Assessing Node.js Web Application Security

Vince Marcovecchio

BlackBerry Security Research Group

BlackBerry

Background

- **Member of BlackBerry's Security Research Group for the last five years**
- **Internal pentesting and vulnerability assessments**

- **Originally trained in computer engineering**

- **Three years ago a dev team approached us with a product they wrote in Node.js…**

Agenda

- **Troublesome Javascript Features**
  - Scope
  - Type conversions
  - REDoS
  - The many paths to RCE
  - Objects & arrays
- **Troublesome Node.js APIs & Modules**
  - Weak cryptography
  - DNS inconsistencies
  - Uninitialized memory
  - v8
  - process, child_process, ffi
- **Automation**
  - ESLint
  - ESLint custom rules
  - Dependencies

Troublesome Javascript Features

Scope

- **How many different scopes exist in JavaScript?**
- **There are actually three, not counting the things you can do with closures:**
  - Block scope – declared with the `let` keyword
  - Function scope – declared with the `var` keyword
  - Global scope – declared with no keyword

- **Convoluted example:**

```
function printPasswords() {
  var users = ['alice', 'bob', 'mallory'];
  var passwords = ['123', '456', '789']
  var password = 'secret';
  for (var i = 0; i < users.length; ++i) {
    var password = (i < passwords.length) ? passwords[i] : null;
    console.log('User ' + users[i] + ' has password ' + password);
  }
  console.log('Admin has password ' + password);
}
```

Scope

MELONCARD

Enjoy more Privacy online. Sign up free today @ meloncard.com

31
10 / 2011

How One Missing `var` Ruined our Launch

Well, that was a veritable [redacted] (sorry for the language). Long story short, MelonCard was featured today on TechCrunch (along with other 500Startups companies, also on VentureBeat, Forbes, ...) and everything broke all at once. Every. little. thing. We had rolled out a huge change to MelonCard over the last few days to make our site a seamless "everything just updates" look-good / feel-good product using NodeJS long-polling with a slick KnockoutJS dynamic jQuery Templates front end. We did our due diligence of manual and unit testing, mixed with a full suite of Vows for Node. All systems check, full steam ahead, right? Not so fast.

Our system in NodeJS takes input from a user describing his state, say "I am waiting on these two records to be updated," and the server (based on a timestamp check) returns back either "Your records are up to date" or

https://web.archive.org/web/20120121104949/http://blog.meloncard.com/post/12175941935/how-one-missing-var-ruined-our-launch

Type conversions

- **JavaScript is weakly typed**
- `==` **and** `!=` **operators allow for silent type conversions**

```
 [] == {};  // false
 [] != {};  // true
 [] == !{}; // true
![] == {};  // false
```

Type conversions

	true	false	1	0	-1	"true"	"false"	"1"	"0"	"-1"	""	null	undefined	Infinity	-Infinity	[]	{}	[[]]	[0]	[1]	NaN
true	■		■					■												■	
false		■		■					■		■					■		■	■		
1	■		■					■												■	
0		■		■					■		■					■		■	■		
-1					■					■											
"true"						■															
"false"							■														
"1"	■		■					■												■	
"0"		■		■					■										■		
"-1"					■					■											
""		■		■							■					■		■			
null												■	■								
undefined												■	■								
Infinity														■							
-Infinity															■						
[]		■		■							■										
{}																					
[[]]		■		■							■										
[0]		■		■					■												
[1]	■		■					■												■	
NaN																					

	true	false	1	0	-1	"true"	"false"	"1"	"0"	"-1"	""	null	undefined	Infinity	-Infinity	[]	{}	[[]]	[0]	[1]	NaN
true	■																				
false		■																			
1			■																		
0				■																	
-1					■																
"true"						■															
"false"							■														
"1"								■													
"0"									■												
"-1"										■											
""											■										
null												■									
undefined													■								
Infinity														■							
-Infinity															■						
[]																					
{}																					
[[]]																					
[0]																					
[1]																					
NaN																					

https://dorey.github.io/JavaScript-Equality-Table/

Type conversions

- **Unexpected types don't always get handled intuitively**

```
/'/.test("normal input");     // false
/'/.test("evil'input");       // true
/'/.test({0: "evil'input"}); // false
```

REDoS

- **A regular expression whose execution time is exponentially related to the length its input**
- **The regex itself needs to be vulnerable & it needs to be sent “bad” input**
- **About as ubiquitous as buffer overflows are in C**

```
/(a+)+$/.test('aaaaaaaaaaaaaaaaaaaaaaaaaaaaa!');     // 3539 ms
/(a+)+$/.test('aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa!');    // 7155 ms
/(a+)+$/.test('aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa!');   // 14072 ms
/(a+)+$/.test('aaaaaaaaaaaaaaaaaaaaaaaaaaaaaaaa!');  // 28229 ms
```

- **Detection:** `safe-regex` **will catch most vulnerable regexes**

The many paths to RCE

- **Suppose you have a string you want to run as code**

```
const vm = require('vm');
let code = 'console.log("Am I malicious code?");';
let script = new vm.Script(code);

/* 1 */ eval(code);
/* 2 */ new Function(code)();
/* 3 */ script.runInContext(vm.createContext());
/* 4 */ script.runInNewContext();
/* 5 */ script.runInThisContext();
/* 6 */ vm.runInContext(script, vm.createContext());
/* 7 */ vm.runInDebugContext(script);
/* 8 */ vm.runInNewContext(script);
/* 9 */ vm.runInThisContext(script);
```

- `setTimeout()` **and** `setInterval()` **do not actually accept strings in Node.js**

Objects & arrays

- **`Object` prototype has many methods that can cause weird side effects:**
  - `Object.defineProperties()`
  - `Object.defineProperty()`
  - `Object.freeze()`
  - `Object.preventExtensions()`
  - `Object.seal()`

- **These same methods can be applied to arrays as well**

```
var foo = [0, 0, 0];
Object.defineProperty(foo, 0, {get: () => { return ++foo[1]; }});
foo[0]; // 1
foo[0]; // 2
foo[0]; // 3
```

Troublesome Node.js APIs & Modules

Weak cryptography

Crypto | Node.js v6.11.5

Secure | https://nodejs.org/docs/latest-v6.x/api/crypto.html

crypto.createCipher(algorithm, password)

Added in: v0.1.94

Creates and returns a `Cipher` object that uses the given `algorithm` and `password`.

The `algorithm` is dependent on OpenSSL, examples are `'aes192'`, etc. On recent OpenSSL releases, `openssl list-cipher-algorithms` will display the available cipher algorithms.

The `password` is used to derive the cipher key and initialization vector (IV). The value must be either a `'latin1'` encoded string or a `Buffer`.

The implementation of `crypto.createCipher()` derives keys using the OpenSSL function `EVP_BytesToKey` with the digest algorithm set to MD5, one iteration, and no salt. The lack of salt allows dictionary attacks as the same password always creates the same key. The low iteration count and non-cryptographically secure hash algorithm allow passwords to be tested very rapidly.

In line with OpenSSL's recommendation to use pbkdf2 instead of `EVP_BytesToKey` it is recommended that developers derive a key and IV on their own using `crypto.pbkdf2()` and to use `crypto.createCipheriv()` to create the `Cipher` object.

DNS inconsistencies

- **Do you expect these two APIs to return the same IP address?**

```
dns.lookup('example.com', callback);
dns.resolve('example.com', callback);
```

- **Depends on:**
  - What's in your /etc/hosts
  - resolv.conf
  - nsswitch.conf

Uninitialized memory

- **Deprecated buffer constructors**

```
let b = new Buffer(x);
let b = new SlowBuffer(x);
```

- **Newer buffer constructors**

```
let b = Buffer.allocUnsafe(size);
let b = Buffer.allocUnsafeSlow(size);
```

v8

- `v8.setFlagsFromString(x)` **lets you change the runtime configuration of v8**

- **“This method should be used with care. Changing settings after the VM has started may result in unpredictable behavior, including crashes and data loss; or it may simply do nothing.”**

- **Flags to enable/disable optimizations & debug tools**
- **Dangerous-looking flags**
  - `--use_strict`
    `(enforce strict mode)`
  - `--allow_unsafe_function_constructor`
    `(allow invoking the function constructor without security checks)`

Modules

- **process**
  - Core module
  - Can change UID, GID
  - Can send signals
- **child_process**
  - Core module
  - Can execute other binaries
- **ffi**
  - Dynamic binding with native libraries

Automation

ESLint

- **Javascript linter based around the Esprima lexer/parser**

- **Useful development rules**

```
no-var
strict
semi
```

- **Useful assessment rules**

```
no-eval
eqeqeq
block-scoped-var
```

ESLint custom rules

- **Blacklisting specific functions**

```
module.exports = function(context) {
  return {
    CallExpression: function(node) {
      if (node.callee.type === 'MemberExpression' &&
          context.options.includes(node.callee.object.name + '.' + node.callee.property.name)) {

        context.report(node, "'" + identifier + "' function is restricted from being called");

      } else if (node.callee.type === 'Identifier' &&
                 context.options.includes(node.callee.name)) {

        context.report(node, "'" + node.callee.name + "' function is restricted from being called");

      }
    }
  };
};
```

ESLint custom rules

- **Checking for DNS consistency**

```
var dnsConsistency = '';
module.exports = function(context) {
  return {
    CallExpression: function(node) {

      if (node.callee.type === 'MemberExpression' &&
          node.callee.object.name === 'dns' &&
          (node.callee.property.name === 'lookup' || node.callee.property.name.startsWith('resolve')
           || node.callee.property.name === 'reverse')) {

         var dnsMethod = (node.callee.property.name === 'lookup') ? 'lookup' : 'resolve';

         if (dnsConsistency === '') {
           dnsConsistency = dnsMethod;
         } else if (dnsConsistency !== dnsMethod) {
           context.report(node, "'dns." + dnsMethod + "' used when 'dns." +
             dnsConsistency + "' was used previously");
         }

       }
    }
  };
};
```

Dependencies

```
tvw-website@1.0.0
+-- async@2.1.4
| +-- lodash@4.17.4
+-- body-parser@1.15.2
| +-- bytes@2.4.0
| +-- content-type@1.0.2
| +-- debug@2.2.0
| | +-- ms@0.7.1
| +-- depd@1.1.0
| +-- http-errors@1.5.1
| | +-- inherits@2.0.3
| | +-- setprototypeof@1.0.2
| | +-- statuses@1.3.1
| +-- iconv-lite@0.4.13
| +-- on-finished@2.3.0
| | +-- ee-first@1.1.1
| +-- qs@6.2.0
| +-- raw-body@2.1.7
| | +-- unpipe@1.0.0
| +-- type-is@1.6.14
| | +-- media-typer@0.3.0
| | +-- mime-types@2.1.13
| | | +-- mime-db@1.25.0
+-- cookie-parser@1.4.3
| +-- cookie@0.3.1
| +-- cookie-signature@1.0.6
+-- ejs@2.5.5
+-- express@4.14.0
| +-- accepts@1.3.3
| | +-- mime-types@2.1.13
| | | +-- mime-db@1.25.0
| | +-- negotiator@0.6.1
| +-- array-flatten@1.1.1
| +-- content-disposition@0.5.1
| +-- content-type@1.0.2
| +-- cookie@0.3.1
| +-- cookie-signature@1.0.6
| +-- debug@2.2.0
| | +-- ms@0.7.1
| +-- depd@1.1.0
| +-- encodeurl@1.0.1
| +-- escape-html@1.0.3
| +-- etag@1.7.0
| +-- finalhandler@0.5.0
| | +-- statuses@1.3.1
| | +-- unpipe@1.0.0
| +-- fresh@0.3.0
| +-- merge-descriptors@1.0.1
| +-- methods@1.1.2
| +-- on-finished@2.3.0
| | +-- ee-first@1.1.1
| +-- parseurl@1.3.1
| +-- path-to-regexp@0.1.7
| +-- proxy-addr@1.1.2
| | +-- forwarded@0.1.0
| | +-- ipaddr.js@1.1.1
| +-- qs@6.2.0
| +-- range-parser@1.2.0
| +-- send@0.14.1
| | +-- destroy@1.0.4
| | +-- http-errors@1.5.1
| | | +-- inherits@2.0.3
| | | +-- setprototypeof@1.0.2
| | +-- mime@1.3.4
| | +-- ms@0.7.1
| | +-- statuses@1.3.1
| +-- serve-static@1.11.1
| +-- type-is@1.6.14
| | +-- media-typer@0.3.0
| | +-- mime-types@2.1.13
| | | +-- mime-db@1.25.0
| +-- utils-merge@1.0.0
| +-- vary@1.1.0
+-- helmet@3.4.0
| +-- connect@3.5.0
| | +-- debug@2.2.0
| | | +-- ms@0.7.1
| | +-- finalhandler@0.5.0
| | | +-- escape-html@1.0.3
| | | +-- on-finished@2.3.0
| | | | +-- ee-first@1.1.1
| | | +-- statuses@1.3.1
| | | +-- unpipe@1.0.0
| | +-- parseurl@1.3.1
| | +-- utils-merge@1.0.0
| +-- dns-prefetch-control@0.1.0
| +-- dont-sniff-mimetype@1.0.0
| +-- frameguard@3.0.0
| +-- helmet-csp@2.3.0
| | +-- camelize@1.0.0
| | +-- content-security-policy-builder@1.1.0
| | | +-- dashify@0.2.2
| | +-- dasherize@2.0.0
| | +-- lodash.reduce@4.6.0
| | +-- platform@1.3.3
| +-- hide-powered-by@1.0.0
| +-- hpkp@2.0.0
| +-- hsts@2.0.0
| | +-- core-util-is@1.0.2
| +-- ienoopen@1.0.0
| +-- nocache@2.0.0
| +-- referrer-policy@1.1.0
| +-- x-xss-protection@1.0.0
+-- morgan@1.7.0
| +-- basic-auth@1.0.4
| +-- debug@2.2.0
| | +-- ms@0.7.1
| +-- depd@1.1.0
| +-- on-finished@2.3.0
| | +-- ee-first@1.1.1
| +-- on-headers@1.0.1
+-- sqlite3@3.1.8
| +-- nan@2.4.0
| +-- node-pre-gyp@0.6.31
| | +-- mkdirp@0.5.1
| | | +-- minimist@0.0.8
+-- nopt@3.0.6
| +-- abbrev@1.0.9
+-- npmlog@4.0.0
| +-- are-we-there-yet@1.1.2
| | +-- delegates@1.0.0
| | +-- readable-stream@2.1.5
| | | +-- buffer-shims@1.0.0
| | | +-- core-util-is@1.0.2
| | | +-- inherits@2.0.3
| | | +-- isarray@1.0.0
| | | +-- process-nextick-args@1.0.7
| | | +-- string_decoder@0.10.31
| | | +-- util-deprecate@1.0.2
| +-- console-control-strings@1.1.0
| +-- gauge@2.6.0
| | +-- aproba@1.0.4
| | +-- has-color@0.1.7
| | +-- has-unicode@2.0.1
| | +-- object-assign@4.1.0
| | +-- signal-exit@3.0.1
| | +-- string-width@1.0.2
| | | +-- code-point-at@1.0.1
| | | | +-- number-is-nan@1.0.1
| | | +-- is-fullwidth-code-point@1.0.0
| | |   +-- number-is-nan@1.0.1
| | +-- strip-ansi@3.0.1
| | | +-- ansi-regex@2.0.0
| | +-- wide-align@1.1.0
| +-- set-blocking@2.0.0
+-- rc@1.1.6
| +-- deep-extend@0.4.1
| +-- ini@1.3.4
| +-- minimist@1.2.0
| +-- strip-json-comments@1.0.4
+-- request@2.76.0
| +-- aws-sign2@0.6.0
| +-- aws4@1.5.0
| +-- caseless@0.11.0
| +-- combined-stream@1.0.5
| | +-- delayed-stream@1.0.0
| +-- extend@3.0.0
| +-- forever-agent@0.6.1
| +-- form-data@2.1.1
| | +-- asynckit@0.4.0
| +-- har-validator@2.0.6
| | +-- chalk@1.1.3
| | | +-- ansi-styles@2.2.1
| | | +-- escape-string-regexp@1.0.5
| | | +-- has-ansi@2.0.0
| | | | +-- ansi-regex@2.0.0
| | | +-- strip-ansi@3.0.1
| | | | +-- ansi-regex@2.0.0
| | | +-- supports-color@2.0.0
| | +-- commander@2.9.0
| | | +-- graceful-readlink@1.0.1
| | +-- is-my-json-valid@2.15.0
| | | +-- generate-function@2.0.0
| | | +-- generate-object-property@1.2.0
| | | | +-- is-property@1.0.2
| | | +-- jsonpointer@4.0.0
| | | +-- xtend@4.0.1
| | +-- pinkie-promise@2.0.1
| | |   +-- pinkie@2.0.4
| +-- hawk@3.1.3
| | +-- boom@2.10.1
| | +-- cryptiles@2.0.5
| | +-- hoek@2.16.3
| | +-- sntp@1.0.9
| +-- http-signature@1.1.1
| | +-- assert-plus@0.2.0
| | +-- jsprim@1.3.1
| | | +-- extsprintf@1.0.2
| | | +-- json-schema@0.2.3
| | | +-- verror@1.3.6
| | +-- sshpk@1.10.1
| | |   +-- asn1@0.2.3
| | |   +-- assert-plus@1.0.0
| | |   +-- bcrypt-pbkdf@1.0.0
| | |   +-- dashdash@1.14.0
| | |   +-- ecc-jsbn@0.1.1
| | |   +-- getpass@0.1.6
| | |   +-- jodid25519@1.0.2
| | |   +-- jsbn@0.1.0
| | |   +-- tweetnacl@0.14.3
| +-- is-typedarray@1.0.0
| +-- isstream@0.1.2
| +-- json-stringify-safe@5.0.1
| +-- mime-types@2.1.12
| | +-- mime-db@1.24.0
| +-- node-uuid@1.4.7
| +-- oauth-sign@0.8.2
| +-- qs@6.3.0
| +-- stringstream@0.0.5
| +-- tough-cookie@2.3.2
| | +-- punycode@1.4.1
| +-- tunnel-agent@0.4.3
+-- rimraf@2.5.4
| +-- glob@7.1.1
| |   +-- fs.realpath@1.0.0
| |   +-- inflight@1.0.6
| |   | +-- wrappy@1.0.2
| |   +-- inherits@2.0.3
| |   +-- minimatch@3.0.3
| |   | +-- brace-expansion@1.1.6
| |   |   +-- balanced-match@0.4.2
| |   |   +-- concat-map@0.0.1
| |   +-- once@1.4.0
| |   | +-- wrappy@1.0.2
| |   +-- path-is-absolute@1.0.1
+-- semver@5.3.0
+-- tar@2.2.1
| +-- block-stream@0.0.9
| +-- fstream@1.0.10
| | +-- graceful-fs@4.1.9
| +-- inherits@2.0.3
+-- tar-pack@3.3.0
| +-- debug@2.2.0
| | +-- ms@0.7.1
| +-- fstream@1.0.10
| | +-- graceful-fs@4.1.9
| | +-- inherits@2.0.3
| +-- fstream-ignore@1.0.5
| | +-- inherits@2.0.3
| | +-- minimatch@3.0.3
| |   +-- brace-expansion@1.1.6
| |     +-- balanced-match@0.4.2
| |     +-- concat-map@0.0.1
| +-- once@1.3.3
| | +-- wrappy@1.0.2
| +-- readable-stream@2.1.5
| | +-- buffer-shims@1.0.0
| | +-- core-util-is@1.0.2
| | +-- inherits@2.0.3
| | +-- isarray@1.0.0
| | +-- process-nextick-args@1.0.7
| | +-- string_decoder@0.10.31
| | +-- util-deprecate@1.0.2
| +-- uid-number@0.0.6
npm-ls                1,1           Top npm-ls                111,18        29% npm-ls                169,11        58% npm-ls                227,15        88% npm-ls                255,0-1       Bot
```

Dependencies

- **Dependencies in Node.js grow exponentially**

- **Vulnerability feeds**
  - https://nodesecurity.io/advisories
  - https://snyk.io/vuln?packageManager=npm

- **Tools**
  - nsp to check for vulnerable modules
  - snyk.io tool to check for vulnerable modules and handle patching

Recap

- **Node.js has *many* language-specific intricacies to bear in mind during an assessment**
- **ESLint can be used to quickly point out high-risk code for further review**
- **Vulnerability scanners like nsp and snyk.io are vital to managing npm dependencies**

Thank you!

Vince Marcovecchio
vmarcovecchio@blackberry.com
@vinkmar