b01lers 2020 scrambled

Saturday, March 7, 2020 12:16 PM

http://web.ctf.b01lers.com:1002/

On every call, you et a Set-Cookie with a value where the middle few chars change everytime seemingly randomly.

Set-Cookie: transmissions=kxkxkxkxshWi10kxkxkxkxsh; expires=Sat, 14-Mar-2020 16:49:48 GMT; Max-Age=600; path=/
Set-Cookie: transmissions=kxkxkxkxshar27kxkxkxkxsh; expires=Sat, 14-Mar-2020 16:51:26 GMT; Max-Age=600; path=/
Set-Cookie: transmissions=kxkxkxkxsh_W9kxkxkxkxsh; expires=Sat, 14-Mar-2020 16:51:48 GMT; Max-Age=600; path=/
Set-Cookie: transmissions=kxkxkxkxshtf2kxkxkxkxsh; expires=Sat, 14-Mar-2020 16:52:12 GMT; Max-Age=600; path=/

Do it enough times and you start to see some of the middle parts repeat.

transmissions=kxkxkxkxshf%7B3kxkxkxkxsh

decodes to **f{3**

The flag format is pctf{blah} so maybe these are flag fragments

pctf{3k

Wrote this program to repeatedly make this request, pull out the middle portion and throw it in a map.
Since they are random, do it lots of times in hopes of fully populating the map with all possible values.

```
import os
import requests
import sys
import urllib


BASE_URL = 'http://web.ctf.b01lers.com:1002/index.php'

def getFragment():
    url = BASE_URL
    cookies = {'frequency': '1', 'transmissions': '0'}
    response = requests.get(url, cookies=cookies, allow_redirects=False)

    header = response.headers['Set-Cookie']
    startIndex = header.index('transmissions=')
    endIndex = header.index(';', startIndex)
    value = header[startIndex+14:endIndex]

    if value.endswith('kxkxkxkxsh'):
        value = value[0:-10]

    if value.startswith('kxkxkxkxsh'):
        value = value[10:]
```

```
    value = urllib.unquote(value)
    return value

fragments = {}

for i in xrange(0, 2000):
    fragment = getFragment()
    if fragment in fragments.keys():
        print "already had: " + fragment
    fragments[fragment] = 1

sorted_keys = sorted(fragments.keys())
print sorted_keys
```

[',C25', ',I45', ',M58', ',T35', 'Be53', 'Ca26', 'De62', 'Do5', 'Fa19', 'Ha49', 'It46', 'My59', 'Te36', 'Wi10', '_B52', '_D61', '_F18', '_H48', '_W9', '_t14', 'al20', 'ar27', 'as50', 'co41', 'ct1', 'e,34', 'e,44', 'e_17', 'eg54', 'el37', 'em63', 'en23', 'es39', 'f{3', 'gu55', 'h_13', 'he16', 'it11', 'iv30', 'le22', 'le38', 'll21', 'mo64', 'n,24', 'n,57', 'n_8', 'ni29', 'ns66', 'on65', 'op42', 'or32', 'ow6', 'pc0', 'pe43', 're33', 'rn28', 's_51', 'sc40', 's}67', 't_47', 'tf2', 'th12', 'th15', 'un56', 'vo31', 'wn7', 'y_60', '{D4']

I get this output every time so it is likely exhaustive!

Upon study, it is clear that there are always 2 letters then a number!!

Some fragments look like part of a flag like f{ which would be part of pctf{

I realized that the number is the position of the two letters in the flag

pc0 --> flag[0] = 'p', flag[1] = 'c'
tf2 --> flag[2] = 't', flag[3] = 'f'
f{3 --> flag[3] = 'f', flag[4] = '{'

Wrote this function to piece together the fragments collected earlier.

```
def studyFragments():
    flagLetters = [0] * 1000
    fragments = [',C25', ',I45', ',M58', ',T35', 'Be53', 'Ca26', 'De62', 'Do5', 'Fa19',
'Ha49', 'It46', 'My59', 'Te36', 'Wi10', '_B52', '_D61', '_F18', '_H48', '_W9', '_t14',
'al20', 'ar27', 'as50', 'co41', 'ct1', 'e,34', 'e,44', 'e_17', 'eg54', 'el37', 'em63',
'en23', 'es39', 'f{3', 'gu55', 'h_13', 'he16', 'it11', 'iv30', 'le22', 'le38', 'll21',
'mo64', 'n,24', 'n,57', 'n_8', 'ni29', 'ns66', 'on65', 'op42', 'or32', 'ow6', 'pc0', 'pe43',
're33', 'rn28', 's_51', 'sc40', 's}67', 't_47', 'tf2', 'th12', 'th15', 'un56', 'vo31', 'wn7',
'y_60', '{D4']
    for fragment in fragments:
        chars = fragment[0:2]
        num = int(fragment[2:])
        for i in xrange(0,2):
            flagLetters[num+i] = chars[i]

    flag = ''
    for c in flagLetters:
```

```
        if c == 0:
            break;
        flag += c

    print flag


studyFragments()
```

pctf{Down_With_the_Fallen,Carnivore,Telescope,It_Has_Begun,My_Demons}