ProxyAsAService

1) Flag is in environment

```
┌──(vigneswar㉿VigneswarPC)-[~/Web/ProxyAsAService/web_proxyasaservice]
└─$ cat Dockerfile
FROM python:3-alpine

# Install packages
RUN apk add --update --no-cache libcurl curl-dev build-base supervisor

# Upgrade pip
RUN python -m pip install --upgrade pip

# Install dependencies
RUN pip install Flask requests

# Setup app
RUN mkdir -p /app

# Switch working environment
WORKDIR /app

# Add application
COPY challenge .

# Setup supervisor
COPY config/supervisord.conf /etc/supervisord.conf

# Expose port the server is reachable on
EXPOSE 1337

# Disable pycache
ENV PYTHONDONTWRITEBYTECODE=1

# Place flag in environ
ENV FLAG=HTB{f4k3_fl4g_f0r_t3st1ng}

# Run supervisord
CMD ["/usr/bin/supervisord", "-c", "/etc/supervisord.conf"]
```

2) There is a server side request functionality

```python
from flask import request, abort
import functools, requests

RESTRICTED_URLS = ['localhost', '127.', '192.168.', '10.', '172.']

def is_safe_url(url):
    for restricted_url in RESTRICTED_URLS:
        if restricted_url in url:
            return False
    return True

def is_from_localhost(func):
    @functools.wraps(func)
    def check_ip(*args, **kwargs):
        if request.remote_addr != '127.0.0.1':
            return abort(403)
        return func(*args, **kwargs)
    return check_ip

def proxy_req(url):
    method = request.method
    headers =  {
        key: value for key, value in request.headers if key.lower() in ['x-csrf-token', 'cookie', 'referer']
    }
    data = request.get_data()

    response = requests.request(
        method,
        url,
        headers=headers,
        data=data,
        verify=False
    )

    if not is_safe_url(url) or not is_safe_url(response.url):
        return abort(403)

    return response, headers
```

3) There is a functionality to view env

```python
@proxy_api.route('/', methods=['GET', 'POST'])
def proxy():
    url = request.args.get('url')

    if not url:
        cat_meme_subreddits = [
            '/r/cats/',
            '/r/catpictures',
            '/r/catvideos/'
        ]

        random_subreddit = random.choice(cat_meme_subreddits)

        return redirect(url_for('.proxy', url=random_subreddit))

    target_url = f'http://{SITE_NAME}{url}'
    response, headers = proxy_req(target_url)

    return Response(response.content, response.status_code, headers.items())

@debug.route('/environment', methods=['GET'])
@is_from_localhost
def debug_environment():
    environment_info = {
        'Environment variables': dict(os.environ),
        'Request headers': dict(request.headers)
    }

    return jsonify(environment_info)
```

4) We can use this to confuse the domain

Domain Confusion

```
# Try also to change attacker.com for 127.0.0.1 to try to access localhost
# Try replacing https by http
# Try URL-encoded characters
https://{domain}@attacker.com
```

5) Got the flag