IT ASSIGNMENT 2

Topic – Employee Management System

Group Members – Pranjal Srivastava
Ojasvi Singh Chauhan
Rishav Raj
Shrishty Dayal
Shahid Afridi

Tools and command used

- **Tool used – retireJS**

RetireJS - Scan a web app for use of vulnerable JavaScript libraries. The goal of retire.js is to help you detect use of version with known vulnerabilities.

- **Commands Used -**

First install retireJS in your application's folder by typing – ***"npm install –g retire"*** in the terminal.

When installed type the command – ***"retire"*** to scan your application for the vulnerabilities.

VULNERABILITIES

Module

- **module name:** base64-url
  version: 1.3.3
  npm page: https://www.npmjs.com/package/base64-url
- **Module Description**

  Base64 encode, decode, escape and un-escape for URL applications.

Vulnerability

Vulnerability Description

- The problem arises when a number is passed in, e.g. from user-submitted JSON-encoded data.
  The API should not propagate the already-bad Buffer issue further.
- On Node.js 6.x and below, this exposes uninitialized memory, which could contain sensitive data.
- This can be also used to cause a DoS on any Node.js version by consuming the memory when large numbers are passed on input.

Overview

- Versions prior to 0.7.1 are affected by a regular expression denial of service vulnerability when extremely long version strings are parsed.

Remediation

- Update to version 0.7.1 or later. Alternatively, apply a reasonable length limit to parsed version strings.

Module

- **module name:** mpath
  version: 0.4.1
  npm page: https://www.npmjs.com/package/mpath

Module Description

- {G,Set javascript object values using MongoDB-like path notation

Vulnerability

Vulnerability Description

- An attacker can specify a path that include the prototype object, and thus overwrite important properties on Object.prototype or add new ones.

CVE-2012-6708, bug: 11290

Current Description

- jQuery before 1.9.0 is vulnerable to Cross-site Scripting (XSS) attacks. The jQuery(strInput) function does not differentiate selectors from HTML in a reliable fashion. In vulnerable versions, jQuery determined whether the input was HTML by looking for the '<' character anywhere in the string, giving attackers more flexibility when attempting to construct a malicious payload. In fixed versions, jQuery only deems the input to be HTML if it explicitly starts with the '<' character, limiting exploitability only to attackers who can control the beginning of a string, which is far less common.

Severity – Medium

Solution(s)

- jquery-upgrade

Screenshot of Vulnerabilities

```
└ jquery 1.7.1
jquery 1.7.1 has known vulnerabilities: severity: medium; CVE: CVE-2012-6708, bug: 11290, summary: Selector interpreted as HTML; http://bugs.jquery.com/ticket
/11290 https://nvd.nist.gov/vuln/detail/CVE-2012-6708 http://research.insecurelabs.org/jquery/test/ severity: medium; issue: 2432, summary: 3rd party CORS req
uest may execute, CVE: CVE-2015-9251; https://github.com/jquery/jquery/issues/2432 http://blog.jquery.com/2016/01/08/jquery-2-2-and-1-12-released/ https://nvd
.nist.gov/vuln/detail/CVE-2015-9251 http://research.insecurelabs.org/jquery/test/ severity: low; CVE: CVE-2019-11358, summary: jQuery before 3.4.0, as used in
 Drupal, Backdrop CMS, and other products, mishandles jQuery.extend(true, {}, ...) because of Object.prototype pollution; https://blog.jquery.com/2019/04/10/j
query-3-4-0-released/ https://nvd.nist.gov/vuln/detail/CVE-2019-11358 https://github.com/jquery/jquery/commit/753d591aea698e57d6db58c9f722cd0808619b1b
node_modules\mongoose\node_modules\mpath/package.json
└ mpath 0.1.1
mpath 0.1.1 has known vulnerabilities: severity: high; summary: Prototype Pollution; https://hackerone.com/reports/390860
node_modules\mongoose\node_modules\ms/package.json
└ ms 0.1.0
ms 0.1.0 has known vulnerabilities: severity: medium; summary: Regular expression denial of service; https://nodesecurity.io/advisories/46
node_modules\express-session\node_modules\uid-safe\node_modules\base64-url/package.json
└ base64-url 1.2.1
base64-url 1.2.1 has known vulnerabilities: severity: high; summary: Out-of-bounds Read; https://hackerone.com/reports/321692
```