**WRITE UP LKSP JATENG 2024**

**SMK N 1 GANTIWARNO**

ATTACK DEFENSE

**Defense**

**12345 - Calculator Service > /home/frieren**

Terdapat suatu service kalkulator dengan file python dimana terdapat kerentanan pada source code-nya

*/home/frieren/calculator.py*

import ast

def header():

    print('=====================================')

    print('=         Simple Calculator         =')

    print('=           LKSJateng 2024          =')

    print('=====================================')

    print('Petunjuk penggunaan:')

    print('Untuk melakukan perhitungan, masukan')

    print('pernyataan matematika pada prompt')

    print('Contoh: \n>>> 2 + 2 atau \n>>> 5 \* 5')

    print('Operator yang tersedia: +, -, \*, /')

    print('Ketik \'exit\' untuk keluar dari program')

def calculate(statement):

    try:

        parsed\_statement = ast.parse(statement, mode='eval')

        result = eval(compile(parsed\_statement, filename='', mode='eval'))

        return str(result)

    except SyntaxError:

        return "Error: Pernyataan tidak valid"

    except ZeroDivisionError:

        return "Error: Tidak bisa melakukan pembagian dengan nol"

    except Exception as e:

        return "Error: " + str(e)

if \_\_name\_\_ == '\_\_main\_\_':

    header()

    while True:

        statement = input('>>> ')

        if statement == 'exit':

            break

        else:

            result = calculate(statement)

            print('Hasilnya adalah : ' + result)

Terdapat kerentanan pada fungsi eval() diamana fungsi tersebut dapat membuka celah untuk command injection, sehingga ketika aplikasi dijalankan dapat digunakan untuk memasukan input ekspresi python yang berbahaya. Dan juga kurangnya validasi terhadap input guna memastikan bahwa pernyataan yang dimasukan benar benar merupakan ekspresi dari matematika.

Kita bisa melakukan patching. Dengan menambahkan fungsi *is\_safe\_expression()* yang menggunakan regex untuk memeriksa apakah pernyataan hanya terdiri dari angka, operator, dan spasi. Selain itu, penggunaan *eval()* diganti dengan *ast.literal\_eval()* untuk mengevaluasi ekspresi matematika secara aman. Sehingga menjadi seperti ini.

import ast

import re

def header():

    print('=====================================')

    print('=         Simple Calculator         =')

    print('=           LKSJateng 2024          =')

    print('=====================================')

    print('Petunjuk penggunaan:')

    print('Untuk melakukan perhitungan, masukan')

    print('pernyataan matematika pada prompt')

    print('Contoh: \n>>> 2 + 2 atau \n>>> 5 \* 5')

    print('Operator yang tersedia: +, -, \*, /')

    print('Ketik \'exit\' untuk keluar dari program')

def is\_safe\_expression(expression):

    # Regex untuk mencocokkan hanya angka, operator, dan spasi

    return bool(re.match(r'^[\d\s\.\+\-\\*/()]+$', expression))

def calculate(statement):

    try:

        if not is\_safe\_expression(statement):

            raise ValueError("Pernyataan tidak valid")

        parsed\_statement = ast.parse(statement, mode='eval')

        # Mengevaluasi ekspresi menggunakan ast.literal\_eval()

        result = ast.literal\_eval(parsed\_statement.body)

        return str(result)

    except Exception:

        return "Error: Terjadi kesalahan dalam perhitungan"

if \_\_name\_\_ == '\_\_main\_\_':

    header()

    while True:

        statement = input('>>> ')

        if statement == 'exit':

            break

        else:

            result = calculate(statement)

            print('Hasilnya adalah : ' + result)

**6969 - File Upload > /opt/cdnapp/app**

Sejujurnya saya pergi ke *chat.openai.com* kemudian mencopy source yang berada di *app.py* kemudian mencari tau apa kerentanan dari service tersebut dan menyuruh beliau untuk mematching kode yang rentan tersebut. Kemudian kita perbaharui source code-nya dengan yang dihasilkan beliau. Berikut adalah source hasil kerja beliau

import os

import hashlib

from flask import Flask, render\_template, request, send\_file

app = Flask(\_\_name\_\_)

app.config['UPLOAD\_FOLDER'] = os.path.join(os.getcwd(), 'uploads')

ALLOWED\_EXTENSIONS = {'txt', 'pdf', 'png', 'jpg', 'jpeg', 'gif'}

MAX\_CONTENT\_LENGTH = 16 \* 1024 \* 1024  # Batasan 16 MB

def allowed\_file(filename):

    return '.' in filename and filename.rsplit('.', 1)[1].lower() in ALLOWED\_EXTENSIONS

def secure\_filename(fname):

    # Menghapus karakter yang tidak aman dari nama file

    fname = ''.join(c for c in fname if c.isalnum() or c in ['.', '\_', '-'])

    return fname

@app.route('/', methods=['GET','POST'])

def upload():

    if request.method == 'POST':

        f = request.files.get('file')

        if not f or f.filename == '':

            return render\_template('upload.html', error='No file selected')

        if not allowed\_file(f.filename):

            return render\_template('upload.html', error='File type not allowed')

        filename = secure\_filename(f.filename)

        path = os.path.join(app.config['UPLOAD\_FOLDER'], filename)

        try:

            f.save(path)

            return render\_template('upload.html', fname=filename, furl=request.host\_url + "uploads/" + filename)

        except Exception as e:

            return render\_template('upload.html', error='Failed to save file: {}'.format(e))

    return render\_template('upload.html')

@app.route('/uploads/<filename>')

def uploaded(filename):

    return send\_file(os.path.join(app.config['UPLOAD\_FOLDER'], filename))

if \_\_name\_\_ == "\_\_main\_\_":

    app.run(host='0.0.0.0', port=6969)

Kata beliau setelah di patching maka app akan melakukan filter terhadap file yang di-upload dan akan menghapus karakter dari nama file yang tidak aman.

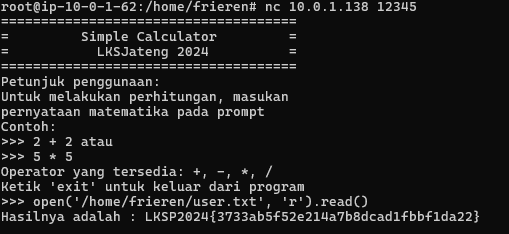
**Attack**

**12345 - Kalkulator**

Dengan memanfaatkan kerentanan yang terdapat pada source code kita bisa melakukan command injection untuk membaca file *user.txt* menggunakan syntax python.

*>>> open(‘/home/frieren/user.txt’, ‘r’).read()*

1. Mencari service kalkulator lawan mana yang sekiranya masih ada kerentanan
2. Akses service kalkulator lawan dengan cara *nc 10.0.1.138 12345*
3. Kemudian kita bisa langsung melakukan input dengan syntax python tersebut



Ketemulah flag-nya rek