Hackim 2019 WRITE-UP

Misc - mIAuth

CHALLENGE DESCRIPTION

* Fool that mlAuth*

SOLUTION

Files:

https://drive.google.com/file/d/1QvZBVns4ei1fqnhDe2uEBKiaEeRkdVwl/view?usp=sharing

This challenge is about fooling the authentication system of the organization which is using a machine learning, Every employee has a profile represented by a string on 784 hex values, Your goal here is to fake a profile that will trick the 99.9% accuracy to get access.

After understanding the algorithm it seems like the hex values in profile affect the accuracy, therefore increasing the value in the profile could change the accuracy drastically.

```python
from keras.models import load_model
import numpy as np

# An organisation has implemented an authentication system "mlAuth" using machine learning,
# which is 99.9% accurate. Every employee has a profile(represented by a string on 784 hex values).
# mlAuth is trained using these profiles to predict the probability of authenticity for an employee.
# System grants access only if the predicted probability is higher than 0.99. Hence,
# your aim is to generate a fake profile that will trick the 99.9% accurate mlAuth in granting you access.
# You can make use of dumped machine learning model to conduct your targeted attack. Are you smart enough to fool the "intelligent" mlAuth?

# profile should contain 784 hex values
# example profile
OLDprofile = '0x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x0
profile = '0x90x90x90x90x90x90x90x90x90x90x90x90x90x90x90x90x90x90x90x90x90x90x90x90x90x90x90x90x90x9

# mlAuth's internal implementation to predict probability of authenticity
def get_proba(profile):
    np.set_printoptions(suppress=True)
    prof_h = profile.split('0x')
    ip = [int(_, 16) for _ in prof_h[1:]]
    ip = np.array(ip, dtype='float32')/255
    # reshape profile as required by the trained model
    ip = ip.reshape([1,28,28,1])
    # load model
    model = load_model('./keras_model')
    predicted = model.predict(ip)[0][1]
    return predicted

res = get_proba(profile)
print 'predicted probability of authenticity is :'+str(res)


# YOU CAN SUBMIT THE FAKE PROFILE AS SHOWN BELOW
# curl http://ml.ctf.nullcon.net/predict?profile=0x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x00x0
```

After replace all the x00 to x90 in profile we managed to get a higher authenticity, After that we used the curl command as shown in the source code above to get the flag.

hackim19{wh0_kn3w_ml_w0uld_61v3_y0u_1337_fl465}