OCR

by PwnaSonic

Tags: python

Rating: 0

Add your writeup

▼▼▼OCR - WEB 73 52/549=9.5%▼▼▼

This writeup is written by @kazkiti_ctf

> **Because** creating real pwn challs was to mainstream, we decided to focus on the devel
> opment of our equation solver using OCR.
> https://ocr.ctf.insecurity-insa.fr/

https://ocr.ctf.insecurity-insa.fr/debug

↓

```
#!/usr/bin/python3
from flask import Flask,request,send_from_directory,render_template,abort
import pytesseract
from PIL import Image
from re import sub
from io import BytesIO
from flask_recaptcha import ReCaptcha
app = Flask(__name__)
app.config.update(
    MAX_CONTENT_LENGTH = 500 * 1024
)
recaptcha = ReCaptcha(app=app)
x = open("private/flag.txt").read()


@app.route('/',methods=['GET'])
def ind():
    return render_template("index.html")

@app.route('/debug',methods=['GET'])
def debug():
    return send_from_directory('./', "server.py")
```

```
@app.route('/equation',methods=['POST'])
def equation():
    if recaptcha.verify():
        if 'file' not in request.files:
            return render_template('result.html', result = "No file uploaded")
        file = request.files['file']
        print(file)
        if file and file.filename == '':
            return render_template('result.html', result = "No correct file uploade
d")
        if file:
            input_text = pytesseract.image_to_string(Image.open(BytesIO(file.read
())))
            print(input_text)
            formated_text = "=".join(input_text.split("\n"))
            formated_text = formated_text.replace("=","==")
            formated_text = sub('===+','==',formated_text)
            formated_text = formated_text.replace(" ","")
            print(formated_text)
            if any(i not in 'abcdefghijklmnopqrstuvwxyz0123456789()[]=+-*' for i in
 formated_text):
                return render_template('result.html', result = "Some features are st
ill in beta !")
            if formated_text.count('(') > 1 or formated_text.count(')') > 1 or forma
ted_text.count('[') > 1 or formated_text.count(']') > 1 :
                return render_template('result.html', result = "We can not solve com
plex equations for now !")
            if any(i in formated_text for i in ["import","exec","compile","tesserac
t","chr","os","write","sleep"]):
                return render_template('result.html', result = "We can not understan
d your equation !")
            if len(formated_text) > 15:
                return render_template('result.html', result = "We can not solve com
plex equations for now !")
            try:
                if "==" in formated_text:
                    parts = formated_text.split("==",maxsplit=2)
                    pa_1 = int(eval(parts[0]))
                    pa_2 = int(eval(parts[1]))
                    if pa_1 == pa_2:
                        return render_template('result.html', result = "Wow, it work
s !")
                    else:
                        return render_template('result.html', result = "Sorry but it
seems that %d is not equal to %d"%(pa_1,pa_2))
                else:
                    return render_template('result.html', result = "Please import a
 valid equation !")
            except (KeyboardInterrupt, SystemExit):
                raise
            except:
                return render_template('result.html', result = "Something went wron
g...")

@app.route('/js/<path:path>')
def send_js(path):
    return send_from_directory('js', path)
```

```
@app.route('/css/<path:path>')
def send_css(path):
    return send_from_directory('css', path)

@app.route('/img/<path:path>')
def send_img(path):
    return send_from_directory('img', path)

@app.route('/private/flag.txt')
def censorship():
    abort(403)

if __name__ == '__main__':
    app.run(host="0.0.0.0")
```

↓

【Read source code】

`x = open("private/flag.txt").read()` ⇒ flag contains `x`

`if any(i not in 'abcdefghijklmnopqrstuvwxyz0123456789()[]=+-*' for i in formated_text):` ⇒ OCR can read `abcdefghijklmnopqrstuvwxyz0123456789()[]=+-*`

`if len(formated_text) > 15:` ⇒ Up to 14 characters can be read

`pa_1 = int(eval(parts[0]))` & `pa_2 = int(eval(parts[1]))` ⇒ The left and right sides of the expression are evaluated with `eval()` and converted to `int`

```
if pa_1 == pa_2:
                        return render_template('result.html', result = "Wow, it work
s !")
                    else:
                        return render_template('result.html', result = "Sorry but it
seems that %d is not equal to %d"%(pa_1,pa_2))
```

⇒ If the values of the left side and the right side are different, they are displayed

---

【exploit】

I created an image called `ord(x[0]) = 0` and loaded it into OCR

↓

Sorry but it seems that 73 is not equal to 0 ⇒I

↓

Since I could guess `INSA {`, I got it from 5

↓

```
ord(x[5]) = 0  Sorry but it seems that 48 is not equal to 0  ⇒0
ord(x[6]) = 0  Sorry but it seems that 99 is not equal to 0  ⇒c
ord(x[7]) = 0  Sorry but it seems that 114 is not equal to 0 ⇒r
ord(x[8]) = 0  Sorry but it seems that 95 is not equal to 0  ⇒_
ord(x[9]) = 0   Sorry but it seems that 76 is not equal to 0 ⇒L
ord(x[10]) = 0   Sorry but it seems that 48 is not equal to 0    ⇒0
ord(x[11]) = 0   Sorry but it seems that 110 is not equal to 0   ⇒n
```

```
ord(x[12]) = 0   Sorry but it seems that 103 is not equal to 0   ⇒g
ord(x[13]) = 0   Sorry but it seems that 125 is not equal to 0   ⇒}
```

↓

`INSA{0cr_L0ng}`

Comments

Commento

Non sono un robot

reCAPTCHA
Privacy - Termini

Send