

## 一、检测点

1.清除cookie, 打script断点, 结论: 动态ob混淆格式化检测

2.静态处理, 过debugger检测:

```
1 // 在混淆代码执行之前在控制台注入
2 Function.prototype.constructor_back = Function.prototype.;
3 Function.prototype. = function() {
4     if(arguments && typeof arguments[0]=== 'string'){
5         if("debugger" === arguments[0]){
6             return '';
7         }
8     }
9     return Function.prototype.constructor_back.apply(this,arguments);
10 }
```

3.过Regex格式化检测:

```
1 // 在混淆代码执行之前在控制台注入
2 RegExp.prototype.test = function () {
3     return true
4 }
```

4.hook cookie参数:

```
1 // 在混淆代码执行之前在控制台注入
2 (function () {
3     var cookieVal = '';
4     Object.defineProperty(document, 'cookie', {
5         set: function (val) {
6             if (val.indexOf('sign') !== -1) {
7                 debugger;
8             }
9             console.log('Hook捕获到cookie设置->', val);
10            cookieVal = val;
11            return val;
12        }, get: function () {
13            return cookieVal;
14        }
15    });
16 })();
```

```
14     },
15   });
16 })();
```

## 二、函数入口

```
-   }
-   );
-   function _0x2843ea() {
-   var _0x1b932f = _0x137971['\x45\x65\x48' + '\x49\x44']['\x73\x70\x6c' + '\x69\x74']('\x7c'); _0x1b932f = (5) ['2', '4', '3', '1', '0']
-   var _0x5c5199 = 0x0; _0x5c5199 = 5
-   while (![]) {
-   switch (_0x1b932f[_0x5c5199++]) { _0x1b932f = (5) ['2', '4', '3', '1', '0'], _0x5c5199 = 5
-   case '\x30':
-   D_0x137971[D_$0x29f4('\x30\x78\x64\x63', '\x6b\x40\x68\x67') + '\x56\x49']D(eval, D_0x3e01d3(_0x137971['\x4d\x72\x59' + '\x56\x49']D(_0x51786c, D_
-   continue;
-   case '\x31':
-   (function() {
-   var _0x16d667 = {};
-   _0x16d667['\x55\x66\x61' + '\x53\x4a'] = _0x5b7dcd['\x68\x6b\x46' + '\x47\x53'];
-   _0x16d667['\x44\x79\x45' + '\x42\x43'] = _0x5b7dcd[$_0x29f4('\x30\x78\x31\x35\x66', '\x29\x39\x24\x30') + '\x64\x57'];
-   _0x16d667[$_0x29f4('\x30\x78\x63', '\x6b\x48\x23\x76') + '\x67\x74'] = _0x5b7dcd[$_0x29f4('\x30\x78\x62\x31', '\x63\x64\x72\x61') + '\x4d\x64'];
```

注意: 将ob混淆代码保存一份在本地(记得自己躺的坑。。。)