

Gotta Decrypt Them All

CSAW 2021 CTF

GoProSlowYo

Gotta Decrypt Them All 375

You are stuck in another dimension while you were riding Solgaleo. You have Rotom-dex with you to contact your friends but he won't activate the GPS unless you can prove yourself to him. He is going to give you a series of phrases that only you should be able to decrypt and you have a limited amount of time to do so. Can you decrypt them all?

nc crypto.chal.csaw.io 5001

2021-09-10

Contents

Gotta Decrypt Them All	3
Solve Script	3
Victory	6

Gotta Decrypt Them All

Writeup by: GoProSlowYo solved by Perryman

Team: OnlyFeet

Writeup URL: GitHub

```
1 You are stuck in another dimension while you were riding Solgaleo. You
  have Rotom-dex with you to contact your friends but he won't
  activate the GPS unless you can prove yourself to him. He is going
  to give you a series of phrases that only you should be able to
  decrypt and you have a limited amount of time to do so. Can you
  decrypt them all?
2
3 nc crypto.chal.csaw.io 5001
```

Solve Script

```
1 from pwn import *
2 import base64
3 import subprocess
4 import re
5 import codecs
6
7
8 def decodemorse(morse):
9     p = morse
10    p = p.replace(' .---- ', '1')
11    p = p.replace(' ..--- ', '2')
12    p = p.replace(' ...-- ', '3')
13    p = p.replace(' ....- ', '4')
14    p = p.replace(' ..... ', '5')
15    p = p.replace(' -.... ', '6')
16    p = p.replace(' --... ', '7')
17    p = p.replace(' ---.. ', '8')
18    p = p.replace(' ----. ', '9')
19    p = p.replace(' ----- ', '0')
20    p = p.split('/')
21    p = ''.join([chr(int(i)) for i in p])
22    print(p)
23    p = base64.b64decode(p).decode("utf-8")
24    p = p.split('\n')
25    p = [i.split(' = ') for i in p]
26    N = int(p[0][1])
```

```
27     e = int(p[1][1])
28     c = int(p[2][1])
29     print(p)
30     print(N)
31     print(e)
32     print(c)
33     command = "python3 ./RsaCtfTool/RsaCtfTool.py -n " + \
34         str(N) + " -e " + str(e) + " --uncipher " + \
35         str(c) + " --attack cube_root"
36
37     process = subprocess.Popen(command.split(), stdout=subprocess.PIPE)
38     output, error = process.communicate()
39     result = re.findall(b"(<=STR : b').*(?=')", output)
40     result = result[0].decode("utf-8")
41     result = codecs.encode(result, 'rot_13')
42     print(f'{result=}')
43     return result
44
45
46 def sendit(r, x, variable_name="", verbose=True):
47     r.sendline(str(x))
48     if verbose:
49         print(f"\tsend {variable_name}: {x}")
50
51
52 def recvit(r, variable_name="", verbose=True):
53     s = r.recv().strip()
54     s = s.decode("utf-8")
55     if verbose:
56         print(f"\treceived {variable_name}: {s}")
57     return s
58
59
60 def tryit():
61     curcase = 1
62     with remote("crypto.chal.csaw.io", 5001) as r:
63         r.recvuntil('What does this mean?')
64         v = recvit(r, verbose=False)
65
66         v = r.recvuntil('\r\n>>')
67         v = v.decode("utf-8")
68         v = v.strip('\r\n>>')
69
70         a = decodemorse(v)
71         print(f'{a=}')
72         sendit(r, a, verbose=True) # decode and send answer back
73
74         q = r.recvuntil('What does this mean?')
75         print(f'{q=}')
76         v = recvit(r, verbose=False)
77         v = r.recvuntil('\r\n>>')
```

```
78     v = v.decode("utf-8")
79     v = v.strip('\r\n>>')
80     a = decodemorse(v)
81     print(f'{a=}')
82     sendit(r, a, verbose=True) # decode and send answer back
83
84     q = r.recvuntil('What does this mean?')
85     print(f'{q=}')
86     v = recvit(r, verbose=False)
87     v = r.recvuntil('\r\n>>')
88     v = v.decode("utf-8")
89     v = v.strip('\r\n>>')
90     a = decodemorse(v)
91     print(f'{a=}')
92     sendit(r, a, verbose=True) # decode and send answer back
93
94     q = r.recvuntil('What does this mean?')
95     print(f'{q=}')
96     v = recvit(r, verbose=False)
97     v = r.recvuntil('\r\n>>')
98     v = v.decode("utf-8")
99     v = v.strip('\r\n>>')
100    a = decodemorse(v)
101    print(f'{a=}')
102    sendit(r, a, verbose=True) # decode and send answer back
103
104    q = r.recvuntil('What does this mean?')
105    print(f'{q=}')
106    v = recvit(r, verbose=False)
107    v = r.recvuntil('\r\n>>')
108    v = v.decode("utf-8")
109    v = v.strip('\r\n>>')
110    a = decodemorse(v)
111    print(f'{a=}')
112    sendit(r, a, verbose=True) # decode and send answer back
113
114    q = r.recvuntil('What does this mean?')
115    print(f'{q=}')
116    v = recvit(r, verbose=False)
117    v = r.recvuntil('\r\n>>')
118    v = v.decode("utf-8")
119    v = v.strip('\r\n>>')
120    a = decodemorse(v)
121    print(f'{a=}')
122    sendit(r, a, verbose=True) # decode and send answer back
123
124    print('*****')
125    v = recvit(r, verbose=True)
126    v = recvit(r, verbose=True)
127    v = recvit(r, verbose=True)
128
```

```
129  
130 tryit()
```

Victory

Submit the flag and claim the points:

flag{some FLAG}