIVITY'].apply( lambda
sensitivity: sensitivity in
selected_sensitivities ) ]

    print(f'Sector selected:
{selected_sector}')
    if selected_sector =/= 'all':
        local_data = local_data [
local_data['SECTOR'].apply( lambda sectors:
selected_sector in sectors ) ]

    graphs = []
    print(f'Methods selected: {methods}')
    for method in methods:
        method_name = methods_dict[method]
        method_data =
get_data_for_method(method, local_data)

        lost_data = method_data['records lost']
        entities = method_data['Entity']

        trace = dict(
            type='bar',
            x=len(lost_data),
            y=lost_data,
            text=entities,
            name='Records lost',
            marker=dict(
                color = method_color[method],
            )
        );

        layout = {
            'title': f'Records lost by
"{method_name}" method(s)',

            'xaxis': {'title': 'Organization',
                      'titlefont': dict(
                          family='Courier
New, monospace',
                          size=18,
                          color='#7f7f7f'
                      ),
            },
            'yaxis': {'title': 'Records lost',
                      'titlefont': dict(
                          family='Courier
New, monospace',
                          size=18,
                          color='#7f7f7f'
                      )
            }
        }

        figure = dict(data=[trace],
layout=layout)
        dash_graph = dcc.Graph(

id=f"graph-{method_name}",
                        figure=figure,
                        config={
                            'hovermode':  "y",
                            'showLink': False,

'modeBarButtonsToRemove': ['sendDataToCloud',
'lasso2d', 'select2d']
                        }
                    )

        graphs.append(html.Div(
            [
                method_header(method_name,
lost_data.sum(), len(lost_data)),
                dash_graph,
                html.Hr()
```

```
            ],
            style={'margin-bottom': '10'}
        ))


    return graphs


if __name__ == '__main__':

    print('Using version ' + dash.__version__ +
' of Dash.')
    print('Using version ' +
dash_renderer.__version__ + ' of Dash
renderer.')
    print('Using version ' + dcc.__version__ +
' of Dash Core Components.')
    print('Using version ' + html.__version__ +
' of Dash Html Components.')

    app.server.run(debug=True, threaded=True)
```

For full code,visit the GITHUB repo
https://github.com/r-ush/viz-for-data-breaches

IX. Conclusion

The project helps the average internet user to understand the values of privacy over the internet using easy visualizations.

This project made us realise the importance of security in web browsing. It is well known that scammers are everywhere and looking for small data breaches to capitalize and use it against us.

Using the product Companies as well as users of Users can analyse the recent targets of these malicious hackers and can stay safe in this world of the internet.

IX. Future Work

Future plans for the project hold great responsibility in providing useful and correct information to the masses. We plan on integrating API's from https://haveibeenpwned.com/ for more detailed information on breaches and also some personalized features based on the user.