

记录一次前端JS加密绕过 | 绕过前端解密的两种方法

原创 lingwu HACK学习呀

2020-06-25原文

一、背景：

一个银行较小的系统，数据包传输的值加密处理。故扣加解密代码编写脚本，以便测试方便。

二、坑的由来：

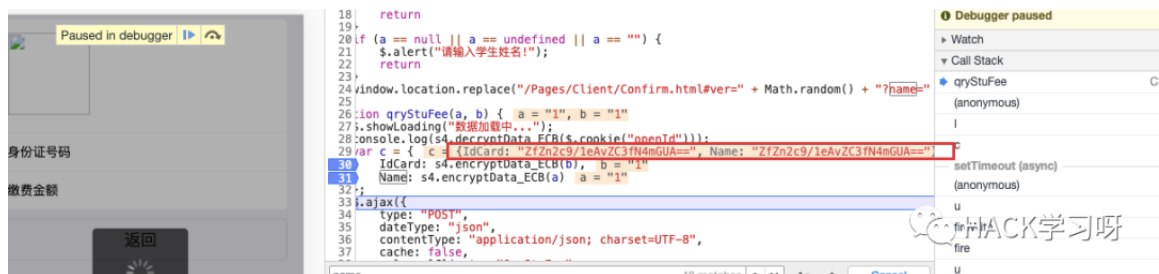
过程：

抓包发现数据加密，意料之中，常规思路寻常分析加密函数。

```
53A930C10B57/E390B/AB0092301/UEB9F1AB944363330599F68C0C13189AA3
DD3F30F7E3E41D6F580AA96A282941D1FCE7D0D6F787AC10814CEB80466173FC16
3EFD0C21A9C8A6FA65A0C92F8337435B5700EDBA9A97F609CF7521D4E7EE581E37
BE566489F2F18D464FB8858891671E286E790C69B1A56527A838091BFEB92F4875F
33852862A2418D558DA90232011B7BF360D3BE3A25714CE7EC1160738901C83B3
3DB994F05EA8E0FFA90989
Content-Type: application/json; charset=UTF-8
Orig:
Refer: Client/Confirm.html
Accept-Encoding: gzip, deflate
Accept-Language: zh-CN,zh;q=0.9
Cookie: ASP.NET_SessionId=aqozgc3fo2xvea3c11xjx02;
openid=vjFvj1BSUKu0DCNbjjPAXA==
Connection: close

{"code":0,"message":"成功","data":{"EnterAcct":{"TradeNo":"10350009262220200507160
007818","OpenId":"123","Sen_IdCard":"Ugt2cv/QP4At/GQqXoOaVw==","Sen_Name":"IO
V3YVpWLPcRyH/CzMKcftRsFaaq/NQsz2MZ/jZmji8=","IdCard":"Ugt2cv/QP4At/GQqXoOa
Vw==","Name":"IOV3YVpWLPcRyH/CzMKcftRsFaaq/NQsz2MZ/jZmji8=","SchoolCode":"0
4507","SchoolName":"P9g0a86rYVAF+IMoE03zUFG1m9RlxayYqcS5T5ZNAg0=","Grade":
"sC2bRmEskMu6DHMIq5veuw==","Class":"8zr5grqrNOBJGxpntooowg==","SumAmt":0.
01,"PayDate":null,"CardNo":null,"BillNo":null,"PayState":0,"State":0,"LogID":null,"CheckSta
te":null,"CheckDate":null,"CheckLogID":null,"InvoiceState":null,"InvoiceDate":null,"Invoice
No":null,"InvoiceType":null,"InvoiceLogID":null,"InvoiceDeletLogID":null,"Notes":null},"En
terAcctDetails":{"EnterAcctID":98,"SubjectID":2844822070bec58f01717221ca230049",
"SubjectCode":"WoR+47LmFbqeY0ssAlhOaU81xg2j0V1AH035v81178PlqfAMfD5xdM
hZxfhV0Y","SubjectName":"KLOVZ/EX/enX2uij7dL","UnitPrice":1.1,"TotalA
mt":0.01,"PayID":"WoR+47LmFbqeY0ssAlhOaUHD6IR7USVgfkCmFFA6gprRPlqfAMfD5xdMhZxfhV0Y",
"PlanID":"WoR+47LmFbqeY0ssAlhOaUHD6IR7USVgfkCmFFA6gprRPlqfAMfD5xdMhZxfhV0Y",
"PlanID":"WoR+47LmFbqeY0ssAlhOaUHD6IR7USVgfkCmFFA6gprRPlqfAMfD5xdMhZxfhV0Y"}}}
```

看代码就能大致推测开发的水平好坏，看着前端代码，越看越乐。。。。



在console执行函数，出正解结果。接下来扣代码，写脚本：

```

502         if (v && one.length == 8) {
503             var bytesLength = v[0].length;
504             var store = _arr[i].toString(2).slice(7 - bytesLength);
505             for (var st = 1; st < bytesLength; st++) {
506                 store += _arr[st + i].toString(2).slice(2);
507             }
508             str += String.fromCharCode(parseInt(store, 2));
509             i += bytesLength - 1;
510         } else {
511             str += String.fromCharCode(_arr[i]);
512         }
513     }
514     return str;
515 }
516
517 var a = "11HDESaAhiHHugDz";
518 var plain = new sm4utils(a).encryptData_ECB("lingwu");
519 console.log(plain);
520

```

问题 31 输出 调试控制台 终端

```

[Running] node "/Users/dy.zhang/Desktop/sm4_enAndDe.js"
/Users/dy.zhang/Desktop/sm4_enAndDe.js:433
    var cipherText = base64js.fromByteArray(encrypted);
                        ^

```

ReferenceError: base64js is not defined


 HACK学习呀

报错，debug追踪一下。

```

(function (r) {
  if (typeof exports === "object" && typeof module !== "undefined") {
    module.exports = r();
  } else {
    if (typeof define ===
      "function" && define.amd) {
      define([], r);
    } else {
      var e;
      if (typeof window !== "undefined") {
        e = window;
      } else {
        if (typeof global
          !== "undefined") {
          e = global;
        } else {
          if (typeof self !== "undefined") {
            e = self;
          } else {
            e = this;
          }
        }
      }
    }
  }
  e.base64js = r();
}
})(function () {
  var r, e, t;
  return function r(e, t, n) {
    function o(i, a) {
      if (!t[i]) {
        if (!e[i]) {

```

 HACK学习呀

发现引用的第三方包，弄了一个自执行函数，闭包了。。。查资料，发现自执行函数在函数内部定义的变量和函数就只能在函数内部访问，**在外部无法访问**，在该上下文环境中，调用函数时就提供了一个创建私有成员的方式，所以我执行脚本报错。

接下来做了N次尝试，这个问题困扰好久没解决，（没办法，js太差）

1.尝试解包。我错了，我高估了自己。

2.直接抽取关键代码。

3.看到是base64js，相直接扣出原版的base64试试，发现base64js是经过二次开发的，加密函数调用的他的方法原版没有。

4.在nodejs上执行不成功，用python调用js执行看下。结果一样。

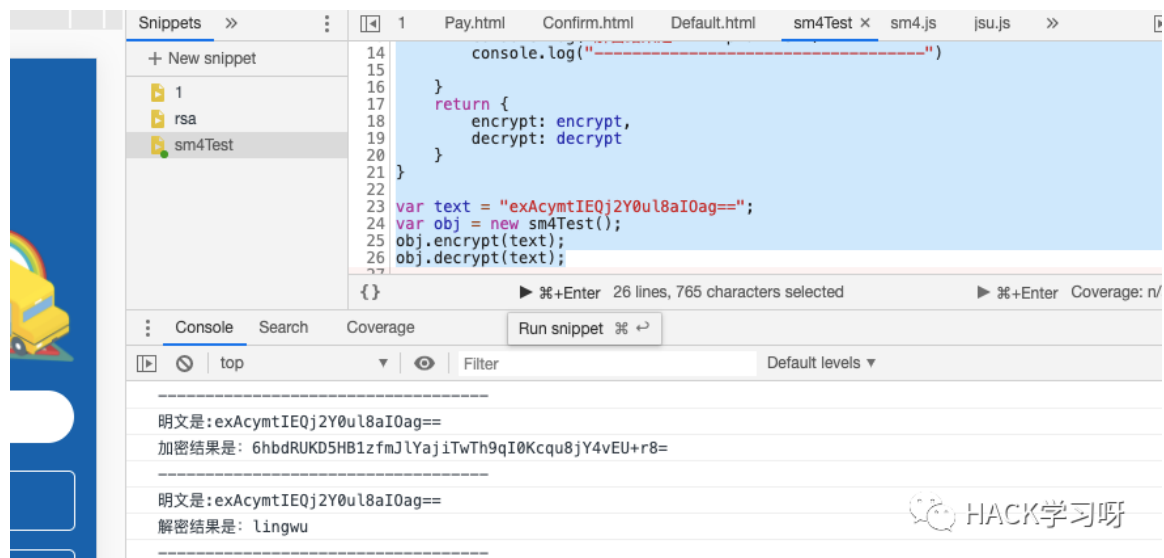
时间成本太高了，，，换方案吧，估计不在我能力范围内。

三、替代方案：

这就是没办法的办法，直接在snippets工具中执行测试脚本。

```
1  var sm4Test = function(text) {
2      var a = "11HDESaAhiHHugDz";
3      function encrypt(text) {
4          console.log("-----")
5          encryptText = new sm4utils(a).encryptData_ECB(text);
6          console.info("明文是:" + text)
7          console.info("加密结果是: " + encryptText);
8          console.log("-----")
9      }
10     function decrypt(text) {
11         plainText = new sm4utils(a).decryptData_ECB(text);
12         console.info("明文是:" + text)
13         console.log("解密结果是: " + plainText)
14         console.log("-----")
15     }
16     }
17     return {
18         encrypt: encrypt,
19         decrypt: decrypt
20     }
21 }
22
23 var text = "exAcymtIEQj2Y0ul8aIOag==";
24 var obj = new sm4Test();
25 obj.encrypt(text);
26 obj.decrypt(text);
```

结果如下：



```
14 console.log("-----")
15
16 }
17 return {
18   encrypt: encrypt,
19   decrypt: decrypt
20 }
21 }
22
23 var text = "exAcymtIEQj2Y0ul8aI0ag==";
24 var obj = new sm4Test();
25 obj.encrypt(text);
26 obj.decrypt(text);
27 }
```

明文是:exAcymtIEQj2Y0ul8aI0ag==
加密结果是: 6hbdRUKD5HB1zfmJlYaj iTwTh9qI0Kcqu8jY4vEU+r8=
明文是:exAcymtIEQj2Y0ul8aI0ag==
解密结果是: lingwu

可看到结果正常返回，虽然没有使用nodejs看起来顺眼，但比在console中进行批量操作要省事的多，为折中办法。

四、总结：

简单总结自执行函数：

1.基本格式：

```
1 (function () { /* code */ }
  (args));
```

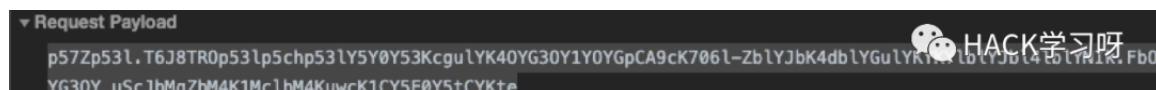
2.作用：

- o js中没有块级作用域，用来隔离作用域避免污染，或者截断作用域链，避免闭包造成引用变量无法释放。
- o 利用立即执行特性，返回需要的业务函数或对象，避免每次通过条件判断来处理。

五、分析前端解密的两种方法：

方法一：常规方法

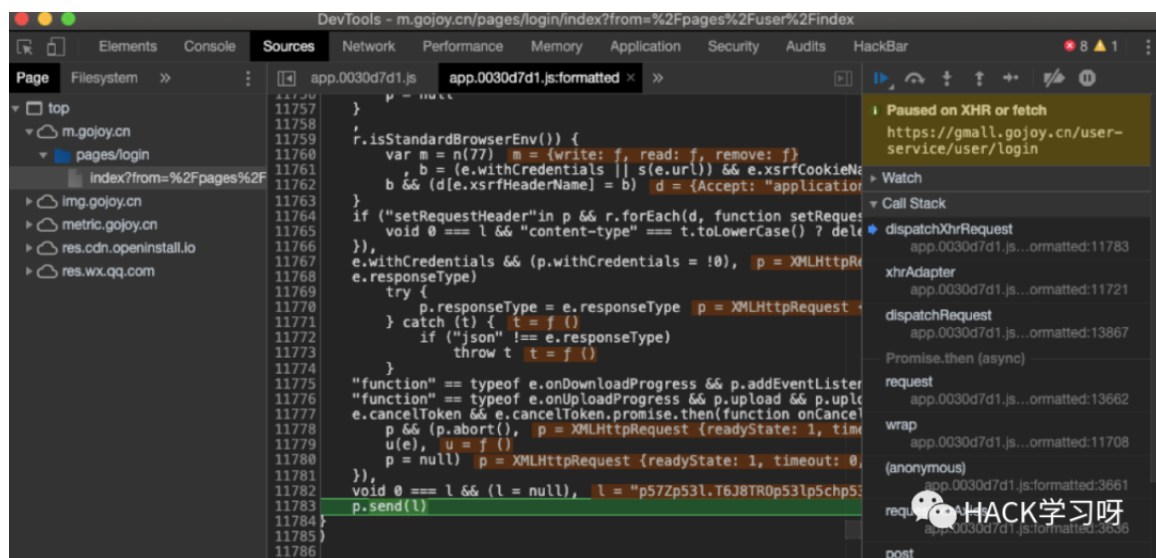
- 1.访问x系统，发现系统js 有反调试，设置条件断点绕过反调试。
- 2.在请求包中发现如下密文。



- 3.此时发现该数据为xhr方式加载。

Name	Status	Type	Initiator	Size
data:image/svg+xml;...	200	svg+xml	Other	(memory ca...
login	200	xhr	app_0030d7d1.js	602 B
login	200	xhr	Other	789 B

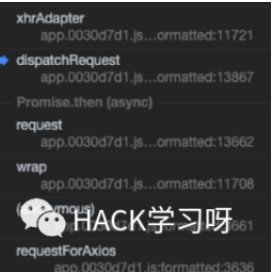
- 5.重新登录，可看到如下所示：



可发现上图中的I参数就是该数据包请求时的密文。

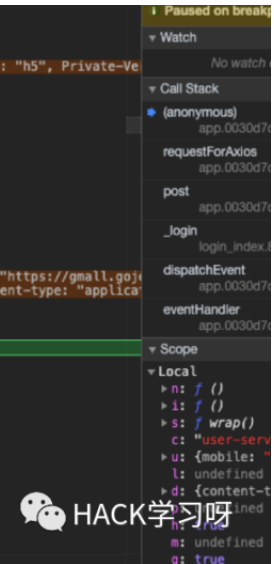
- 6.在xhr中下断一般直接看堆栈调用。

```
3653     , T = a.c.requestUrlProd
3654     , S = Object(r.a){"host":
3655     S && (T = S),
3656     -1 === c.indexOf("http") && (y = T + ("/" === c[0] ? c.substr(1) : c)),
3657     "POST" === t && (d["content-type"] = d["content-type"] ? d["content-type"] : "application
3658     g && !d[a.c.tokenName] && (d[a.c.tokenName] = Object(r.a){"token"} || ""),
3659     d["Private-Platform"] = f ? "pub" : "h5",
3660     d["Private-Version"] = "1.9.1",
3661     s({
3662       "method": t,
3663       "url": y,
3664       "data": h ? Q(u || {}) : u,
3665       "headers": d
3666     }).then(function(e) {
3667       return A ? void A(e) : 200 === e.data.code || 0 === e.data.code ? (Object(o.a)(),
3668       void n(e)) : (Object(o.a)(),
3669       function responseHandle(e) {
3670         var t = e.data.code;
```



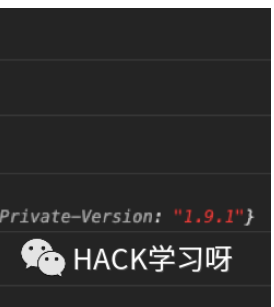
7. 逐个往下看。在“anonymous”处看到如下：

```
3637     var s = b.a.create({}) s = /wrap()
3638     , c = e.url c = "user-service/user/login"
3639     , u = e.data u = {mobile: "13612341234", password: "123333", zone: 86}
3640     , l = e.header l = undefined
3641     , d = void 0 === l ? {} : l d = {content-type: "application/json", Authorized-Token: "", Private-Platform: "h5", Private-Ver
3642     , p = e.blur p = undefined
3643     , h = void 0 === p || p.h = true
3644     , m = e.login m = undefined
3645     , g = void 0 === m || m.g = true
3646     , A = e.onHandle; A = undefined
3647     e.consoleName;
3648     A || Object(o.c)({ A = undefined
3649     "title": "加载中...",
3650     "mask": !0
3651     });
3652     var y = c y = "https://gmall.gojoy.cn/user-service/user/login", c = "user-service/user/login"
3653     , T = a.c.requestUrlProd T = "https://gmall.gojoy.cn/"
3654     , S = Object(r.a){"host": S = ""
3655     S && (T = S), T = "https://gmall.gojoy.cn/"
3656     -1 === c.indexOf("http") && (y = T + ("/" === c[0] ? c.substr(1) : c)), c = "user-service/user/login", y = "https://gmall.gojoy
3657     "POST" === t && (d["content-type"] = d["content-type"] ? d["content-type"] : "application/json"), d = {content-type: "applica
3658     g && !d[a.c.tokenName] && (d[a.c.tokenName] = Object(r.a){"token"} || ""), g = true
3659     d["Private-Platform"] = f ? "pub" : "h5",
3660     d["Private-Version"] = "1.9.1",
3661     s({
3662       "method": t,
3663       "url": y,
3664       "data": h ? Q(u || {}) : u,
3665       "headers": d
3666     }).then(function(e) {
3667       return A ? void A(e) : 200 === e.data.code || 0 === e.data.code ? (Object(o.a)(),
3668       void n(e)) : (Object(o.a)(),
3669       function responseHandle(e) {
3670         var t = e.data.code;
3671         switch (t) {
3672           case 403:
3673             clearUserData(),
3674             Object(o.d)({
3675               "title": a.a[t],
```



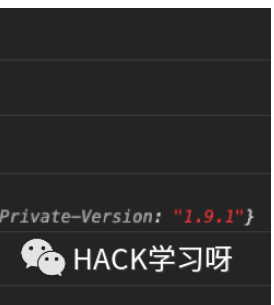
8. 在该处下断，并取消xhr处的断点。重新登录。

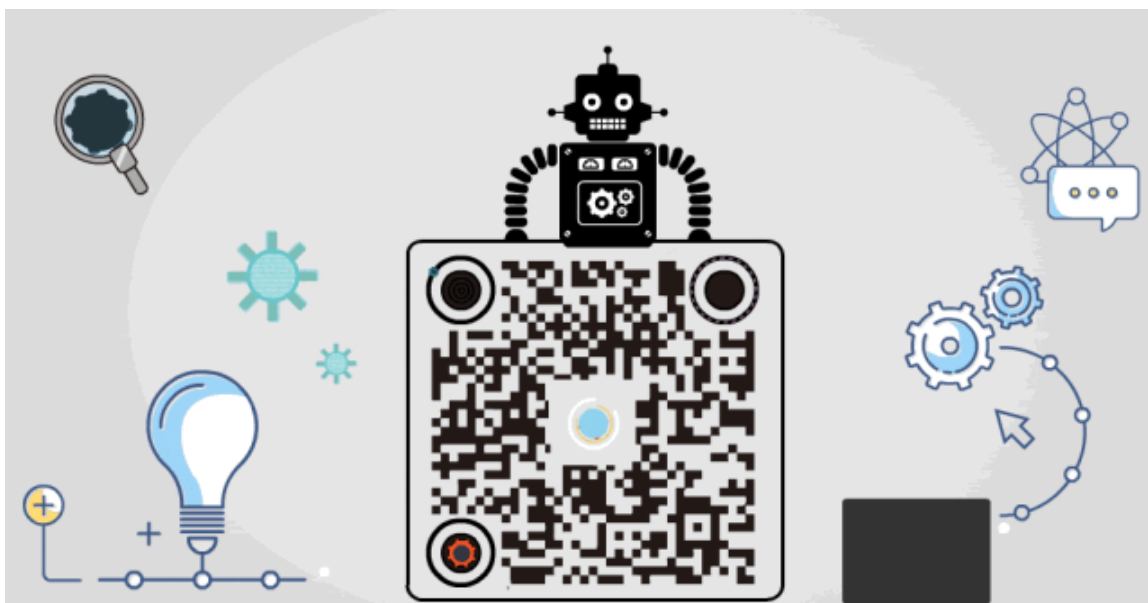
```
> y
< "https://gmall.gojoy.cn/user-service/user/login"
> h
< true
> u
< {mobile: "13612341234", password: "123333", zone: 86}
> d
< {content-type: "application/json", Authorized-Token: "", Private-Platform: "h5", Private-Version: "1.9.1"}
> Q
< f () { [native code] }
```



9. 查看相关参数值

```
> y
< "https://gmall.gojoy.cn/user-service/user/login"
> h
< true
> u
< {mobile: "13612341234", password: "123333", zone: 86}
> d
< {content-type: "application/json", Authorized-Token: "", Private-Platform: "h5", Private-Version: "1.9.1"}
> Q
< f () { [native code] }
```





精选留言

用户设置不下载评论

[阅读全文](#)