SIG CODECLUEDO

CAN YOU FIND ALL THE SECURITY AND PRIVACY FLAWS?

1ST PRIZE:
AI-Powered
Security Robot

RUNNER UP:
Lego
Tallneck

HOW TO PLAY

Fill in your answers overleaf and hand them into SIG (Booth G14) before Thursday 16th Feb, 2:00 pm. The winner will be announced during the OWASP closing ceremony.

```
01: import base64 as b, logging as logger, imageClassificationModel as model, flask
02: import confiscatedImages, webCrawlImages, os , sys, shutil
03: sys.path.append(os.path.join('~','downloads', 'geolib'))
04: import geolib
05:
06:
07: # Copyright National Intelligence
08: #Image location predictor - takes an image (e.g.posted on social media) by
09: # a Person Of Interest and guesses its location in the world based on
10: # visual clues,to help track them down and bring them to justice
11:
12:
13: def train(numRuns, confiscatedImageArchive, crawledImageArchive):
14:   iterable = confiscatedImageArchive.append(crawledImageArchive)
15:   trainSet = []
16:   for i in range(len(iterable)):
17:       x=iterable[i]
18:       ##don't train on older images to preserve privacy
19:       if not int(x.get("Date")[1:2]) + int(x.get("Date")[2:4])*30+ \
20:              (int(x.get("Date")[4:7])-1900) * 365 > 36531:
21:           updated = x.update("location",geolib.geolocate(x.get("imageEXIFgps")))
22:           trainSet.append(updated)
23:       try:
24:           #Obfuscate identity of Case owner employees in trainSet
25:           trainSet[-1].update("caseOwner", b.b64encode(x.get("caseOwner")))
26:       except TypeError:
27:           logger.error("%s could not be encoded to base64" %(x.get("caseOwner")))
28:
29:
30:   deeplearningModel = model.train(trainSet, numRuns)
31:   return deeplearningModel
32:
33: trainedModel = train(1000, confiscatedImages(), webCrawlImages())
34: def func(image, category):
35:   trainedModel.execute(image)
36:   shutil.copy2(image, 'queryarchive\\'+ category + '\\' )
37:   return trainedModel.guessedLocation(), trainedModel.guessProbability()
38:
39:
40: app = Flask(__name__)
41: @app.before_request
42: @app.route('/applyModel', methods=['POST'])
43: def applyModel ():
44:   if flask.remote_addr not in ['11.886.66.1']:
45:       flask.abort(403)
46:   else:
47:       file = flask.request.files['img']
48:       image = img.open(file.stream)
49:       category = flask.request.form.get('category')
50:       return func(image, category)
51:
52: if __name__ == '__main__':
53:    app.run(host='11.886.66.1', port='80', debug=True) #Requires sudo
```

Software Improvement Group

SIG CodeCluedo

Please provide the following details:

Name: _____

Business Email: _____

We will send you the answers and later in the year invite you to the four other Code Cluedo challenges, each increasing in difficulty. Test your security expertise.

Findings: *Find the security and privacy flaws*

Whodunnit? *Determine the root cause – be creative*

The root cause of these flaws is the _____ in the _____
(role) (location)
Because……

Software Improvement Group