SECURIN ASSESMENT

DATE: 2024-05-10

NAME: RITHESH S

Problem Statement

NVD - CVE API

The CVE API is used to easily retrieve information on a single CVE or a collection of CVE from the NVD. Pls refer to the below NVD CVE documentation to get more information

SOLUTION

TECH STACK

- `python flask`
- `mongodb`
- `docker`

FUNCTIONALITIES

- Retrieve CVE information from the CVE API for all the CVE's batch wise using `startIndex` and `perPage` get parameters. https://services.nvd.nist.gov/rest/json/cves/2.0 store all the CVE INFORMATION in local **MONGO DB**.
- Retrieve CVE changes from the CVE HISTORY API and update the changes in the local db
- Removed **REJECTED** vulnerabilities from the database.
- Every 30 minutes it will query the HISTORY API for any changes and apply those changes to the local database
- All data are kept up-to-date with the server in an interval of 30 minutes. can be changed.
- Pagination on the **server side** is implemented
- Multiple Search filters are implemented can be using combined
  - cve id
  - year
  - baseScore

API ENDPOINT

ENDPOINT : `http://<domain>:<port>/`

`/` and `/cve/lists`

- Lists out the CVE INFORMATION from the database in a clean table.

GET PARAMETERS

- `?page=`
  - datatype: `int`
  - returns: returns the content of the page requested
- `?limit=`
  - datatype: `int`
  - returns: restricts the number of CVES in the page
  - options: `10, 50, 100`
- `?year=`
  - datatype: `string`
  - returns: returns all the CVES in that given year
- `?lastmodified=`
  - datatype: `int`
  - returns: sorts the table with last modified in descending order and limits it to the number of days given
- `?lt=`
  - datatype: `float`
  - returns: BaseScore value lesser that the input
  - ref - metrics.cvssMetricV2.cvssData.baseScore or metrics.cvssMetricV3.cvssData.baseScore
- `?gt=`
  - datatype: `float`
  - returns: BaseScore value greater that the input
  - ref - metrics.cvssMetricV2.cvssData.baseScore or metrics.cvssMetricV3.cvssData.baseScore
- `?id=`
  - datatype: `string`
  - returns: Searches the CVE id either completely or partially. `REGEX MATCHING`

`/cve/lists/id/<id>`

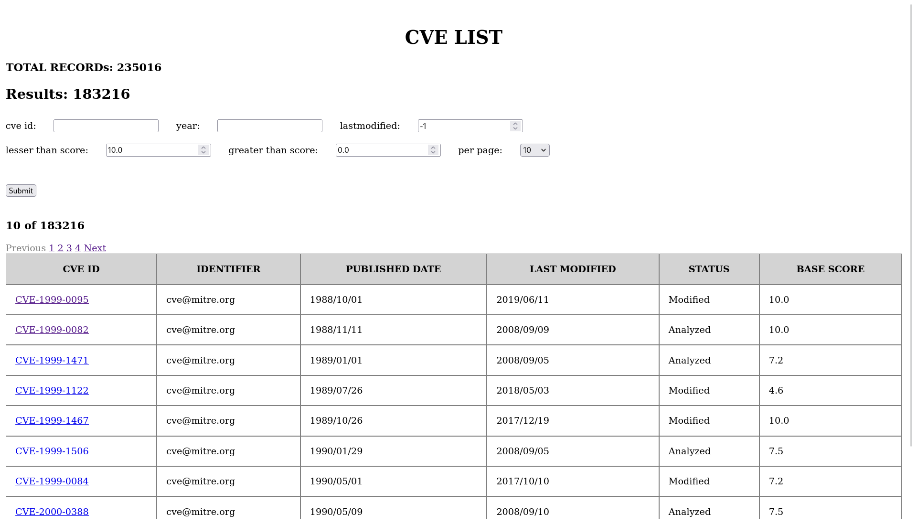
- returns the details of the specific CVE INFORMATION

UI

SEARCH

The previous requests are saved and previous values or loaded after page refresh

```
{
    "limit": limit,
    "year": year,
    "lastmodified": lastmodified,
    "lt": lt,
    "gt": gt,
```


CVE LIST

TOTAL RECORDs: 235016

Results: 183216

cve id: year: lastmodified: -1

lesser than score: 10.0 greater than score: 0.0 per page: 10

Submit

10 of 183216

Previous 1 2 3 4 Next

CVE ID	IDENTIFIER	PUBLISHED DATE	LAST MODIFIED	STATUS	BASE SCORE
CVE-1999-0095	cve@mitre.org	1988/10/01	2019/06/11	Modified	10.0
CVE-1999-0082	cve@mitre.org	1988/11/11	2008/09/09	Analyzed	10.0
CVE-1999-1471	cve@mitre.org	1989/01/01	2008/09/05	Analyzed	7.2
CVE-1999-1122	cve@mitre.org	1989/07/26	2018/05/03	Modified	4.6
CVE-1999-1467	cve@mitre.org	1989/10/26	2017/12/19	Modified	10.0
CVE-1999-1506	cve@mitre.org	1990/01/29	2008/09/05	Analyzed	7.5
CVE-1999-0084	cve@mitre.org	1990/05/01	2017/10/10	Modified	7.2
CVE-2000-0388	cve@mitre.org	1990/05/09	2008/09/10	Analyzed	7.5

Figure 1: index page

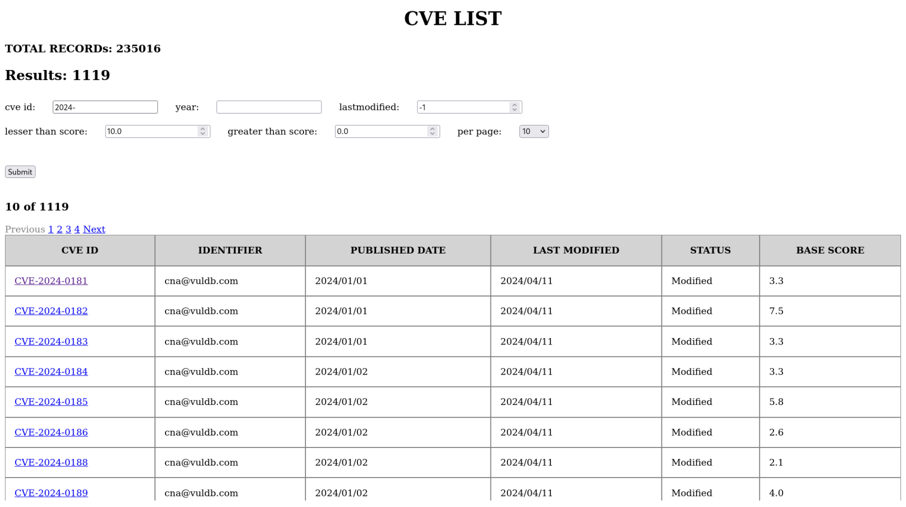
```
    "id": id
}
```

CODE IMPLEMENTATION

On first time run the `util.py` it will populate the database by querying the CVE api endpoint After the database is synced. It will delete all the `REJECTED` vulnerabilities. It will then wait for 30 minutes or specified time. to then query the history api and make changes again The last queried time is written to `config.json` and history after that time is then quried

```
payload = {
        "published": {
                "$regex": year
            },
        "$or": [
            {"metrics.cvssMetricV2.cvssData.baseScore": {"$gte": gt, "$lte": lt}},
            {"metrics.cvssMetricV3.cvssData.baseScore": {"$gte": gt, "$lte": lt}}
            ],
        "id": {
            "$regex": id
            }
        }
```

This sets the filter query paramter in the mongodb based on the input args from the `GET` parameters



CVE LIST

TOTAL RECORDs: 235016

Results: 1119

cve id: 2024- year: lastmodified: -1

lesser than score: 10.0 greater than score: 0.0 per page: 10

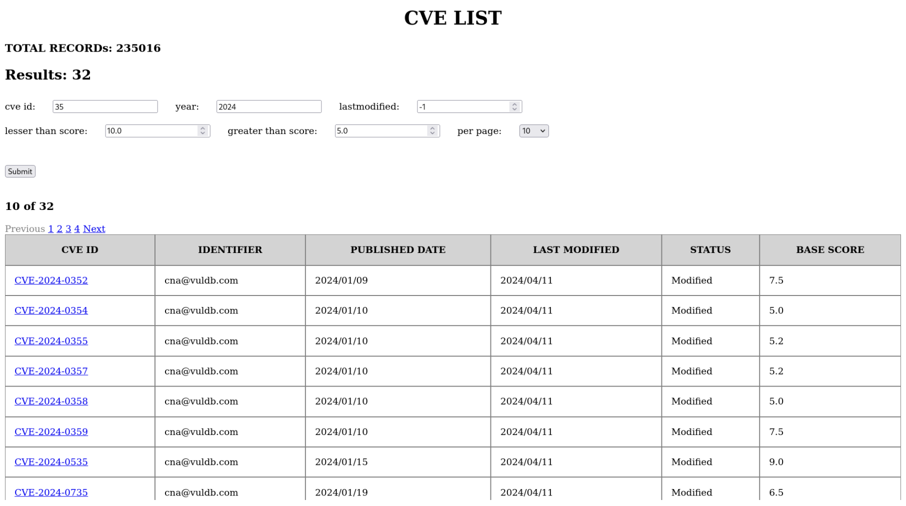
Submit

10 of 1119

Previous 1 2 3 4 Next

CVE ID	IDENTIFIER	PUBLISHED DATE	LAST MODIFIED	STATUS	BASE SCORE
CVE-2024-0181	cna@vuldb.com	2024/01/01	2024/04/11	Modified	3.3
CVE-2024-0182	cna@vuldb.com	2024/01/01	2024/04/11	Modified	7.5
CVE-2024-0183	cna@vuldb.com	2024/01/01	2024/04/11	Modified	3.3
CVE-2024-0184	cna@vuldb.com	2024/01/02	2024/04/11	Modified	3.3
CVE-2024-0185	cna@vuldb.com	2024/01/02	2024/04/11	Modified	5.8
CVE-2024-0186	cna@vuldb.com	2024/01/02	2024/04/11	Modified	2.6
CVE-2024-0188	cna@vuldb.com	2024/01/02	2024/04/11	Modified	2.1
CVE-2024-0189	cna@vuldb.com	2024/01/02	2024/04/11	Modified	4.0

Figure 2: SEARCHED FOR ID WITH 2024-



CVE LIST

TOTAL RECORDs: 235016

Results: 32

cve id: 35 year: 2024 lastmodified: -1

lesser than score: 10.0 greater than score: 5.0 per page: 10

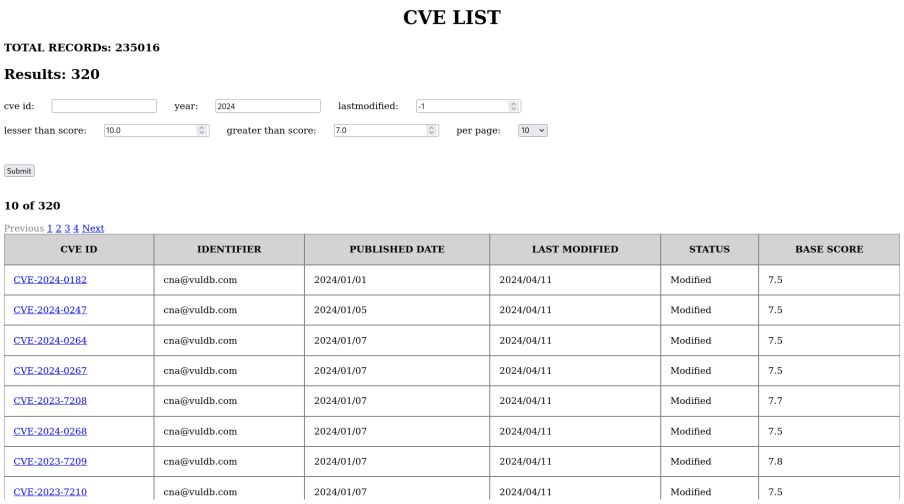
Submit

10 of 32

Previous 1 2 3 4 Next

CVE ID	IDENTIFIER	PUBLISHED DATE	LAST MODIFIED	STATUS	BASE SCORE
CVE-2024-0352	cna@vuldb.com	2024/01/09	2024/04/11	Modified	7.5
CVE-2024-0354	cna@vuldb.com	2024/01/10	2024/04/11	Modified	5.0
CVE-2024-0355	cna@vuldb.com	2024/01/10	2024/04/11	Modified	5.2
CVE-2024-0357	cna@vuldb.com	2024/01/10	2024/04/11	Modified	5.2
CVE-2024-0358	cna@vuldb.com	2024/01/10	2024/04/11	Modified	5.0
CVE-2024-0359	cna@vuldb.com	2024/01/10	2024/04/11	Modified	7.5
CVE-2024-0535	cna@vuldb.com	2024/01/15	2024/04/11	Modified	9.0
CVE-2024-0735	cna@vuldb.com	2024/01/19	2024/04/11	Modified	6.5

Figure 3: CVE ID CONTAINT 35 AND IN YEAR 2024 AND BASESCORE GREATER THAN 5.0



CVE LIST

TOTAL RECORDs: 235016

Results: 320

cve id: year: 2024 lastmodified: -1

lesser than score: 10.0 greater than score: 7.0 per page: 10

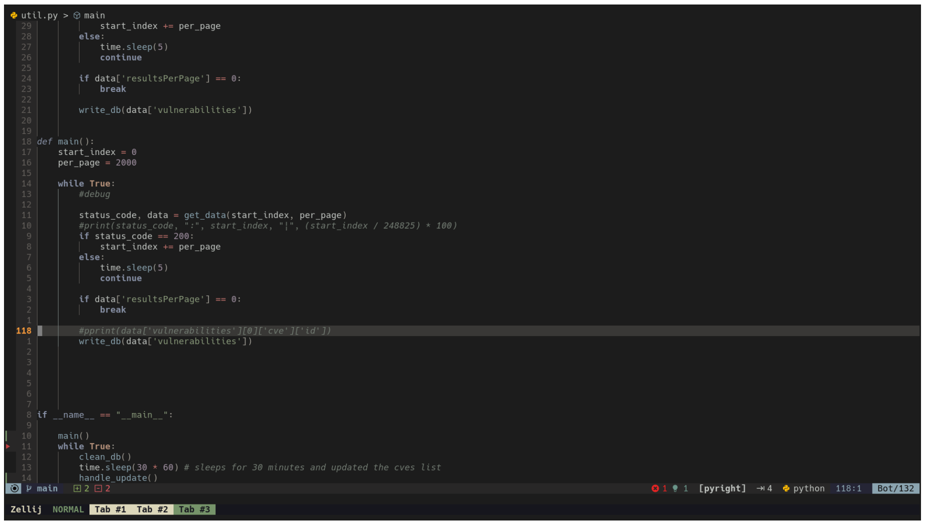
Submit

10 of 320

Previous 1 2 3 4 Next

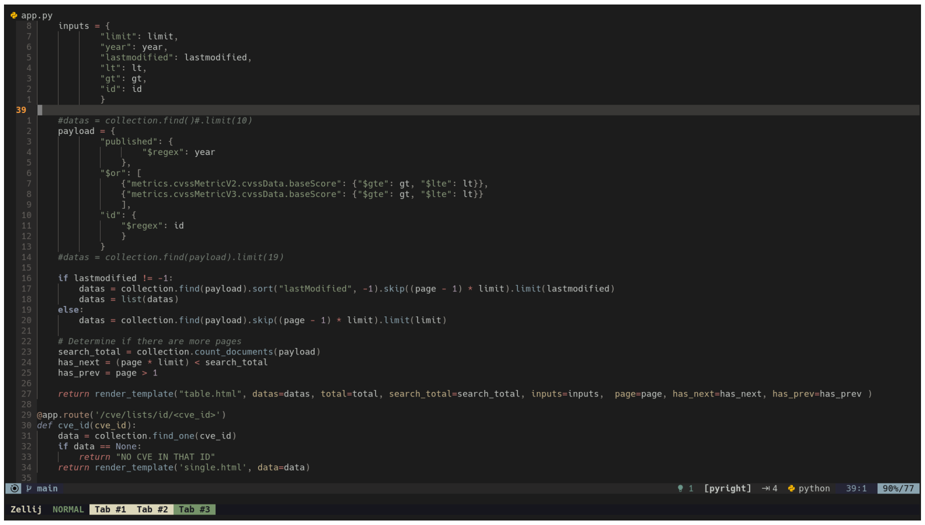
CVE ID	IDENTIFIER	PUBLISHED DATE	LAST MODIFIED	STATUS	BASE SCORE
CVE-2024-0182	cna@vuldb.com	2024/01/01	2024/04/11	Modified	7.5
CVE-2024-0247	cna@vuldb.com	2024/01/05	2024/04/11	Modified	7.5
CVE-2024-0264	cna@vuldb.com	2024/01/07	2024/04/11	Modified	7.5
CVE-2024-0267	cna@vuldb.com	2024/01/07	2024/04/11	Modified	7.5
CVE-2023-7208	cna@vuldb.com	2024/01/07	2024/04/11	Modified	7.7
CVE-2024-0268	cna@vuldb.com	2024/01/07	2024/04/11	Modified	7.5
CVE-2023-7209	cna@vuldb.com	2024/01/07	2024/04/11	Modified	7.8
CVE-2023-7210	cna@vuldb.com	2024/01/07	2024/04/11	Modified	7.5

Figure 4: PAGINATION EXAMPLE



```
util.py > main
            start_index += per_page
        else:
            time.sleep(5)
            continue

        if data['resultsPerPage'] == 0:
            break

        write_db(data['vulnerabilities'])


def main():
    start_index = 0
    per_page = 2000

    while True:
        #debug

        status_code, data = get_data(start_index, per_page)
        #print(status_code, ":", start_index, "¦", (start_index / 248825) * 100)
        if status_code == 200:
            start_index += per_page
        else:
            time.sleep(5)
            continue

        if data['resultsPerPage'] == 0:
            break

        #pprint(data['vulnerabilities'][0]['cve']['id'])
        write_db(data['vulnerabilities'])







if __name__ == "__main__":

    main()
    while True:
        clean_db()
        time.sleep(30 * 60) # sleeps for 30 minutes and updated the cves list
        handle_update()
 main   2  2                                          1  1  [pyright]  ⇥ 4  python  118:1  Bot/132

Zellij  NORMAL  Tab #1  Tab #2  Tab #3
```

Figure 5: util.py sample code



```python
    inputs = {
            "limit": limit,
            "year": year,
            "lastmodified": lastmodified,
            "lt": lt,
            "gt": gt,
            "id": id
            }

    #datas = collection.find()#.limit(10)
    payload = {
            "published": {
                    "$regex": year
                },
            "$or": [
                {"metrics.cvssMetricV2.cvssData.baseScore": {"$gte": gt, "$lte": lt}},
                {"metrics.cvssMetricV3.cvssData.baseScore": {"$gte": gt, "$lte": lt}}
                ],
            "id": {
                "$regex": id
                }
            }
    #datas = collection.find(payload).limit(19)

    if lastmodified != -1:
        datas = collection.find(payload).sort("lastModified", -1).skip((page - 1) * limit).limit(lastmodified)
        datas = list(datas)
    else:
        datas = collection.find(payload).skip((page - 1) * limit).limit(limit)

    # Determine if there are more pages
    search_total = collection.count_documents(payload)
    has_next = (page * limit) < search_total
    has_prev = page > 1

    return render_template("table.html", datas=datas, total=total, search_total=search_total, inputs=inputs,  page=page, has_next=has_next, has_prev=has_prev )

@app.route('/cve/lists/id/<cve_id>')
def cve_id(cve_id):
    data = collection.find_one(cve_id)
    if data == None:
        return "NO CVE IN THAT ID"
    return render_template('single.html', data=data)
```

```
 main                                        1  [pyright]  ⇥4  python  39:1  90%/77
Zellij  NORMAL  Tab #1  Tab #2  Tab #3
```

Figure 6: app.py sample code

WORKFLOW

LIST VIEW

You can see the list of CVES in `/` or `/cve/lists` endpoint.

You can **chain multiple search paramters** together to have a strong searching capabilities

Results per page can be configured

Results are paginated and previous and next page pagination are done server side

SINGLE VIEW

View specific details about the CVE

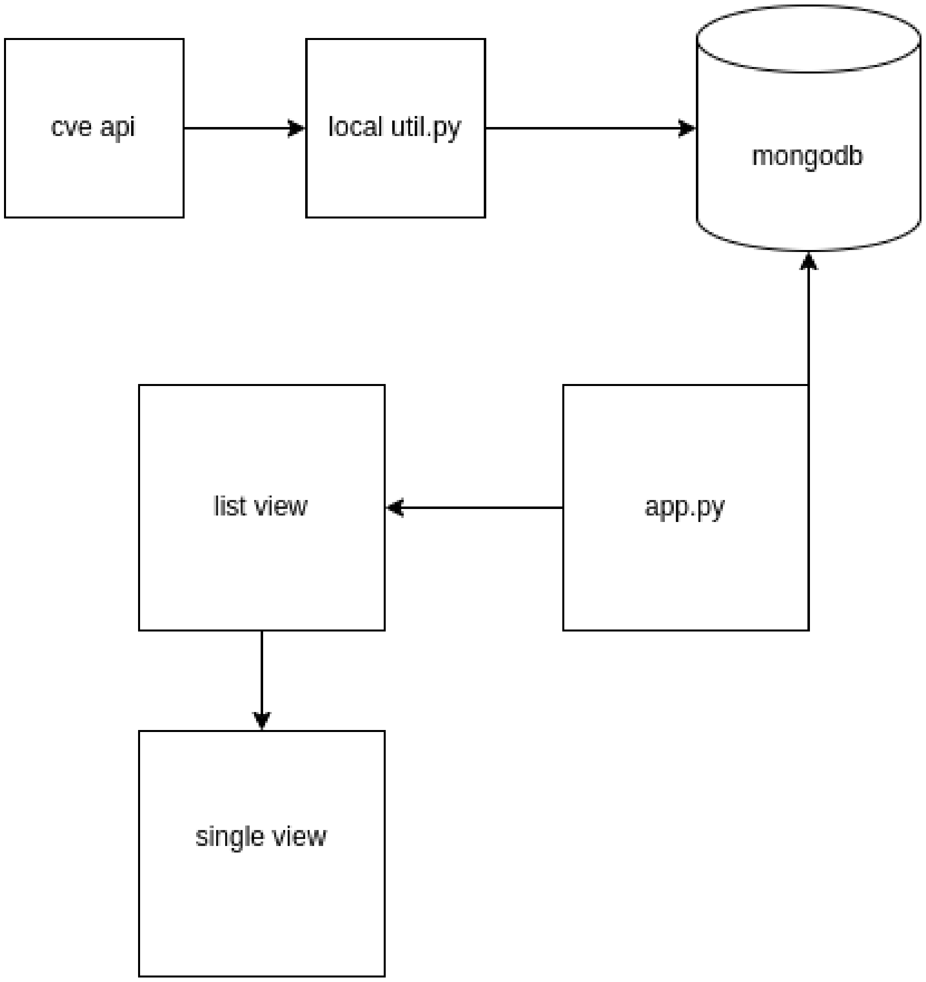
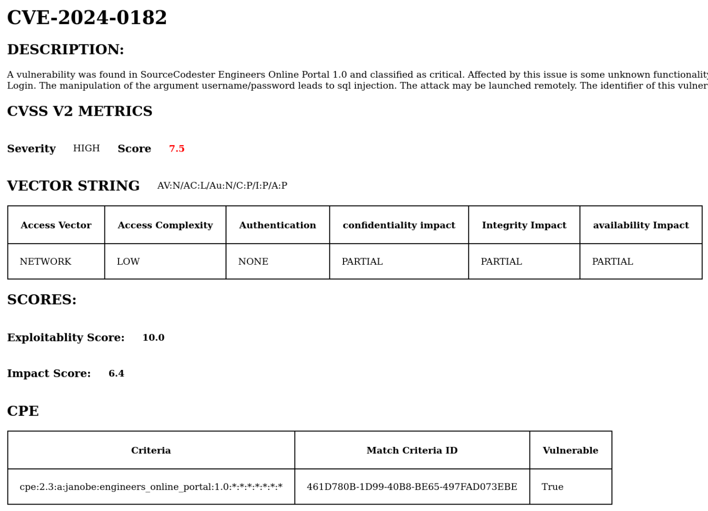


Figure 7: workflow architecture diagram



CVE-2024-0182

DESCRIPTION:

A vulnerability was found in SourceCodester Engineers Online Portal 1.0 and classified as critical. Affected by this issue is some unknown functionality of the file /admin/ of the component Admin Login. The manipulation of the argument username/password leads to sql injection. The attack may be launched remotely. The identifier of this vulnerability is VDB-249440.

CVSS V2 METRICS

Severity HIGH **Score** **7.5**

VECTOR STRING AV:N/AC:L/Au:N/C:P/I:P/A:P

Access Vector	Access Complexity	Authentication	confidentiality impact	Integrity Impact	availability Impact
NETWORK	LOW	NONE	PARTIAL	PARTIAL	PARTIAL

SCORES:

Exploitablity Score: **10.0**

Impact Score: **6.4**

CPE

Criteria	Match Criteria ID	Vulnerable
cpe:2.3:a:janobe:engineers_online_portal:1.0:*:*:*:*:*:*:*	461D780B-1D99-40B8-BE65-497FAD073EBE	True

Figure 8: SPECIFIC CVE