detect-non-literal-regexp

Open

Analyzed | ⋮ | Ignore... | Fix

sims-nosql-api/wwwroot/lib/.../jquery.validate.unobtrusive.js:349

Assistant thinks this is likely a false positive

This finding is in a third-party jQuery validation library (jquery.validate.unobtrusive.js) located in wwwroot/lib. The 'params' argument comes from validation attributes defined by developers, not direct user input. Additionally, this is client-side validation where ReDoS would only affect the user's own browser, not the server or other users, making it a low security risk.

Agree and Ignore or Disagree

RULE DETAILS

- Medium severity
- Low confidence
- Monitor
- Denial-of-Service (DoS)
- CWE-1333
- detect-non-literal-regexp

FINDING DETAILS

- 3 days ago
- is241307/simsfh_ws25_SAST
  managed-scan
- main
- 5cef085
  by puywei

Finding description | Rule description

The `regex` validation method constructs a RegExp directly from the `params` argument, which comes from user-provided HTML attributes. An attacker can craft a malicious regex pattern in the `data-val-regex-pattern` HTML attribute to trigger a ReDoS (Regular Expression Denial-of-Service) attack.

Here's how an attacker would exploit this:

1. An attacker injects a malicious regex pattern into the HTML form via the `data-val-regex-pattern` attribute: `data-val-regex-pattern="(a+)+$"`
2. When the form is submitted, jQuery Validation calls the `regex` method, passing this malicious pattern as `params`
3. The code executes `new RegExp(params).exec(value)`, which constructs a RegExp from the untrusted pattern string
4. The ReDoS-vulnerable regex causes catastrophic backtracking—for input like `aaaaaaaaaaaaaaaaab`, the regex engine burns CPU cycles exponentially, freezing the main thread
5. The browser becomes unresponsive, degrading user experience or enabling a denial-of-service attack against your application

The root cause is that `params` is treated as a trusted, hardcoded string when it's actually attacker-controlled data from HTML attributes.

Your code | Example code

/.../jquery.validate.unobtrusive.js

Line 343 source
params

Line 343
params

Line 349 Sink
new RegExp(params)

sims-nosql-api/.../jquery.validate.unobtrusive.js:343 vendored code

```
338
339  $jQval.addMethod("__dummy__", function (value, element, params) {
340      return true;
341  });
342
343  $jQval.addMethod("regex", function (value, element, params) {
344      var match;
345      if (this.optional(element)) {
346          return true;
347      }
348
349      match = new RegExp(params).exec(value);
350      return (match && (match.index === 0) && (match[0].length ===
         value.length));
351  });
352
353  $jQval.addMethod("nonalphamin", function (value, element,
     nonalphamin) {
354      var match;
355      if (nonalphamin) {
356          match = value.match(/\W/g);
357          match = match && match.length >= nonalphamin;
358      }
359      return match;
360  });
361
```

Assistant suggested fix | Rule fix

Semgrep Assistant analyzed this finding, but doesn't recommend a fix because it's likely a false positive.

Activity

New note