挖src的时候经常会遇到分析前端js加密，我一般模拟前端加密的时候会选择bp+jsEncrypt

考点一，前端js加密分析+工具模拟

本来是单纯md5，怕直接使用python发包，于是简单魔改了下md5.js

发现是salt+明文，然后调用`md5.js`中的`hex_md5()`

考点二、密码字典

工具的使用

`main.js`

```
//Java调用的主函数
function burpJsEncrypter(rawPayload){

    var encryptedPayload;


    //===============加密开始=================
    var param = "salt" + rawPayload;
    encryptedPayload = hex_md5(param);

    //===============加密结束=================

    return encryptedPayload;
}

//md5.js
...
```

三、结果

Filter: Showing all items

Request	Payload	Status	Error	Timeout	Length	Comment
3	087bdf4c11317af76020ea61c1026...	200	☐	☐	331	
7	087bdf4c11317af76020ea61c1026...	200	☐	☐	331	
0		200	☐	☐	288	
1	04a814aafc0ab828939e7a9370dfd...	200	☐	☐	288	
2	5c38e47784b1ed5a2c18bffd94bf6b...	200	☐	☐	288	
4	b65eb7ffa9f399cb52df79653a9ae6...	200	☐	☐	288	
5	99b2ccc9eafd1d708b3e5361e53e7...	200	☐	☐	288	
6	685d643910641b9042013ce9253f...	200	☐	☐	288	
8	b65eb7ffa9f399cb52df79653a9ae6...	200	☐	☐	288	
9	99183ff2828747cf4d923f280ed089...	200	☐	☐	288	
10	97126d9245646c33fc7fa940500e0...	200	☐	☐	288	
11	64a75656a69448a97c16c480b0c6...	200	☐	☐	288	
12	142edf9666265942c2b313b6357e...	200	☐	☐	288	
13	98be7354a959d3e8403fc5cbf5682...	200	☐	☐	288	

Request | Response

Raw | Headers | Hex

```
HTTP/1.1 200 OK
Date: Mon, 11 May 2020 17:30:59 GMT
Server: Apache/2.4.39 (Win64) OpenSSL/1.1.1b mod_fcgid/2.3.9a mod_log_rotate/1.02
X-Powered-By: PHP/7.2.9
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Transfer-Encoding: chunked
Content-Type: text/html;charset=utf-8

"flag{0bcbd1cf-ae88-49e7-b895-964eff6469de}"
```