



Javascript混淆与 反混淆研究

人工智能中心-数据抓取组

TITLE CONTENTS

目 录



js 混淆的利与弊



常见Js混淆与反混淆方法



高级Js混淆与解析方法



项目分析



一、js混淆的利与弊



Js混淆的利与弊

站在网站开发者的角度

- 1、是为了保护我们的前端代码逻辑
- 2、精简代码、加快传输

站在爬虫者的角度

- 1、增加了获取数据的难度
- 2、增加了获取数据的难度

二、常见JS混淆方法

常见JS混淆方法（一）

JS压缩

特征：

```
function safeAdd(r,t){var n=(65535&r)+(65535&t);return(r>>16)+(t>>16)+(n>>16)<<16|65535&n}function bitRotateLeft(r,t){return r<<t|r>>>32-t}function cmn(r,t,n,i,e,f){return safeAdd(bitRotateLeft(safeAdd(safeAdd(t,r),safeAdd(i,f)),e),n)}function ff(r,t,n,i,e,f,o){return cmn(t&n|~t&i,r,t,e,f,o)}function gg(r,t,n,i,e,f,o){return cmn(t&i|n&~i,r,t,e,f,o)}function hh(r,t,n,i,e,f,o){return cmn(t^i,r,t,e,f,o)}function ii(r,t,n,i,e,f,o){return cmn(n^(t|~i),r,t,e,f,o)}function binl(r,t){var n,i,e,f,o;r[t>>5]|=128<<t%32,r[14+(t+64>>>9<<4)]=t;var h=1732584193,a=-271733879,c=-1732584194,g=271733878;for(n=0;n<r.length;n+=16)i=h,e=a,f=c,o=g,a=ii(a=ii(a=ii(a=ii(a=hh(a=hh(a=hh(a=hh(a=gg(a=gg(a=gg(a=gg(a=ff(a=ff(a=ff(a=ff(a,c=ff(c,g=ff(g,h=ff(h,a,c,g,r[n],7,-680876936),a,c,r[n+1],12,-389564586),h,a,r[n+2],17,606105819),g,h,r[n+3],22,-1044525330),c=ff(c,g=ff(g,h=ff(h,a,c,g,r[n+4],7,-176418897),a,c,r[n+5],12,1200080426),h,a,r[n+6],17,-1473231341),g,h,r[n+7],22,-45705983),c=ff(c,g=ff(g,h=ff(h,a,c,g,r[n+8],7,1770035416),a,c,r[n+9],12,-1958414417),h,a,r[n+10],17,-42063),g,h,r[n+11],22,-1990404162),c=ff(c,g=ff(g,h=ff(h,a,c,g,r[n+12],7,1804603682),a,c,r[n+13],12,-40341101),h,a,r[n+14],17,-1502002290),g,h,r[n+15],22,1236535329),c=gg(c,g=gg(g,h=gg(h,a,c,g,r[n+1],5,-165796510),a,c,r[n+6],9,-1069501632),h,a,r[n+11],14,643717713),g,h,r[n],20,-373897302),c=gg(c,g
```

原理：

削减是一个从源代码中删除不必要的字符的技术使它看起来简单而整洁。这种技术也被称为代码压缩和最小化

常见JS混淆方法（一）

常用压缩工具：

<https://javascript-minifier.com/>

破解方法-代码格式化：

<http://tool.oschina.net/codeformat/js/>

常见JS混淆方法（二）

Base62混淆

特征：

最明显的特征是生成的代码以eval(function(p, a, c, k, e, r))开头

```
eval(function(p,a,c,k,e,r){e=function(c){return(c<a?'':e(parseInt(c/a)))+((c=c%a)>35?String.fromCharCode(c+29):c.toString(36))};if(!''.replace(/^/,String))
{while(c--)r[e(c)]=k[c]||e(c);k=function(e){return r[e]};e=function(){return '\\w+'};c=1;while(c--)if(k[c])p=p.replace(new RegExp('\\b'+e(c)+'\\b','g'),k[c]);return p}('f u(x,y){e 0=(x&P)+(y&P);e 1g=(x>>16)+(y>>16)+(0>>16);h(1g<<16)|(0&P)}f 1h(Q,R){h(Q<<R)|(Q>>>(32-R))}f C(q,a,b,x,s,t){h u(1h(u(
a,q),u(x,t)),s),b)}f j(a,b,c,d,x,s,t){h C((b&c)|((~b)&d),a,b,x,s,t)}f k(a,b,c,d,x,s,t){h C((b&d)|(c&(~d)),a,b,x,s,t)}f l(a,b,c,d,x,s,t){h C(
b^c^d,a,b,x,s,t)}f m(a,b,c,d,x,s,t){h C(c^(b|(~d)),a,b,x,s,t)}f D(x,w){x[w>>5]|=1K<<(w%32);x[(((w+1L)>>>9)<<4)+14]=w;e i;e S;e T;e U;e V;e a=1M;e
b=-1N;e c=-10;e d=1P;v(i=0;i<x.n;i+=16){S=a;T=b;U=c;V=d;a=j(a,b,c,d,x[i],7,-1Q);d=j(d,a,b,c,x[i+1],12,-1R);c=j(c,d,a,b,x[i+2],17,1S);b=j(b,c,d,a,x[
i+3],22,-1T);a=j(a,b,c,d,x[i+4],7,-1U);d=j(d,a,b,c,x[i+5],12,1V);c=j(c,d,a,b,x[i+6],17,-1W);b=j(b,c,d,a,x[i+7],22,-1X);a=j(a,b,c,d,x[i+8],7,1Y);d=j(
i+9,1Z,-1A);i+10,1B,-1C);i+11,1D,-1E);i+12,1F,-1G);i+13,1H,-1I);i+14,1J,-1K);i+15,1L,-1M);i+16,1N,-1O);i+17,1P,-1Q);i+18,1R,-1S);i+19,1T,-1U);i+20,1V,-1W);i+21,1X,-1Y);i+22,1Z,-1A);i+23,1B,-1C);i+24,1D,-1E);i+25,1F,-1G);i+26,1H,-1I);i+27,1J,-1K);i+28,1L,-1M);i+29,1N,-1O);i+30,1P,-1Q);i+31,1R,-1S);i+32,1T,-1U);i+33,1V,-1W);i+34,1X,-1Y);i+35,1Z,-1A);i+36,1B,-1C);i+37,1D,-1E);i+38,1F,-1G);i+39,1H,-1I);i+40,1J,-1K);i+41,1L,-1M);i+42,1N,-1O);i+43,1P,-1Q);i+44,1R,-1S);i+45,1T,-1U);i+46,1V,-1W);i+47,1X,-1Y);i+48,1Z,-1A);i+49,1B,-1C);i+50,1D,-1E);i+51,1F,-1G);i+52,1H,-1I);i+53,1J,-1K);i+54,1L,-1M);i+55,1N,-1O);i+56,1P,-1Q);i+57,1R,-1S);i+58,1T,-1U);i+59,1V,-1W);i+60,1X,-1Y);i+61,1Z,-1A);i+62,1B,-1C);i+63,1D,-1E);i+64,1F,-1G);i+65,1H,-1I);i+66,1J,-1K);i+67,1L,-1M);i+68,1N,-1O);i+69,1P,-1Q);i+70,1R,-1S);i+71,1T,-1U);i+72,1V,-1W);i+73,1X,-1Y);i+74,1Z,-1A);i+75,1B,-1C);i+76,1D,-1E);i+77,1F,-1G);i+78,1H,-1I);i+79,1J,-1K);i+80,1L,-1M);i+81,1N,-1O);i+82,1P,-1Q);i+83,1R,-1S);i+84,1T,-1U);i+85,1V,-1W);i+86,1X,-1Y);i+87,1Z,-1A);i+88,1B,-1C);i+89,1D,-1E);i+90,1F,-1G);i+91,1H,-1I);i+92,1J,-1K);i+93,1L,-1M);i+94,1N,-1O);i+95,1P,-1Q);i+96,1R,-1S);i+97,1T,-1U);i+98,1V,-1W);i+99,1X,-1Y);i+100,1Z,-1A);i+101,1B,-1C);i+102,1D,-1E);i+103,1F,-1G);i+104,1H,-1I);i+105,1J,-1K);i+106,1L,-1M);i+107,1N,-1O);i+108,1P,-1Q);i+109,1R,-1S);i+110,1T,-1U);i+111,1V,-1W);i+112,1X,-1Y);i+113,1Z,-1A);i+114,1B,-1C);i+115,1D,-1E);i+116,1F,-1G);i+117,1H,-1I);i+118,1J,-1K);i+119,1L,-1M);i+120,1N,-1O);i+121,1P,-1Q);i+122,1R,-1S);i+123,1T,-1U);i+124,1V,-1W);i+125,1X,-1Y);i+126,1Z,-1A);i+127,1B,-1C);i+128,1D,-1E);i+129,1F,-1G);i+130,1H,-1I);i+131,1J,-1K);i+132,1L,-1M);i+133,1N,-1O);i+134,1P,-1Q);i+135,1R,-1S);i+136,1T,-1U);i+137,1V,-1W);i+138,1X,-1Y);i+139,1Z,-1A);i+140,1B,-1C);i+141,1D,-1E);i+142,1F,-1G);i+143,1H,-1I);i+144,1J,-1K);i+145,1L,-1M);i+146,1N,-1O);i+147,1P,-1Q);i+148,1R,-1S);i+149,1T,-1U);i+150,1V,-1W);i+151,1X,-1Y);i+152,1Z,-1A);i+153,1B,-1C);i+154,1D,-1E);i+155,1F,-1G);i+156,1H,-1I);i+157,1J,-1K);i+158,1L,-1M);i+159,1N,-1O);i+160,1P,-1Q);i+161,1R,-1S);i+162,1T,-1U);i+163,1V,-1W);i+164,1X,-1Y);i+165,1Z,-1A);i+166,1B,-1C);i+167,1D,-1E);i+168,1F,-1G);i+169,1H,-1I);i+170,1J,-1K);i+171,1L,-1M);i+172,1N,-1O);i+173,1P,-1Q);i+174,1R,-1S);i+175,1T,-1U);i+176,1V,-1W);i+177,1X,-1Y);i+178,1Z,-1A);i+179,1B,-1C);i+180,1D,-1E);i+181,1F,-1G);i+182,1H,-1I);i+183,1J,-1K);i+184,1L,-1M);i+185,1N,-1O);i+186,1P,-1Q);i+187,1R,-1S);i+188,1T,-1U);i+189,1V,-1W);i+190,1X,-1Y);i+191,1Z,-1A);i+192,1B,-1C);i+193,1D,-1E);i+194,1F,-1G);i+195,1H,-1I);i+196,1J,-1K);i+197,1L,-1M);i+198,1N,-1O);i+199,1P,-1Q);i+200,1R,-1S);i+201,1T,-1U);i+202,1V,-1W);i+203,1X,-1Y);i+204,1Z,-1A);i+205,1B,-1C);i+206,1D,-1E);i+207,1F,-1G);i+208,1H,-1I);i+209,1J,-1K);i+210,1L,-1M);i+211,1N,-1O);i+212,1P,-1Q);i+213,1R,-1S);i+214,1T,-1U);i+215,1V,-1W);i+216,1X,-1Y);i+217,1Z,-1A);i+218,1B,-1C);i+219,1D,-1E);i+220,1F,-1G);i+221,1H,-1I);i+222,1J,-1K);i+223,1L,-1M);i+224,1N,-1O);i+225,1P,-1Q);i+226,1R,-1S);i+227,1T,-1U);i+228,1V,-1W);i+229,1X,-1Y);i+230,1Z,-1A);i+231,1B,-1C);i+232,1D,-1E);i+233,1F,-1G);i+234,1H,-1I);i+235,1J,-1K);i+236,1L,-1M);i+237,1N,-1O);i+238,1P,-1Q);i+239,1R,-1S);i+240,1T,-1U);i+241,1V,-1W);i+242,1X,-1Y);i+243,1Z,-1A);i+244,1B,-1C);i+245,1D,-1E);i+246,1F,-1G);i+247,1H,-1I);i+248,1J,-1K);i+249,1L,-1M);i+250,1N,-1O);i+251,1P,-1Q);i+252,1R,-1S);i+253,1T,-1U);i+254,1V,-1W);i+255,1X,-1Y);i+256,1Z,-1A);i+257,1B,-1C);i+258,1D,-1E);i+259,1F,-1G);i+260,1H,-1I);i+261,1J,-1K);i+262,1L,-1M);i+263,1N,-1O);i+264,1P,-1Q);i+265,1R,-1S);i+266,1T,-1U);i+267,1V,-1W);i+268,1X,-1Y);i+269,1Z,-1A);i+270,1B,-1C);i+271,1D,-1E);i+272,1F,-1G);i+273,1H,-1I);i+274,1J,-1K);i+275,1L,-1M);i+276,1N,-1O);i+277,1P,-1Q);i+278,1R,-1S);i+279,1T,-1U);i+280,1V,-1W);i+281,1X,-1Y);i+282,1Z,-1A);i+283,1B,-1C);i+284,1D,-1E);i+285,1F,-1G);i+286,1H,-1I);i+287,1J,-1K);i+288,1L,-1M);i+289,1N,-1O);i+290,1P,-1Q);i+291,1R,-1S);i+292,1T,-1U);i+293,1V,-1W);i+294,1X,-1Y);i+295,1Z,-1A);i+296,1B,-1C);i+297,1D,-1E);i+298,1F,-1G);i+299,1H,-1I);i+300,1J,-1K);i+301,1L,-1M);i+302,1N,-1O);i+303,1P,-1Q);i+304,1R,-1S);i+305,1T,-1U);i+306,1V,-1W);i+307,1X,-1Y);i+308,1Z,-1A);i+309,1B,-1C);i+310,1D,-1E);i+311,1F,-1G);i+312,1H,-1I);i+313,1J,-1K);i+314,1L,-1M);i+315,1N,-1O);i+316,1P,-1Q);i+317,1R,-1S);i+318,1T,-1U);i+319,1V,-1W);i+320,1X,-1Y);i+321,1Z,-1A);i+322,1B,-1C);i+323,1D,-1E);i+324,1F,-1G);i+325,1H,-1I);i+326,1J,-1K);i+327,1L,-1M);i+328,1N,-1O);i+329,1P,-1Q);i+330,1R,-1S);i+331,1T,-1U);i+332,1V,-1W);i+333,1X,-1Y);i+334,1Z,-1A);i+335,1B,-1C);i+336,1D,-1E);i+337,1F,-1G);i+338,1H,-1I);i+339,1J,-1K);i+340,1L,-1M);i+341,1N,-1O);i+342,1P,-1Q);i+343,1R,-1S);i+344,1T,-1U);i+345,1V,-1W);i+346,1X,-1Y);i+347,1Z,-1A);i+348,1B,-1C);i+349,1D,-1E);i+350,1F,-1G);i+351,1H,-1I);i+352,1J,-1K);i+353,1L,-1M);i+354,1N,-1O);i+355,1P,-1Q);i+356,1R,-1S);i+357,1T,-1U);i+358,1V,-1W);i+359,1X,-1Y);i+360,1Z,-1A);i+361,1B,-1C);i+362,1D,-1E);i+363,1F,-1G);i+364,1H,-1I);i+365,1J,-1K);i+366,1L,-1M);i+367,1N,-1O);i+368,1P,-1Q);i+369,1R,-1S);i+370,1T,-1U);i+371,1V,-1W);i+372,1X,-1Y);i+373,1Z,-1A);i+374,1B,-1C);i+375,1D,-1E);i+376,1F,-1G);i+377,1H,-1I);i+378,1J,-1K);i+379,1L,-1M);i+380,1N,-1O);i+381,1P,-1Q);i+382,1R,-1S);i+383,1T,-1U);i+384,1V,-1W);i+385,1X,-1Y);i+386,1Z,-1A);i+387,1B,-1C);i+388,1D,-1E);i+389,1F,-1G);i+390,1H,-1I);i+391,1J,-1K);i+392,1L,-1M);i+393,1N,-1O);i+394,1P,-1Q);i+395,1R,-1S);i+396,1T,-1U);i+397,1V,-1W);i+398,1X,-1Y);i+399,1Z,-1A);i+400,1B,-1C);i+401,1D,-1E);i+402,1F,-1G);i+403,1H,-1I);i+404,1J,-1K);i+405,1L,-1M);i+406,1N,-1O);i+407,1P,-1Q);i+408,1R,-1S);i+409,1T,-1U);i+410,1V,-1W);i+411,1X,-1Y);i+412,1Z,-1A);i+413,1B,-1C);i+414,1D,-1E);i+415,1F,-1G);i+416,1H,-1I);i+417,1J,-1K);i+418,1L,-1M);i+419,1N,-1O);i+420,1P,-1Q);i+421,1R,-1S);i+422,1T,-1U);i+423,1V,-1W);i+424,1X,-1Y);i+425,1Z,-1A);i+426,1B,-1C);i+427,1D,-1E);i+428,1F,-1G);i+429,1H,-1I);i+430,1J,-1K);i+431,1L,-1M);i+432,1N,-1O);i+433,1P,-1Q);i+434,1R,-1S);i+435,1T,-1U);i+436,1V,-1W);i+437,1X,-1Y);i+438,1Z,-1A);i+439,1B,-1C);i+440,1D,-1E);i+441,1F,-1G);i+442,1H,-1I);i+443,1J,-1K);i+444,1L,-1M);i+445,1N,-1O);i+446,1P,-1Q);i+447,1R,-1S);i+448,1T,-1U);i+449,1V,-1W);i+450,1X,-1Y);i+451,1Z,-1A);i+452,1B,-1C);i+453,1D,-1E);i+454,1F,-1G);i+455,1H,-1I);i+456,1J,-1K);i+457,1L,-1M);i+458,1N,-1O);i+459,1P,-1Q);i+460,1R,-1S);i+461,1T,-1U);i+462,1V,-1W);i+463,1X,-1Y);i+464,1Z,-1A);i+465,1B,-1C);i+466,1D,-1E);i+467,1F,-1G);i+468,1H,-1I);i+469,1J,-1K);i+470,1L,-1M);i+471,1N,-1O);i+472,1P,-1Q);i+473,1R,-1S);i+474,1T,-1U);i+475,1V,-1W);i+476,1X,-1Y);i+477,1Z,-1A);i+478,1B,-1C);i+479,1D,-1E);i+480,1F,-1G);i+481,1H,-1I);i+482,1J,-1K);i+483,1L,-1M);i+484,1N,-1O);i+485,1P,-1Q);i+486,1R,-1S);i+487,1T,-1U);i+488,1V,-1W);i+489,1X,-1Y);i+490,1Z,-1A);i+491,1B,-1C);i+492,1D,-1E);i+493,1F,-1G);i+494,1H,-1I);i+495,1J,-1K);i+496,1L,-1M);i+497,1N,-1O);i+498,1P,-1Q);i+499,1R,-1S);i+500,1T,-1U);i+501,1V,-1W);i+502,1X,-1Y);i+503,1Z,-1A);i+504,1B,-1C);i+505,1D,-1E);i+506,1F,-1G);i+507,1H,-1I);i+508,1J,-1K);i+509,1L,-1M);i+510,1N,-1O);i+511,1P,-1Q);i+512,1R,-1S);i+513,1T,-1U);i+514,1V,-1W);i+515,1X,-1Y);i+516,1Z,-1A);i+517,1B,-1C);i+518,1D,-1E);i+519,1F,-1G);i+520,1H,-1I);i+521,1J,-1K);i+522,1L,-1M);i+523,1N,-1O);i+524,1P,-1Q);i+525,1R,-1S);i+526,1T,-1U);i+527,1V,-1W);i+528,1X,-1Y);i+529,1Z,-1A);i+530,1B,-1C);i+531,1D,-1E);i+532,1F,-1G);i+533,1H,-1I);i+534,1J,-1K);i+535,1L,-1M);i+536,1N,-1O);i+537,1P,-1Q);i+538,1R,-1S);i+539,1T,-1U);i+540,1V,-1W);i+541,1X,-1Y);i+542,1Z,-1A);i+543,1B,-1C);i+544,1D,-1E);i+545,1F,-1G);i+546,1H,-1I);i+547,1J,-1K);i+548,1L,-1M);i+549,1N,-1O);i+550,1P,-1Q);i+551,1R,-1S);i+552,1T,-1U);i+553,1V,-1W);i+554,1X,-1Y);i+555,1Z,-1A);i+556,1B,-1C);i+557,1D,-1E);i+558,1F,-1G);i+559,1H,-1I);i+560,1J,-1K);i+561,1L,-1M);i+562,1N,-1O);i+563,1P,-1Q);i+564,1R,-1S);i+565,1T,-1U);i+566,1V,-1W);i+567,1X,-1Y);i+568,1Z,-1A);i+569,1B,-1C);i+570,1D,-1E);i+571,1F,-1G);i+572,1H,-1I);i+573,1J,-1K);i+574,1L,-1M);i+575,1N,-1O);i+576,1P,-1Q);i+577,1R,-1S);i+578,1T,-1U);i+579,1V,-1W);i+580,1X,-1Y);i+581,1Z,-1A);i+582,1B,-1C);i+583,1D,-1E);i+584,1F,-1G);i+585,1H,-1I);i+586,1J,-1K);i+587,1L,-1M);i+588,1N,-1O);i+589,1P,-1Q);i+590,1R,-1S);i+591,1T,-1U);i+592,1V,-1W);i+593,1X,-1Y);i+594,1Z,-1A);i+595,1B,-1C);i+596,1D,-1E);i+597,1F,-1G);i+598,1H,-1I);i+599,1J,-1K);i+600,1L,-1M);i+601,1N,-1O);i+602,1P,-1Q);i+603,1R,-1S);i+604,1T,-1U);i+605,1V,-1W);i+606,1X,-1Y);i+607,1Z,-1A);i+608,1B,-1C);i+609,1D,-1E);i+610,1F,-1G);i+611,1H,-1I);i+612,1J,-1K);i+613,1L,-1M);i+614,1N,-1O);i+615,1P,-1Q);i+616,1R,-1S);i+617,1T,-1U);i+618,1V,-1W);i+619,1X,-1Y);i+620,1Z,-1A);i+621,1B,-1C);i+622,1D,-1E);i+623,1F,-1G);i+624,1H,-1I);i+625,1J,-1K);i+626,1L,-1M);i+627,1N,-1O);i+628,1P,-1Q);i+629,1R,-1S);i+630,1T,-1U);i+631,1V,-1W);i+632,1X,-1Y);i+633,1Z,-1A);i+634,1B,-1C);i+635,1D,-1E);i+636,1F,-1G);i+637,1H,-1I);i+638,1J,-1K);i+639,1L,-1M);i+640,1N,-1O);i+641,1P,-1Q);i+642,1R,-1S);i+643,1T,-1U);i+644,1V,-1W);i+645,1X,-1Y);i+646,1Z,-1A);i+647,1B,-1C);i+648,1D,-1E);i+649,1F,-1G);i+650,1H,-1I);i+651,1J,-1K);i+652,1L,-1M);i+653,1N,-1O);i+654,1P,-1Q);i+655,1R,-1S);i+656,1T,-1U);i+657,1V,-1W);i+658,1X,-1Y);i+659,1Z,-1A);i+660,1B,-1C);i+661,1D,-1E);i+662,1F,-1G);i+663,1H,-1I);i+664,1J,-1K);i+665,1L,-1M);i+666,1N,-1O);i+667,1P,-1Q);i+668,1R,-1S);i+669,1T,-1U);i+670,1V,-1W);i+671,1X,-1Y);i+672,1Z,-1A);i+673,1B,-1C);i+674,1D,-1E);i+675,1F,-1G);i+676,1H,-1I);i+677,1J,-1K);i+678,1L,-1M);i+679,1N,-1O);i+680,1P,-1Q);i+681,1R,-1S);i+682,1T,-1U);i+683,1V,-1W);i+684,1X,-1Y);i+685,1Z,-1A);i+686,1B,-1C);i+687,1D,-1E);i+688,1F,-1G);i+689,1H,-1I);i+690,1J,-1K);i+691,1L,-1M);i+692,1N,-1O);i+693,1P,-1Q);i+694,1R,-1S);i+695,1T,-1U);i+696,1V,-1W);i+697,1X,-1Y);i+698,1Z,-1A);i+699,1B,-1C);i+700,1D,-1E);i+701,1F,-1G);i+702,1H,-1I);i+703,1J,-1K);i+704,1L,-1M);i+705,1N,-1O);i+706,1P,-1Q);i+707,1R,-1S);i+708,1T,-1U);i+709,1V,-1W);i+710,1X,-1Y);i+711,1Z,-1A);i+712,1B,-1C);i+713,1D,-1E);i+714,1F,-1G);i+715,1H,-1I);i+716,1J,-1K);i+717,1L,-1M);i+718,1N,-1O);i+719,1P,-1Q);i+720,1R,-1S);i+721,1T,-1U);i+722,1V,-1W);i+723,1X,-1Y);i+724,1Z,-1A);i+725,1B,-1C);i+726,1D,-1E);i+727,1F,-1G);i+728,1H,-1I);i+729,1J,-1K);i+730,1L,-1M);i+731,1N,-1O);i+732,1P,-1Q);i+733,1R,-1S);i+734,1T,-1U);i+735,1V,-1W);i+736,1X,-1Y);i+737,1Z,-1A);i+738,1B,-1C);i+739,1D,-1E);i+740,1F,-1G);i+741,1H,-1I);i+742,1J,-1K);i+743,1L,-1M);i+744,1N,-1O);i+745,1P,-1Q);i+746,1R,-1S);i+747,1T,-1U);i+748,1V,-1W);i+749,1X,-1Y);i+750,1Z,-1A);i+751,1B,-1C);i+752,1D,-1E);i+753,1F,-1G);i+754,1H,-1I);i+755,1J,-1K);i+756,1L,-1M);i+757,1N,-1O);i+758,1P,-1Q);i+759,1R,-1S);i+760,1T,-1U);i+761,1V,-1W);i+762,1X,-1Y);i+763,1Z,-1A);i+764,1B,-1C);i+765,1D,-1E);i+766,1F,-1G);i+767,1H,-1I);i+768,1J,-1K);i+769,1L,-1M);i+770,1N,-1O);i+771,1P,-1Q);i+772,1R,-1S);i+773,1T,-1U);i+774,1V,-1W);i+775,1X,-1Y);i+776,1Z,-1A);i+777,1B,-1C);i+778,1D,-1E);i+779,1F,-1G);i+780,1H,-1I);i+781,1J,-1K);i+782,1L,-1M);i+783,1N,-1O);i+784,1P,-1Q);i+785,1R,-1S);i+786,1T,-1U);i+787,1V,-1W);i+788,1X,-1Y);i+789,1Z,-1A);i+790,1B,-1C);i+791,1D,-1E);i+792,1F,-1G);i+793,1H,-1I);i+794,1J,-1K);i+795,1L,-1M);i+796,1N,-1O);i+797,1P,-1Q);i+798,1R,-1S);i+799,1T,-1U);i+800,1V,-1W);i+801,1X,-1Y);i+802,1Z,-1A);i+803,1B,-1C);i+804,1D,-1E);i+805,1F,-1G);i+806,1H,-1I);i+807,1J,-1K);i+808,1L,-1M);i+809,1N,-1O);i+810,1P,-1Q);i+811,1R,-1S);i+812,1T,-1U);i+813,1V,-1W);i+814,1X,-1Y);i+815,1Z,-1A);i+816,1B,-1C);i+817,1D,-1E);i+818,1F,-1G);i+819,1H,-1I);i+820,1J,-1K);i+821,1L,-1M);i+822,1N,-1O);i+823,1P,-1Q);i+824,1R,-1S);i+825,1T,-1U);i+826,1V,-1W);i+827,1X,-1Y);i+828,1Z,-1A);i+829,1B,-1C);i+830,1D,-1E);i+831,1F,-1G);i+832,1H,-1I);i+833,1J,-1K);i+834,1L,-1M);i+835,1N,-1O);i+836,1P,-1Q);i+837,1R,-1S);i+838,1T,-1U);i+839,1V,-1W);i+840,1X,-1Y);i+841,1Z,-1A);i+842,1B,-1C);i+843,1D,-1E);i+844,1F,-1G);i+845,1H,-1I);i+846,1J,-1K);i+847,1L,-1M);i+848,1N,-1O);i+849,1P,-1Q);i+850,1R,-1S);i+851,1T,-1U);i+852,1V,-1W);i+853,1X,-1Y);i+854,1Z,-1A);i+855,1B,-1C);i+856,1D,-1E);i+857,1F,-1G);i+858,1H,-1I);i+859,1J,-1K);i+860,1L,-1M);i+861,1N,-1O);i+862,1P,-1Q);i+863,1R,-1S);i+864,1T,-1U);i+865,1V,-1W);i+866,1X,-1Y);i+867,1Z,-1A);i+868,1B,-1C);i+869,1D,-1E);i+870,1F,-1G);i+871,1H,-1I);i+872,1J,-1K);i+873,1L,-1M);i+874,1N,-1O);i+875,1P,-1Q);i+876,1R,-1S);i+877,1T,-1U);i+878,1V,-1W);i+879,1X,-1Y);i+880,1Z,-1A);i+881,1B,-1C);i+882,1D,-1E);i+883,1F,-1G);i+884,1H,-1I);i+885,1J,-1K);i+886,1L,-1M);i+887,1N,-1O);i+888,1P,-1Q);i+889,1R,-1S);i+890,1T,-1U);i+891,1V,-1W);i+892,1X,-1Y);i+893,1Z,-1A);i+894,1B,-1C);i+895,1D,-1E);i+896,1F,-1G);i+897,1H,-1I);i+898,1J,-1K);i+899,1L,-1M);i+900,1N,-1O);i+901,1P,-1Q);i+902,1R,-1S);i+903,1T,-1U);i+904,1V,-1W);i+905,1X,-1Y);i+906,1Z,-1A);i+907,1B,-1C);i+908,1D,-1E);i+909,1F,-1G);i+910,1H,-1I);i+911,1J,-1K);i+912,1L,-1M);i+913,1N,-1O);i+914,1P,-1Q);i+915,1R,-1S);i+916,1T,-1U);i+917,1V,-1W);i+918,1X,-1Y);i+919,1Z,-1A);i+920,1B,-1C);i+921,1D,-1E);i+922,1F,-1G);i+923,1H,-1I);i+924,1J,-1K);i+925,1L,-1M);i+926,1N,-1O);i+927,1P,-1Q);i+928,1R,-1S);i+929,1T,-1U);i+930,1V,-1W);i+931,1X,-1Y);i+932,1Z,-1A);i+933,1B,-1C);i+934,1D,-1E);i+935,1F,-1G);i+936,1H,-1I);i+937,1J,-1K);i+938,1L,-1M);i+939,1N,-1O);i+940,1P,-1Q);i+941,1R,-1S);i+942,1T,-1U);i+943,1V,-1W);i+944,1X,-1Y);i+945,1Z,-1A);i+946,1B,-1C);i+947,1D,-1E);i+948,1F,-1G);i+949,1H,-1I);i+950,1J,-1K);i+951,1L,-1M);i+952,1N,-1O);i+953,1P,-1Q);i+954,1R,-1S);i+955,1T,-1U);i+956,1V,-1W);i+957,1X,-1Y);i+958,1Z,-1A);i+959,1B,-1C);i+960,1D,-1E);i+961,1F,-1G);i+962,1H,-1I);i+963,1J,-1K);i+964,1L,-1M);i+965,1N,-1O);i+966,1P,-1Q);i+967,1R,-1S);i+968,1T,-1U);i+969,1V,-1W);i+970,1X,-1Y);i+971,1Z,-1A);i+972,1B,-1C);i+973,1D,-1E);i+974,1F,-1G);i+975,1H,-1I);i+976,1J,-1K);i+977,1L,-1M);i+978,1N,-1O);i+979,1P,-1Q
```


常见JS混淆方法（二）

原理：

这类混淆的关键思想在于将需要执行的代码进行一次编码，在执行的时候还原出浏览器可执行的合法的脚本

常用混淆工具：

<http://dean.edwards.name/packer/>

破解方法-浏览器

1. 打开 谷歌 或者 火狐 浏览器
2. 按 F12 打开控制台
3. 把代码复制进去
4. 删除开头 eval 这4个字母
5. 按回车键

常见JS混淆方法（三）

公钥加密关键字段

特征：

```
var publicKey = "MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCXC4vfS+K5OWpDTSq/wxwjHGZm"  
+ "suvmeSkVQzXMk+yfABciLKOEyEfVyf4zHl9yoSigS4k6shS2Esx4aZqPxcJjFgvt"  
+ "PxNAEARsvvFvb7hPqFWwcAOWWozTxT7yjM4x0P/bmbClvMfeWGzE/OQZFRQOQxfg"  
+ "ReeSKR9IQphr//mAdQIDAQAB";  
var encrypt = new JSEncrypt();  
    encrypt.setPublicKey(publicKey);  
    //将data数组赋给ajax对象  
var PassWords = encrypt.encrypt(password);  
var rst = {username:userName,password:PassWords};  
dmsCommon.ajaxRestRequest({  
    url: dmsCommon.getDmsPath()["sysAuth"] + "/login",  
    data: JSON.stringify(rst),  
    type: 'POST',  
    sucessCallBack: function (response) {
```

常见JS混淆方法（三）

原理：

1. 用户访问客户端，客户端向服务器请求获取一个RSA公钥以及键值key，存储在本地
2. 用户在本地公钥失效前发起登录请求，则使用已有公钥对用户密码进行加密；若已过期则执行1后再加密
3. 客户端将密文与key一起传回后台
4. 后台通过key找到缓存里面的私钥，对密文进行解密

常用混淆工具：

<http://travistidwell.com/jsencrypt/>

破解方法：

```
from Crypto.Cipher import PKCS1_v1_5 as Cipher_pkcs1v1_5
from Crypto.PublicKey import RSA
```

三、高级JS混淆方法

高级JS混淆方法（一）

变量名混淆

特征：

```
var _0x1b14=["\x6C\x65\x6E\x67\x74\x68","","\x66\x72\x6F\x6D\x43\x68\x61\x72\x43\x6F\x64\x65","\x63\x68\x61\x72\x43\x6F\x64\x65\x41\x74","\x63\x6F\x6E\x63\x61\x74","\x30\x31\x32\x33\x34\x35\x36\x37\x38\x39\x61\x62\x63\x64\x65\x66","\x63\x68\x61\x72\x41\x74","\x41\x42\x43\x44\x45\x46\x47\x48\x49\x4A\x4B\x4C\x4D\x4E\x4F\x50\x51\x52\x53\x54\x55\x56\x57\x58\x59\x5A\x61\x62\x63\x64\x65\x66\x67\x68\x69\x6A\x6B\x6C\x6D\x6E\x6F\x70\x71\x72\x73\x74\x75\x76\x77\x78\x79\x7A","\x72\x61\x6E\x64\x6F\x6D","\x66\x6C\x6F\x6F\x72","\x23","\x72\x65\x70\x6C\x61\x63\x65","\x73\x70\x6C\x69\x74","\x6A\x6F\x69\x6E","\x67\x65\x74\x54\x69\x6D\x65","\x73\x65\x74\x54\x69\x6D\x65","\x3B\x20\x65\x78\x70\x69\x72\x65\x73\x3D","\x74\x6F\x55\x54\x43\x53\x74\x72\x69\x6E\x67","\x63\x6F\x6F\x6B\x69\x65","\x3D","\x3B\x20\x70\x61\x74\x68\x3D\x2F","\x69\x6E\x64\x65\x78\x4F\x66","\x63\x6F\x6F\x6B\x69\x65\x45\x6E\x61\x62\x6C\x65\x64","\u8BF7\u4FEE\u6539\u6D4F\u89C8\u5668\u8BBE\u7F6E\x2C\u5141\u8BB8\x63\x6F\x6F\x6B\x69\x65\u7F13\u5B58","\x68\x72\x65\x66","\x70\x72\x6F\x74\x6F\x63\x6F\x6C","\x68\x74\x74\x70\x73\x3A","\x6C\x6F\x63\x61\x74\x69\x6F\x6E","\x73\x75\x62\x73\x74\x72\x69\x6E\x67","\x58\x59\x32\x6C\x35\x53\x53\x4E\x4D\x4F\x6C\x7A\x43\x75\x4F\x57\x50\x72\x56\x41\x51\x36\x4F\x38\x62\x6D\x42\x31\x30\x52\x39\x30\x35\x32\x57\x72\x75\x53\x44\x36\x2B\x58\x77\x3D","\x31\x39\x33\x31\x33\x38\x37\x37\x35\x39\x37\x32\x31\x33","\x61\x6E\x74\x69\x70\x61\x73","\x68\x74\x74\x70\x73\x3A\x2F\x2F"];function safeAdd(_0xcab3x2,_0xcab3x3){var _0xcab3x4=(_0xcab3x2& 0xFFFF)+ (_0xcab3x3& 0xFFFF);var _0xcab3x5=(_0xcab3x2>> 16)+ (_0xcab3x3>> 16)+ (_0xcab3x4>> 16);return (_0xcab3x5<< 16)| (_0xcab3x4& 0xFFFF)}function bitRotateLeft(_0xcab3x7,_0xcab3x8){return (_0xcab3x7<< _0xcab3x8)| (_0xcab3x7>>> (32- _0xcab3x8))}function cmn(_0xcab3xa,_0xcab3xb,_0xcab3xc,_0xcab3x2,_0xcab3xd,_0xcab3xe){return safeAdd(bitRotateLeft(safeAdd(safeAdd(_0xcab3xb,_0xcab3xa),safeAdd(_0xcab3x2,_0xcab3xe)),_0xcab3xd),_0xcab3xc)}function ff(_0xcab3xb,_0xcab3xc,_0xcab3x10,_0xcab3x11,_0xcab3x2,_0xcab3xd,_0xcab3xe){return cmn((~_0xcab3xc& _0xcab3x10)| ((~_0xcab3xc) & _0xcab3x11),_0xcab3xb,_0xcab3xc,_0xcab3x2,_0xcab3xd,_0xcab3xe)}function gg(_0xcab3xb,_0xcab3xc,_0xcab3x10,_0xcab3x11,_0xcab3x2,_0xcab3xd,_0xcab3xe){return cmn((~_0xcab3xc& _0xcab3x11)| (_0xcab3x10& (~_0xcab3x11)),_0xcab3xb,_0xcab3xc,_0xcab3x2,_0xcab3xd,_0xcab3xe)}
```

```
eval(function(p,a,c,k,e,d){e=function(c){return(c<a?"":e(parseInt(c/a)))+((c=c%a)>35?String.fromCharCode(c+29):c.toString(36))};if(!''.replace(/^/,String)){while(c-->)d[e(c)]=k[c]||e(c);k=[function(e){return d[e]};e=function(){return'\\w+'};c=1;while(c-->)if(k[c])p=p.replace(new RegExp('\\b'+e(c)+'\\b','g'),k[c]);return p;}('f u(x,y){e W=(x&Y)+(y&Y);e 1E=(x>>16)+(y>>16)+(W>>16);h(1E<<16)|(W&Y)}f 1D(X,Q){h(X<<Q)|(X>>>(32-Q))}f F(q,a,b,x,s,t){h u(1D(u(a,q),u(x,t)),s),b)}f j(a,b,c,d,x,s,t){h F((b&c)|((~b)&d),a,b,x,s,t)}f l(a,b,c,d,x,s,t){h F((b&d)|(c&(~d)),a,b,x,s,t)}f m(a,b,c,d,x,s,t){h F(b^c^d,a,b,x,s,t)}f k(a,b,c,d,x,s,t){h F(c^(b|(~d)),a,b,x,s,t)}f C(x,w){x[w>>5]|=1K<<(w%32);x[(((w+2t)>>>9)<<4)+14]=w;e i;e T;e P;e 1f;e 18;e a=2B;e b=-2G;e c=-2I;e d=2D;v(i=0;i<x.n;i+=16){T=a;P=b;1f=c;18=d;a=j(a,b,c,d,x[i],7,-2C);d=j(d,a,b,c,x[i+1],12,-2F);c=j(c,d,a,b,x[i+2],17,2E);b=j(b,c,d,a,x[i+3],22,-2h);a=j(a,b,c,d,x[i+4],7,-2i);d=j(d,a,b,c,x[i+5],12,2e);c=j(c,d,a,b,x[i+6],17,-2k);b=j(b,c,d,a,x[i+7],22,-2q);a=j(a,b,c,d,x[i+8],7,2p);d=j(d,a,b,c,x[i+9],12,-2m);c=j(c,d,a,b,x[i+10],17,-2l);b=j(b,c,d,a,x[i+11],22,-2o);a=j(a,b,c,d,x[i+12],7,2n);d=j(d,a,b,c,x[i+13],12,-2r);c=j(c,d,a,b,x[i+14],17,-2s);b=j(b,c,d,a,x[i+15],22,2f);a=l(a,b,c,d,x[i+1],5,-2c);d=l(d,a,b,c,x[i+6],9,-2d);c=l(c,d,a,b,x[i+11],14,2j);b=l(b,c,d,a,x[i],20,-2g);a=l(
```

高级JS混淆方法（一）

原理：

字符串字面量混淆：首先提取全部的字符串，在全局作用域创建一个字符串数组，同时转义字符增大阅读难度，然后将字符串出现的地方替换成为数组元素的引用

变量名混淆：不同于压缩器的缩短命名，此处使用了下划线加数字的格式，变量之间区分度很低，相比单个字母更难以阅读

成员运算符混淆：将点运算符替换为字符串下标形式，然后对字符串进行混淆

删除多余的空白字符：减小文件体积，这是所有压缩器都会做的事

常用混淆工具：

<http://javascriptobfuscator.com/Javascript-Obfuscator.aspx>

<http://js.5ltools.info/>

破解方法-IDE、解密工具、浏览器：

<http://jsnice.org/>

<http://js.5ltools.info/>

高级JS混淆方法（二）

使用特定符号编写js脚本

特征:

[illegible]

高级JS混淆方法（二）

原理：

jsfuck源于一门编程语言brainfuck，其主要的思想就是只使用8种特定的符号来编写代码。而jsfuck也是沿用了这个思想，它仅仅使用6种符号来编写代码。它们分别是(、)、+、[、]、!。

常用混淆工具：

<http://www.jsfuck.com/>

破解方法：

无

高级JS混淆方法（三）

引入加密密钥串

特征：

```
    }("v[x++]=<0x10>v[--x]<0x10>t.charCodeAt(b++)-32<0x10>function <0x10>return <0x10>))<0x10>++<0x10>.substr<0x10>var  
    <0x10>.length<0x10>())<0x10>,b+=<0x10>;break;case <0x10>;break}").split("<0x10>"))))('gr$Daten Nb/s!l  
    ŷy[g,(lfi~ah`{mv,-n|jqewVxp{rvmmx,&eff<0x7f>kx[lcs"l".Pq%widthl"@q&heightl"vr*getContextx$"2d[!cs#l#,*;?|u.|uc{uq$fontl#vr(  
    fillTextx$$$경2<[#c}l#2q*shadowBlurl#1q-shadowOffsetXl#$$limeq+shadowColorl#vr#arcx88802[%c}l#vr&strokex[ c}l"v,)}e0myoZB]mx[ cs!0s$1$Pb<k7l  
    l!r&lengthb%^l$1+s$j<0x02>l s#i$1ek1s$gr#tack4)zgr#tac$! +0o![#cj?o ]!l$b%s"o ]!l"l$b*b^0d#>>>s!0s%yA0s"l"l!r&lengthb<k+1"^l"1+s"j<0x05>l  
    s&l&z0l!$ +["cs\'(0l#i\'1ps9wx&s() &{s)/s(gr&Stringr,fromCharCodes)0s*yWl ._b&s o!]]l l Jb<k$.aj;l .Tb<k$.gj/l .^b<k&i"-4j!<0x1f>+&  
    s+yPo!]+s!l!l Hd>&l!l Bd>&+l!l <d>&+l!l 6d>&+l!l &+ s,y=o!o!]/q"13o!l q"10o!],l 2d>& s.{s-yMo!o!]0q"13o!]*Ld<l 4d#>>>b|s!o!l q"10o!],l!& s/  
    yIo!o!].q"13o!],o!]*Jd<l 6d#>>>b|&o!]+l &+ s0l-1!&l-1!i\'1z141z4b/@d<l"b|&+1-l(1!b^&+1-l&z1\'g,)gk}ejo{<0x7f>cm,)|yn~Lij~em["c1$b%&d<l&z1\'l $  
    +["c1$b%b|&+1-l%8d<@b|l!b^&+ q$sign ', [Object.defineProperty(e, "__esModule", {  
    value: !0  
    }))]  
});
```

高级JS混淆方法（三）

原理：

加密核心函数，保护关键代码

常用混淆工具：

无

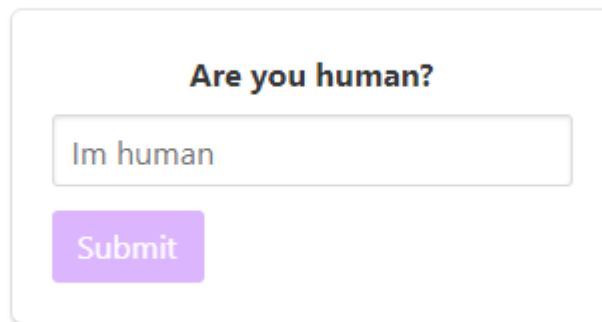
破解方法：

浏览器

高级JS混淆方法（四）

对浏览器自动化检测

<https://defyun.oss-cn-shanghai.aliyuncs.com/areyouhuman.html>



Are you human?

Submit

高级JS混淆方法（四）

原理：

通过js脚本检测环境中是否存在selenium调用浏览器参数

破解方法：

通过Linux shell命令启动浏览器

```
runBotDetection = function () {  
  var documentDetectionKeys = [  
    "__webdriver_evaluate",  
    "__selenium_evaluate",  
    "__webdriver_script_function",  
    "__webdriver_script_func",  
    "__webdriver_script_fn",  
    "__fxdriver_evaluate",  
    "__driver_unwrapped",  
    "__webdriver_unwrapped",  
    "__driver_evaluate",  
    "__selenium_unwrapped",  
    "__fxdriver_unwrapped",  
  ];  
  
  var windowDetectionKeys = [  
    "_phantom",  
    "_nightmare",  
    "_selenium",  
    "callPhantom",  
    "callSelenium",  
    "_Selenium_IDE_Recorder",  
  ];  
};
```

四、项目分析

瓜子二手车js混淆分析

路虎登录界面js混淆分析

蝉大师js混淆分析

抖音视频js混淆分析