技术分享：几种常见的JavaScript混淆和反混淆工具分析实战

JavaScript (/lib/tag/JavaScript) 2016-03-05 13:41:34 发布

来自： http://www.freebuf.com/articles/web/97945.html (http://www.freebuf.com/articles/web/97945.html)

信息安全常被描述成一场军备竞赛，白帽与黑帽，渗透测试者与黑客，善与恶，本文将聚焦这场永无止境决斗中的一个小点。

HTML5 & JS 应用中充满着对输入进行验证/注入的问题，需要开发人员始终保持警惕。但同时还存在着另一个问题，就是应用中程序专用代码的易访问性。为了防止盗版或者至少使盗版更加困难，常会使用混淆工具对 JS 代码进行混淆。作为对立面，反混淆工具也可以将混淆过的 JS 代码进行还原。我曾经接触过双方的一些工具，下面是我的一些研究成果。

首先，下面这是我们的示例代码（取自Google Closure Compiler的 Wiki 页面）。一个完整的应用程序中代码会更加复杂，但这里足以用于实验了：

```
function displayNoteTitle(note) {
alert(note['title']);
}
var flowerNote = {};
flowerNote['title'] = "Flowers";
displayNoteTitle(flowerNote);
```

接下来，让我们来列举下要进行实验的混淆和反混淆工具，本文中会实验 4 个混淆工具和 2 个反混淆工具。

混淆工具：

- YUI Compressor (http://yui.github.io/yuicompressor/)
- Google Closure Compiler (https://developers.google.com/closure/compiler/)

- UglifyJS (https://github.com/mishoo/UglifyJS)
- JScrambler (https://jscrambler.com/en/)

反混淆工具：

- jsbeautifier.org (http://damilarefagbemi.com/experiments-in-js-obfuscation-deobfuscation-for-hacking-html5-apps-and-malware-analysis/jsbeautifier.org)
- JSDetox (http://relentless-coding.org/projects/jsdetox)

以上除了 JScrambler 是商业软件需要付费使用外，其余全部为免费软件。

缩小和混淆

下面首先让我们看看混淆工具的混淆效果如何，随后在看看反混淆工具的表现又如何。

YUI Compressor

```
function displayNoteTitle(a){alert(a.title)}var flowerNote={};flowerNote.title="Flowers";d
```

Google Closure Compiler

这个工具有优化和混淆两种类型：

简单优化：

```
function displayNoteTitle(a){alert(a.title)}var flowerNote={title:"Flowers"};displayNoteTi
```

深度优化：

```
alert("Flowers");
```

UglifyJS

同前一个工具一样，UglifyJS 也有两种层次的混淆：

默认：

```
function displayNoteTitle(e){alert(e.title)}var flowerNote={};flowerNote.title="Flowers",d
```

高级：

```
function t(t){alert(t.title)}var e={};e.title="Flowers",t(e);
```

JScrambler

```
/* Obfuscate your JavaScript at https://jscrambler.com */var g5b={'S':"A",'A':function(b){
```

那么，上面的代码是什么意思呢？显而易见，YUI Compressor，Google closure compiler 的简单优化模式和 UglifyJS 的默认模式都使用了相同的方法对 JS 代码进行缩小和混淆。缩小意味着压缩代码、减小应用程序的体积或者降低浏览器的加载时间。所有的这一切，在将变量名改为一个无意义的字符后，代码会变得难以阅读。

UglifyJS 的高级模式会进一步混淆函数名和全局变量的名称。Google closure compiler 的深度优化模式同时还会积极的删除无用代码，它追求最简。

而 JScrambler 则是另一种方式，它专注于对代码进行混淆，不仅不对代码进行缩小，反而通过增加代码数量使代码变的难以阅读。

美化和反混淆

jsbeautifier.org

正如其名字一样，这个在线工具试图将缩小后的代码变的更加具有可读性，但似乎它不会对代码进行进一步的反混淆。

YUI Compressor -> jsbeautified

```
function displayNoteTitle(e) {
    alert(e.title)
}
var flowerNote = {};
flowerNote.title = "Flowers", displayNoteTitle(flowerNote);
```

UglifyJS Toplevel -> jsbeautified:

```
function t(t) {
    alert(t.title)
}
var e = {};
e.title = "Flowers", t(e);
```

JSDetox

对 UglifyJS 高级模式的代码使用 JSDetox 似乎并不比 jsbeautifier.org 好多少，这点可以理解的，毕竟对变量/函数名进行转换这是不可逆的过程。

高级的反混淆和恶意代码检测

一般的代码混淆常用于知识产权保护，而高级的代码混淆则常会被用于隐藏 WEB 应用中的恶意代码。对恶意代码进行混淆是为了躲避杀毒软件的检测，这些代码在被混淆扩充后会难以被识别为恶意软件。Metasploit 的 Javascript 混淆器常被用于开发恶意代码，所以我们下面使用 Metasploit 的混淆器对我们的代码进行混淆（ 参考文档 (https://github.com/rapid7/metasploit-framework/wiki/How-to-obfuscate-JavaScript-in-Metasploit) ）。JSDetox 声称其具有进行反混淆 JS 代码的能力，所以下面让我们来尝试下对 Metasploit 和 JScrambler 混淆后的代码进行高级的反混淆。

Metasploit Javascript 混淆器

```
function L(t){window[String.fromCharCode(0141,0x6c,101,0162,0164)](t[String.fromCharCode(0
```

使用 JSDetox 进行反混淆

JScrambler -> JSDetoxed

```
var g5b = {
  'S': "A",
  'A': function(b) {
    flowerNote['title'] = b;
  },
  'X': "V",
  'o': (function(E) {
    return (function(s, p) {
      return (function(G) {
        return {
          K: G
};
      })(function(m) {
        var c, R = 0;
        for(var U = s; R < m["length"]; R++) {
          var O = p(m, R);
          c = R === 0 ? O : c ^ O;
        }
        return c ? U : !U;
      });
    })((function(h, n, a, M) {
      return h(E, 28) - M(n, a) > 28;
    })(parseInt, Date, (function(n) {
      return ('' + n)["substring"](1, (n + '')["length"] - 1);
    })('_getTime2'), function(n, a) {
      return new n()[a]();
    }), function(m, R) {
      var d = parseInt(m["charAt"](R), 16)["toString"](2);
      return d["charAt"](d["length"] - 1);
    });
  })('3lrno3f7c'),
  'e': 'title',
  'V': function(b) {
    x = b;
  },
  'Q': "Flowers"
};
function displayNoteTitle(b){
  alert(b[g5b.e]);
}
var flowerNote = g5b.o.K("3d3") ? { } : "Flowers";
g5b[g5b.S](g5b.Q);
displayNoteTitle(flowerNote);
g5b[g5b.X](g5b.D);
```

Metasploit -> JSDetoxed

```
function L(t){
  window["alert"](t["title"]);
}
var C = { };
C["title"] = "Flowers";
L(C);
```

尽管经过 Metasploit 混淆后的 JS 代码依旧可以躲避杀毒软件，但看起来也会轻易被 JSDetox 进行反混淆。有趣的是，看起来 JSDetox 无法反混淆 JScrambled 的代码。我不确定为什么 JSDetox 可以反混淆出 metasploit 的代码却不能反混淆出 JScrambler 的，不过我猜测是 JSDetox 专门针对 metasploit 的混淆方法做过专门的支持。另一方面，JScrambler 完全是一个黑盒，但这并不意味着 JScrambled 混淆后的 Javascript 代码不能被反混淆，也许有另一个工具专门用于或包含反混淆 JScrambled 代码功能。

*原文： damilarefagbemi (http://damilarefagbemi.com/experiments-in-js-obfuscation-deobfuscation-for-hacking-html5-apps-and-malware-analysis/) ，FB小编xiaix编译，转自须注明来自FreeBuf黑客与极客（FreeBuf.COM）

同类热门经验

1. Node.js 初体验 (/lib/view/open1326870121968.html)
2. JavaScript开发规范要求 (/lib/view/open1352263831610.html)
3. 使用拖拉操作来自定义网页界面布局并保存结果 (/lib/view/open1325064347889.html)
4. Nodejs入门学习，nodejs web开发入门，npm、express、socket配置安装、nodejs聊天室开发 (/lib/view/open1329050007640.html)
5. 利用HTML5同时上传多个文件 - resumable.js (/lib/view/open1327591300671.html)
6. nide：一个不错的Node.js开发工具IDE (/lib/view/open1325834128750.html)

相关文档 — 更多 (http://www.open-open.com/doc)

- 恶意软件分析诀窍与工具箱－对抗流氓软件的技术与利器.pdf (http://www.open-open.com/doc/view/4196d9d4e812487eb6cce7351e0d770b)
- JavaScript征途.pdf (http://www.open-open.com/doc/view/049c259f2fb0425593acff6b9db532fa)
- Ajax技术全解之一.doc (http://www.open-open.com/doc/view/40523d92cdf9438689affc7e1acf7831)
- 超实用的JavaScript代码段.pdf (http://www.open-open.com/doc/view/22c46cdaa9914f70bc42af8eff901e0c)
- 超实用的JavaScript代码段.pdf (http://www.open-open.com/doc/view/ec6204317dd64fc7b076f7320c990639)
- Professional Javascript for Web Developers 2nd Edition.pdf (http://www.open-open.com/doc/view/14bac88367c34cdf8e007aed85df066a)
- JavaScript 高级程序设计(中文版 全书).pdf (http://www.open-open.com/doc/view/21c506d593594e5daa2f2cbd7815f132)
- JavaScript设计模式 (Pro JavaScript Design Patterns).pdf (http://www.open-open.com/doc/view/66656c6552b84c5099d97b38eb1387a6)
- 深入浅出JavaScript.pdf (http://www.open-open.com/doc/view/ea3975ab0b6241b589242885f5b9c1f1)
- JavaScript教程.pdf (http://www.open-open.com/doc/view/0c19fe31cf2a4df9805bdcd776c0beaa)
- Maintainable JavaScript .pdf (http://www.open-open.com/doc/view/726c5f60027b4af89f912cdb133eb678)
- 真正的JavaScript忍者秘籍.pdf (http://www.open-open.com/doc/view/d697952596ec4682a4c835aa351fcb8c)
- 《JavaScript编程精解》迷你书.pdf (http://www.open-open.com/doc/view/2c1d4907f86c4783813ae4891ec42b48)
- [JavaScript权威指南(第6版)].JavaScript:The DefinitiveGuide.pdf (http://www.open-open.com/doc/view/b25771d26d2744d5ab9aa8a723ff5866)
- JavaScript - Web客户端脚本语言.pptx (http://www.open-open.com/doc/view/fc6aa9d6c2cf47808cd4b54542b9a3d4)
- 深入浅出之 JavaScript.doc (http://www.open-open.com/doc/view/8bf745497369475d87af4fc40a1cc13c)
- 精彩绝伦的CSS.pdf (http://www.open-open.com/doc/view/3f7d5f5b077349ffaddbe357a7c970cc)
- JavaScript 教程 (张明).pptx (http://www.open-open.com/doc/view/51c431e6d1f84153a61f8e9e5885be86)
- 基于HTML5的DojoWidget开发简介.doc (http://www.open-open.com/doc/view/10814c78f8764eec9b98f90c668323d2)
- XML入门教程(初学者用).pdf (http://www.open-open.com/doc/view/4be7676b1a544f6aa1df6086aec11c4c)

相关经验 — 更多 (http://www.open-open.com/lib)

- GitHub上整理的一些工具 (/lib/view/open1447852696713.html)
- GitHub上整理的一些资料 (/lib/view/open1416968753752.html)
- JavaScript Web 应用最佳实践分析 (/lib/view/open1463444261574.html)
- 结合个人经历总结的前端入门方法 (/lib/view/open1449542023941.html)
- Airbnb：我们的安卓客户端是如何使用 RxJava 的 (/lib/view/open1462200983733.html)
- Airbnb：我们的安卓客户端是如何使用 RxJava 的 (/lib/view/open1459932495525.html)
- 对抗假人 —— 前后端结合的 WAF (/lib/view/open1421851705906.html)
- 细数Javascript技术栈中的四种依赖注入 (/lib/view/open1456238872917.html)
- 码农周刊分类整理 (/lib/view/open1416282051852.html)
- Riot.js — 1Kb 大小的 JavaScript 的 MVP 框架 (/lib/view/open1384130466180.html)
- GitHub 优秀的 Android 开源项目 (/lib/view/open1416808977430.html)
- [译] Martin Fowler - Web 应用安全基础 (/lib/view/open1461307546378.html)
- Yeoman：Web 应用开发流程与工具 (/lib/view/open1394242080754.html)
- 100+ 超全的 web 开发工具和资源 (/lib/view/open1464676693194.html)

相关讨论 — 更多 (http://www.open-open.com/solution)

- 程序员技术练级攻略 (http://www.open-open.com/solution/view/1319276210452)
- 76个JavaScript教程资源免费下载 (http://www.open-open.com/solution/view/1372818526987)
- 关于编程学习的七点思索 (http://www.open-open.com/solution/view/1341747080025)
- 再谈JavaScript的数据类型问题 (http://www.open-open.com/solution/view/1318472797249)
- 那些年，追过的开源软件和技术 (http://www.open-open.com/solution/view/1425959150201)
- 什么是Node.js？ (http://www.open-open.com/solution/view/1318473088937)
- 优化网站加载速度的14个技巧 (http://www.open-open.com/solution/view/1423107429311)