JavaScript Obfuscation

Prasanna Kanagasabai

- **Working in Information Security for more than 10 years**
- **null Moderator for Bangalore Chapter**
- **Have a passion towards Security**
- **Penetration Tester in Day, Coder, Researcher @ night**
- **Coder of “IronSAP”**
- **Just completed OSCP**

n|u

Topics to be covered

- JavaScript
- JavaScript Obfuscation
- JavaScript D-Obfuscation Techniques

Obfuscation

Obfuscation is the concealment of intended meaning in communication, making communication confusing, intentionally ambiguous, and more difficult to interpret.
–Wikipedia definition

- Art of Hiding Execution from plain text

Example

```
<pre>
function wprcm(){   var uUHIjMJVFJET = navigator.userAgent.toLowerCase();
if(uUHIjMJVFJET.indexOf(String.fromCharCode(0157,112,0145,114,97)) != -
'Z'[720094129..toString(16<<1)+""]) {     return
String.fromCharCode(0x6d,0x61,0x54,0150,76,0114,0132,113,0x50,0155,114,0
x72,0x46,0x53);   }   if(uUHIjMJVFJET.indexOf(523090424..toString(1<<5)+"x")
!= -'c'[720094129..toString(4<<3)+""]) {     return (-~-~-
~'Nday'[720094129..toString(1<<5)+""]<(-~-
~'bp'[720094129..toString(2<<4)+""]*010+2)?(function () { var
qeNX='sG',YMkg='XfkU',PQmI='I',IuIx='oMAYc'; return
PQmI+IuIx+YMkg+qeNX })():String.fromCharCode(106,0x67,0143,120,117));   }
```

Why Create Obfuscated Code

- Bypass WAF's, filters
- Decrypt Exploit Packs
- Bypass filters (in-house and commercial)
- hide implementation details
- Social engineering payloads

Javascript

JavaScript

- Loosely Typed Language
- Gibberish Looking Data can convey valid information
- Web Depends on JS
- Mostly used in client side by recently server side impletions like node.js are becoming famous

Sample:

```
function factorial(n) { if (n === 0) { return 1; } return n * factorial(n - 1); }
```

n|u

JavaScript Strings

- “I am a normal string ” – Normal String
- ‘I am a normal string’ – Normal String
- / I am a regex string/+”” – Regex Strings
- /I am a regex string/.source – Regex Source facility
- [‘I am a String ’]+[] – Square notation to access string.

- JavaScript provides various methods to create strings
- Strings play a very major role in obfuscation
- Some implementations can be browser specific only

Operators

- JavaScript supports many infix operators: +,-,~,++,—,!,
- Plays a very active role in obfuscation

Regular Expressions (RE)

- What is Regular Expressions ?
- Browsers Support RE as function and arguments to it.
- The result is either first matched or if parentheses is used the result is stored in a array.

Comments

- // single Line comments
- /**/ is a multiline comments.
- JavaScript supports <!—> HTML comments inline in JavaScript.

Encoding

- Critical part of Obfuscation
- 3 Modes Supported :
  - 1. Unicode ======> \u0061
  - 2. Octal ======> \141
  - 3. Hex ======>\x61

Hide EVAL from the previous Slide

Hiding Eval

```
(a = {}.Valueof,
   a())['String.fromCharCode(String.fromCharCode(101,1
   18,97,108);
)']
```

Basic Obfuscation !!!

JavaScript Variables

- variables can be used to store values
- Can be defined with or without “var”
- 1. Alphanumeric characters
- 2. numbers except the first character
- 3. _ and $
- 4. Unicode characters

JavaScript Variables

- JS allows various methods to create JavaScript variables:

- x = "string";
- (x)=('string');
- this.x='string';
- x ={'a':'string'}.a;
- [x,y,z]=['str1','str2','str3'];
- x=/z(.*)/('zstring')[1];x='string';
- x=1?'string':0

Built Variables

- Essential to interact with browser objects like:

- Document – Get Access to DOM, URL,Cookies
- Name – Sets property name from parent window.
- Location.hash
- The URL variable

Alpha-Numeric JS

(+[][+[]]+[])[++[[]][+[]]]

- Would you believe this is JavaScript

Alpha Numeric JS

- Creating a JavaScript Snippet Without any Alphanumeric characters

(+[][+[]]+[])[++[[]][+[]]] = “a”

Detailed steps :

1. +[] = 0
2. [+[]] = 0 inside object accessor
3. [] [+[]] = Create a blank Array with trying to 0 which creates error ‘undefined’

Alpha Numeric JS

4. +[] [+[]] = We use infix operator + to perform a mathematical operation on result of previous operation which results a error NaN (Not a Number)

We now have to extract the middle ‘a’ from the result:

1. (+[] [+[]] +[]) = Nan in string

2.++[[]] [+[]] = 1 (quirk by oxotonick)

3. (+[][+[]]+[])[++[[]][+[]]] = ‘a’

Manual De-obfuscation

Obfuscated Code:

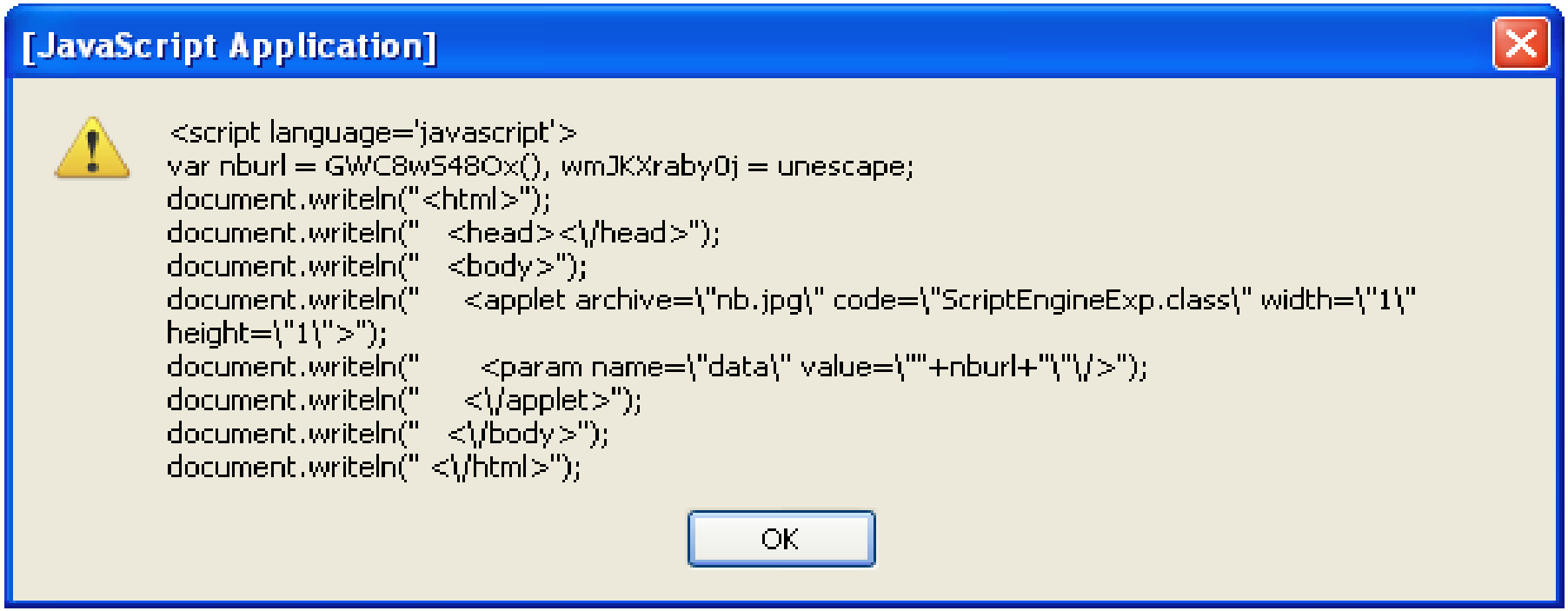
```
<pre>
function wprcm(){   var uUHIjMJVFJET = navigator.userAgent.toLowerCase();
if(uUHIjMJVFJET.indexOf(String.fromCharCode(0157,112,0145,114,97)) != -
'Z'[720094129..toString(16<<1)+""]) {     return
String.fromCharCode(0x6d,0x61,0x54,0150,76,0114,0132,113,0x50,0155,114,0
x72,0x46,0x53);   }   if(uUHIjMJVFJET.indexOf(523090424..toString(1<<5)+"x")
!= -'c'[720094129..toString(4<<3)+""]) {     return (-~-~-
~'Nday'[720094129..toString(1<<5)+""]<(-~-
~'bp'[720094129..toString(2<<4)+""]*010+2)?(function () { var
qeNX='sG',YMkg='XfkU',PQmI='I',IuIx='oMAYc'; return
PQmI+IuIx+YMkg+qeNX })():String.fromCharCode(106,0x67,0143,120,117));   }
```

Identify Essential Bits of information

```
<pre>
function wprcm(){   var uUHIjMJVFJET = navigator.userAgent.toLowerCase();
if(uUHIjMJVFJET.indexOf(String.fromCharCode(0157,112,0145,114,97)) != -
'Z'[720094129..toString(16<<1)+""]) {     return
String.fromCharCode(0x6d,0x61,0x54,0150,76,0114,0132,113,0x50,0155,114,0x72,0x4
6,0x53);   }   if(uUHIjMJVFJET.indexOf(523090424..toString(1<<5)+"x") != -
'c'[720094129..toString(4<<3)+""]) {     return (-~-~-
~'Nday'[720094129..toString(1<<5)+""]<(-~-
~'bp'[720094129..toString(2<<4)+""]*010+2)?(function () { var
qeNX='sG',YMkg='XfkU',PQmI='I',lulx='oMAYc'; return PQmI+lulx+YMkg+qeNX
})():String.fromCharCode(106,0x67,0143,120,117));   }
```

Reverse Separately

- if(uUHIjMJVFJET.indexOf(String.fromCharCode(0157,112,0145,114,97)) = *if(uUHIjMJVFJET.indexOf("opera")*
- -'Z'[720094129..toString(16<<1)+""] = *-1*
- return String.fromCharCode(0x6d,0x61,0x54,0150,76,0114,0132,113,0x50,0155,114,0x72,0x46,0x53); = *return "maThLLZqPmrrFS"*



Always de-obfuscate the script by replacing “document.write” with “alert”. Same applies to “Eval”

Auto De-obfuscation

Revelo

Demo

Thanks

- I would like to the thank the following people:
- Gareth Heyes
- Mario Heiderich
- Any one if I Have missed ….

Questions

Prasanna Kanagasabai

Prasanna.in@gmail.com