

CERTCC / cveClient Public









<> Code Issues 3 Pull requests 4 Actions Projects Security and quality

7 Branches 0 Tags Go to file Go to file <> Code

sei-vsarvepalli Merge pull request #46 from jgamblin/update/swagg... ✓

ac2cf61 · 2 hours ago

.github/workflows	fix: drop EOL Node versions...	yesterday
ace-builds	Ace builds updates	4 years ago
api-docs	feat: add local copies of CV...	3 hours ago
api/cve	Further updates to Issue #24	3 years ago
schema	feat: add local copies of CV...	3 hours ago
sweetalert2	chore: update SweetAlert2 fr...	2 days ago
tests	fix: remove tests for getadp/...	3 hours ago
.gitignore	feat: add test infrastructure ...	yesterday
CHANGELOG.md	CHANGELOG updated	9 months ago
INSTALL.md	Added Google API URL	4 years ago
LICENSE	CERT/CC License template	4 years ago
README.md	docs: add Node.js usage an...	yesterday
RISKS.md	Added RISKS.md according...	4 years ago
autoCompleter.js	refactor: import test function...	yesterday
certicon.ico	Added favicon	4 years ago
cveClientlib.js	Merge pull request #45 from...	6 hours ago
cveInterface.css	Small updates to cveInterface	9 months ago
	feat: add local copies of CV...	3 hours ago

cveInterface.js		
 cwe-common.json	Added cwe-common.json th...	9 months ago
 encrypt-storage.js	refactor: import test function...	yesterday
 index.html	fix: remove unsupported AD...	yesterday
 language-codes.json	Ace builds updates	4 years ago
 package-lock.json	feat: add test infrastructure ...	yesterday
 package.json	docs: add Node.js usage an...	yesterday
 schemaToForm.js	feat: make schema referenc...	yesterday
 vitest.config.mjs	fix: drop EOL Node versions...	yesterday

 **README**  License  Security



cveClient

A web-based client interface and a client library to cve-services 2.x to provide CVE JSON (5.x) vulnerability management for CVE Numbering Authorities (CNAs) and Roots.

Introduction

A JavaScript library and simple HTML interface for CVE services that is like RedHat's [cvelib](#) and [Vulnogram](#). Provides capabilities to interface with [cve-services](#) with a JavaScript client and simple HTML UI. You can just serve the index.html file from any static content area of your webserver (Apache2, NGINX, thttpd, lighttpd, Caddy).

A JavaScript encryption toolkit is also served from [encrypt-storage.js](#) file. This is a simple asymmetric encryption to protect your API Key while in `localStorage` (if *Keep me logged in* checkbox is enabled) or `sessionStorage` with PKI stored in native `indexedDB`. This provides some limited protection of your API keys. Currently cve-services expects API key for every transaction, there is no middleware providing session capability or related CSRF protection.

Demo of the UI and client library can be accessed at <https://certcc.github.io/cveClient/> or CERT's [demo site](#). Currently the allowed servers (cve-services servers) are limited with Content Security Policy headers to cve-services 2.x production site, testing site and a localhost instance.

Dependency libraries for HTML UI only.

- [jQuery - 3.5.1](#)
- [Bootstrap - 4.3.1](#)
- [Popper - 1.14.7](#)
- [SweetAlert2 - 2.11](#)
- [Bootstrap-Table - 1.19.1](#)
- [Ace Editor - 1.2.4](#)

Except for Ace Editor and SweetAlert2 library all the dependencies are served from CDN sources with sha-284 [Subsource Integrity](#) checks. There is some issue with the nightly builds, so these two libraries are served from local. Please ensure you understand their license agreements as well.

Roll out your own using cveClientlib

If you plan to rollour your own UI, this JavaScript library is helpful. Also note that there is another client library with similar capabilities built with ECMAScrip6 requirement at <https://github.com/xdr/cve.js>.

You can use the [cveClientlib.js](#) to do all the tasks being performed by the current UI. The file is called cveClientlib.js as Safari browsers have trouble with filename and a Class name being the same.

Using cveClientlib in Node.js

The `cveClientlib.js` file includes conditional exports for Node.js environments. You can use it directly in Node.js scripts (note: `fetch` is available natively in Node 18+):

```
const cveClient = require("../cveClientlib.js");

const client = new cveClient(
  "your_org_short_name",
  "your_username",
  "your_api_key",
  "https://cveawg.mitre.org/api"
```



```
);

// Get CVE details
client.getcvedetail("CVE-2024-1234").then(function(cve) {
  console.log(JSON.stringify(cve, null, 2));
});

// Reserve a CVE ID
client.reservecve(1).then(function(result) {
  console.log("Reserved:", result);
});

// Get org quota
client.getquota().then(function(quota) {
  console.log("Quota:", quota);
});
```

For older Node.js versions without native `fetch`, you can use [node-fetch](#):

```
const fetch = require("node-fetch");
globalThis.fetch = fetch;
const cveClient = require("../cveClientlib.js");
```



Running Tests

Tests use [Vitest](#) with `jsdom` and cover pure functions, security regressions, and API client behavior.

```
npm ci
npm test
```



Test suites:

- **Pure function tests** (24 tests) — `get_deep`, `set_deep`, `simpleCopy`, `checkurl`, `check_json`, `queryParser`
- **Security regression tests** (13 tests) — prototype pollution protection, XSS prevention via `safeHTML` and `cleanHTML`
- **API client tests** (14 tests) — URL construction, auth headers, CVE/ADP operations

Risks of using API keys in browser for

CVE-Services

See the [RISKS.md](#) that captures some of the inherent risks of using API keys to access an API service. If you decide to use these web base clients to access `cve-services`, please be aware of these risks.

Installation on your own webserver

Releases

No releases published

Packages

No packages published

Contributors 6



Languages

