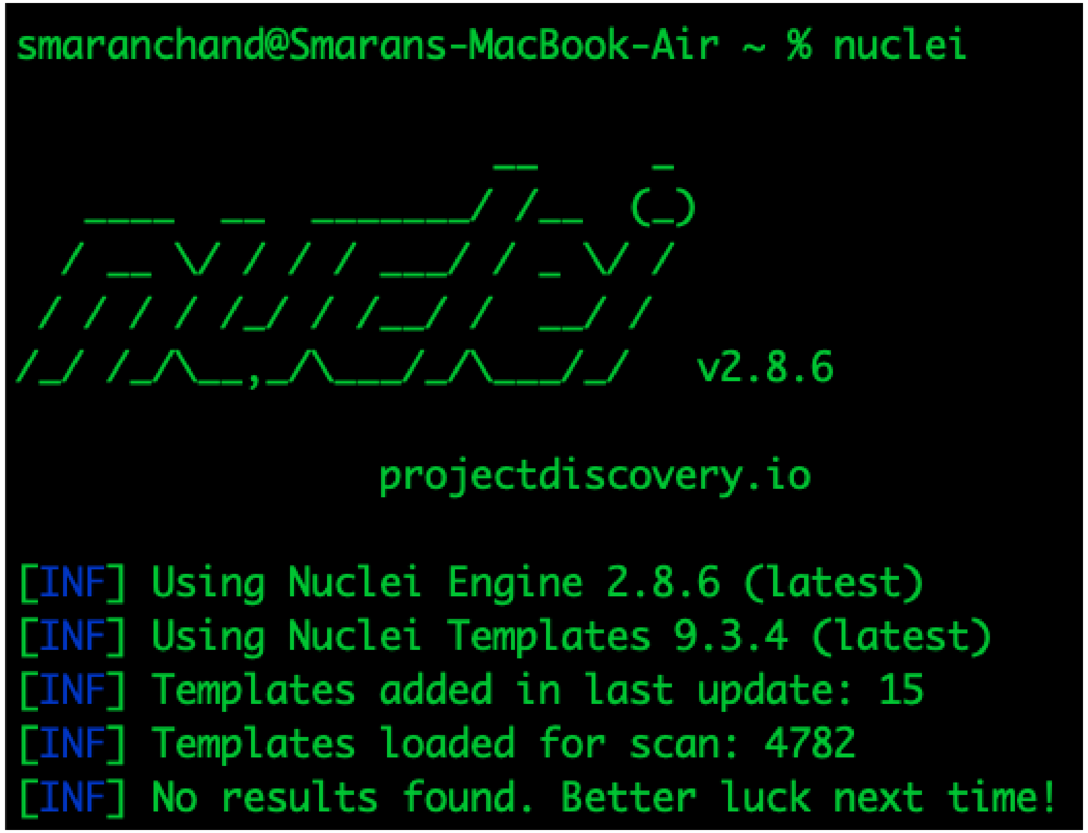
Introducing Nucci

Smaran Chand

- Application Security Enthusiast
- Sr. Penetration Tester at Eminence Ways
- One of the OWASP Kathmandu Leader
- Synack Red Teamer, Occasional Bug Bounty hunter
- Microsoft Azure/AWS certified Cloud Security Engineer
- Currently Researching Google Cloud Platform and MMM
- Noob Programmer
- Problem Solver, Unicorn
- https://smaranchand.com.np
- https://twitter.com/@smaranchand

How many of you use Nuclei scanner tool ?



```
smaranchand@Smarans-MacBook-Air ~ % nuclei

                     __     _
   ____  __  _______/ /__  (_)
  / __ \/ / / / ___/ / _ \/ /
 / / / / /_/ / /__/ /  __/ /
/_/ /_/\__,_/\___/_/\___/_/   v2.8.6

                projectdiscovery.io

[INF] Using Nuclei Engine 2.8.6 (latest)
[INF] Using Nuclei Templates 9.3.4 (latest)
[INF] Templates added in last update: 15
[INF] Templates loaded for scan: 4782
[INF] No results found. Better luck next time!
```

New bird in town?

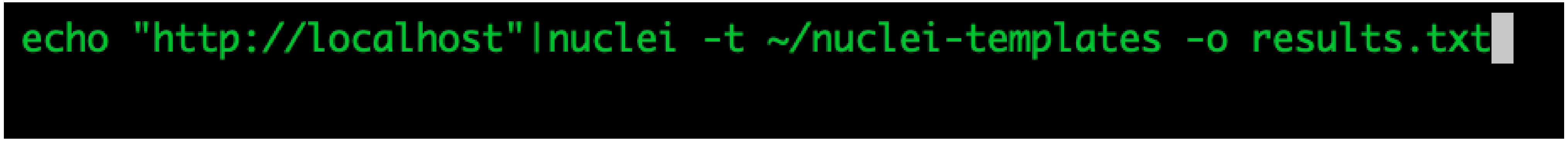
Nucci

Save your Nuclei scan results to cloud.

Nucci

- Just another tool that saves your Nuclei scan results in a cloud database.
- Takes input using stdin, sanitize and save in a Mongo DB.
- Python Flask app to view and create a basic PDF report.
- Does nothing much ¯_(ツ)_/¯ but helps sometimes.

Nucci



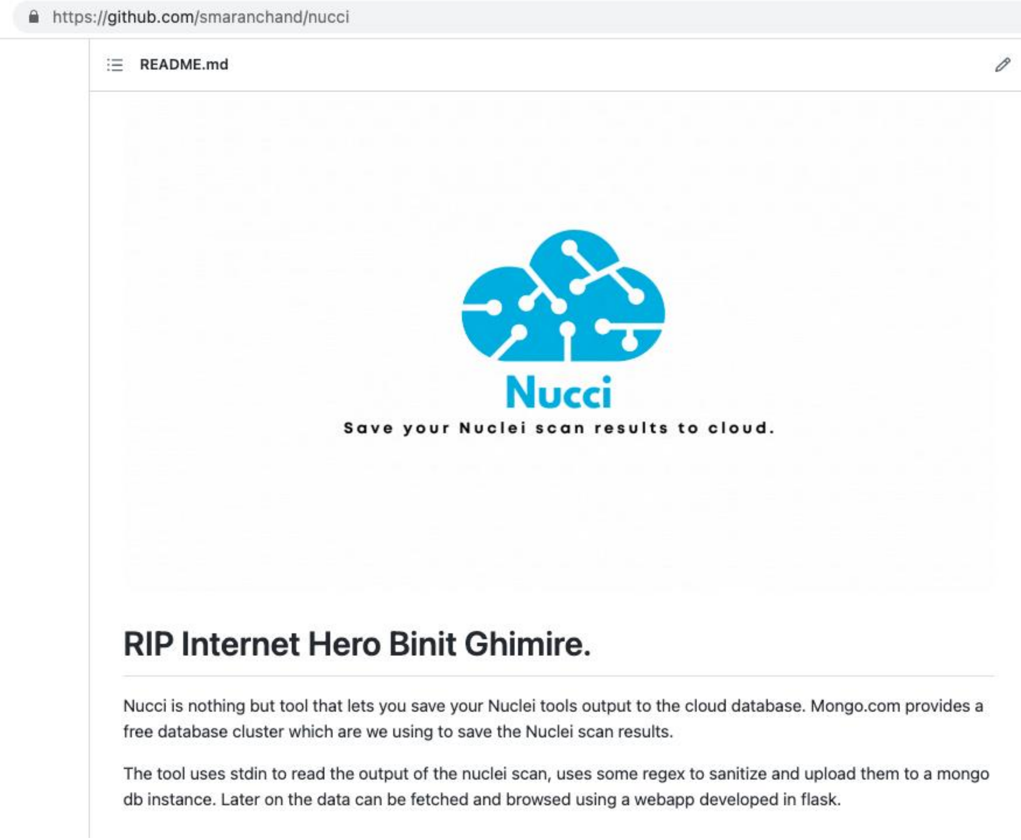
```
echo "http://localhost"|nuclei -t ~/nuclei-templates -o results.txt
```

Why Nucci when we can use –o results.txt ?
Or default report generator.

Because

- Retrieve the scan results when required remotely.
- Bugbounty collaboration/Group Pentesting.
- Run scan in cloud, Get results in local dashboard.
- Better view, search filters.
- Save few miliseconds by using |nucci instead of –o results.txt 😛

Nucci

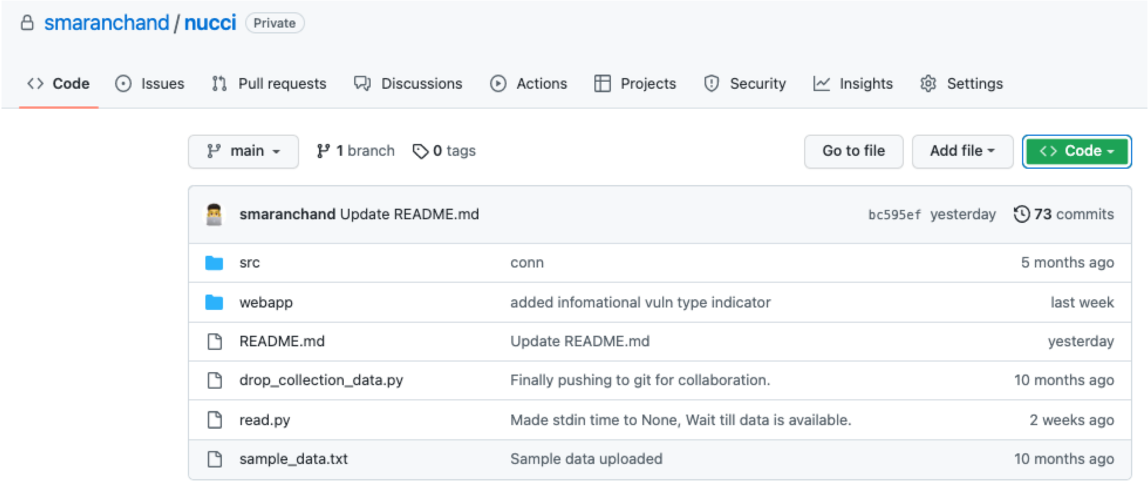


Save your Nuclei scan results to cloud.

RIP Internet Hero Binit Ghimire.

Nucci is nothing but tool that lets you save your Nuclei tools output to the cloud database. Mongo.com provides a free database cluster which are we using to save the Nuclei scan results.

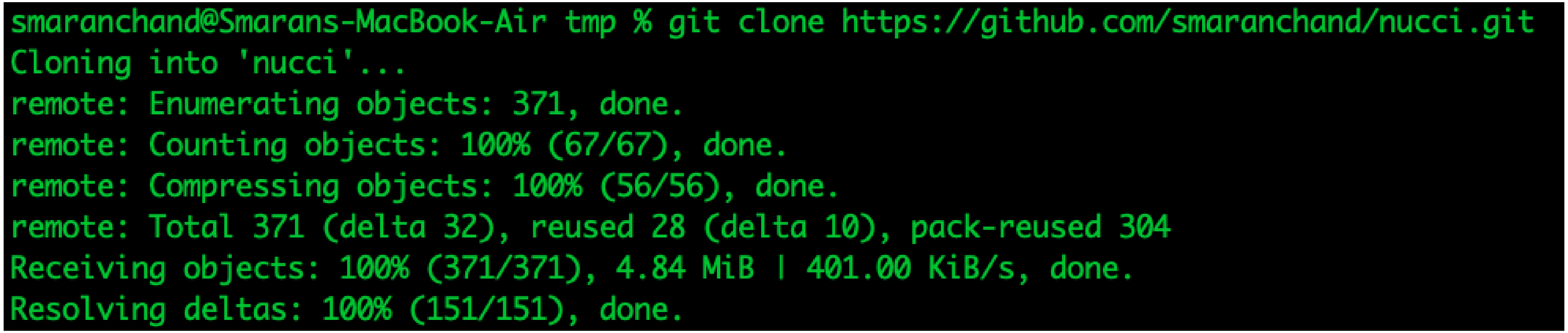
The tool uses stdin to read the output of the nuclei scan, uses some regex to sanitize and upload them to a mongo db instance. Later on the data can be fetched and browsed using a webapp developed in flask.



smaranchand Update README.md bc595ef yesterday 73 commits

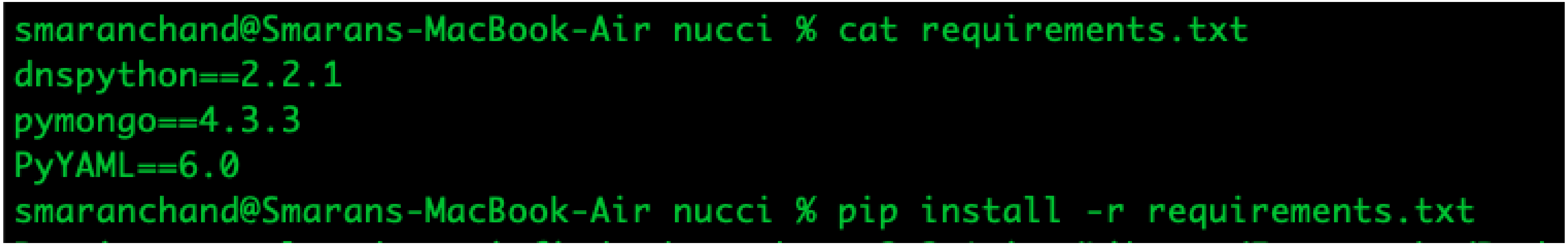
src	conn	5 months ago
webapp	added infomational vuln type indicator	last week
README.md	Update README.md	yesterday
drop_collection_data.py	Finally pushing to git for collaboration.	10 months ago
read.py	Made stdin time to None, Wait till data is available.	2 weeks ago
sample_data.txt	Sample data uploaded	10 months ago

git clone https://github.com/smaranchand/nucci.git



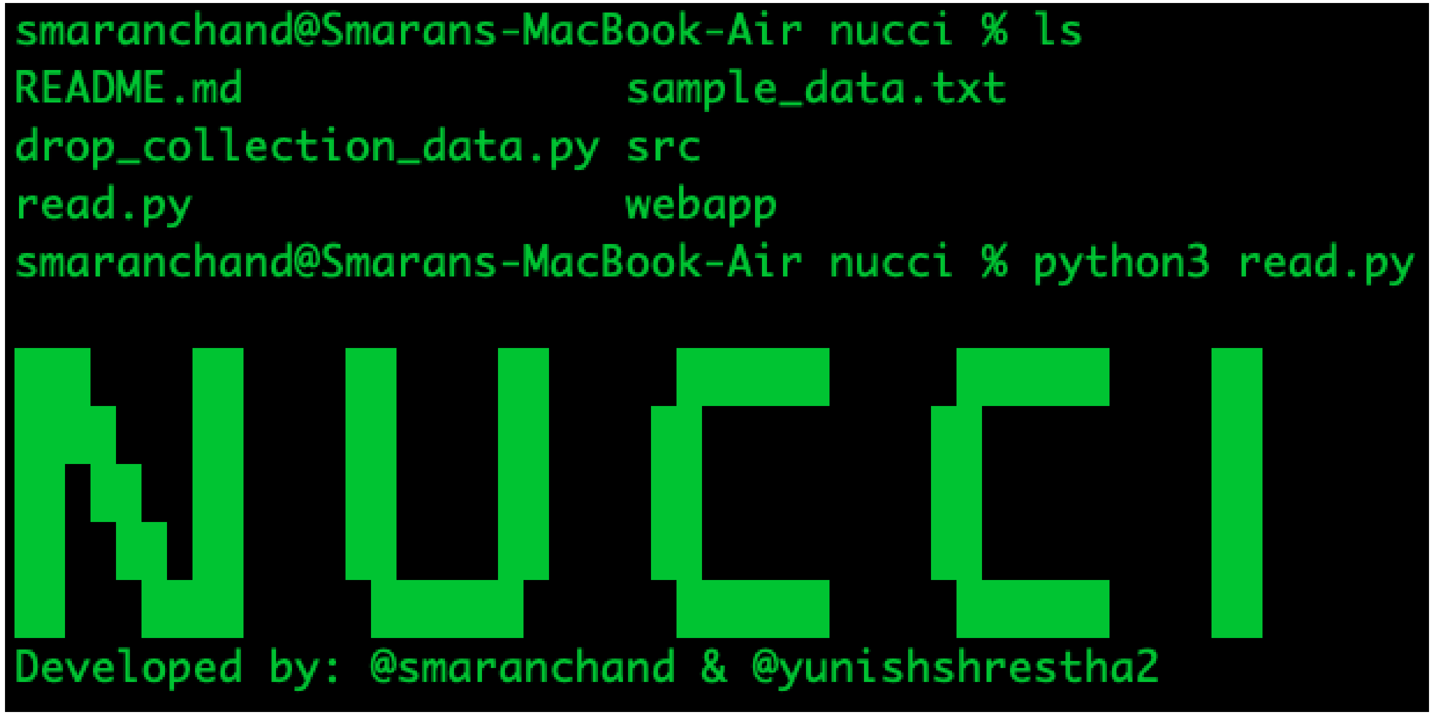
```
smaranchand@Smarans-MacBook-Air tmp % git clone https://github.com/smaranchand/nucci.git
Cloning into 'nucci'...
remote: Enumerating objects: 371, done.
remote: Counting objects: 100% (67/67), done.
remote: Compressing objects: 100% (56/56), done.
remote: Total 371 (delta 32), reused 28 (delta 10), pack-reused 304
Receiving objects: 100% (371/371), 4.84 MiB | 401.00 KiB/s, done.
Resolving deltas: 100% (151/151), done.
```

pip install –r requirements.txt



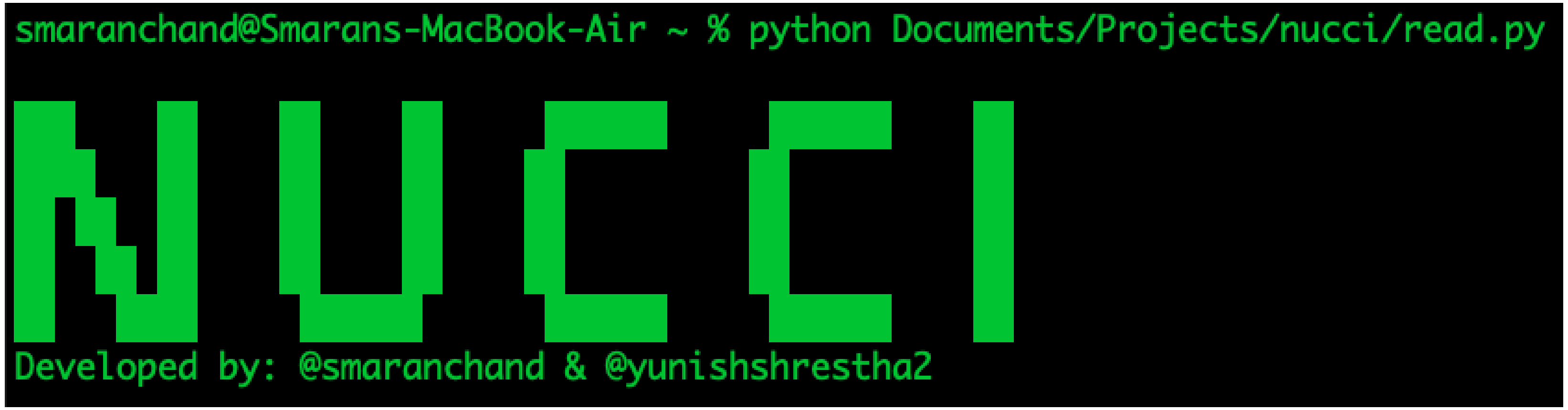
```
smaranchand@Smarans-MacBook-Air nucci % cat requirements.txt
dnspython==2.2.1
pymongo==4.3.3
PyYAML==6.0
smaranchand@Smarans-MacBook-Air nucci % pip install -r requirements.txt
```

Python3
read.py



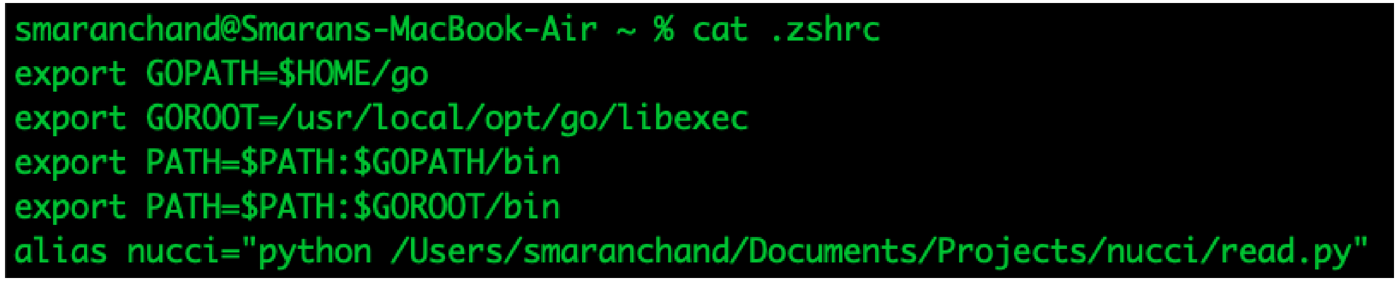
```
smaranchand@Smarans-MacBook-Air nucci % ls
README.md               sample_data.txt
drop_collection_data.py src
read.py                 webapp
smaranchand@Smarans-MacBook-Air nucci % python3 read.py

NUCCI

Developed by: @smaranchand & @yunishshrestha2
```

Set an alias



```
smaranchand@Smarans-MacBook-Air ~ % python Documents/Projects/nucci/read.py
```

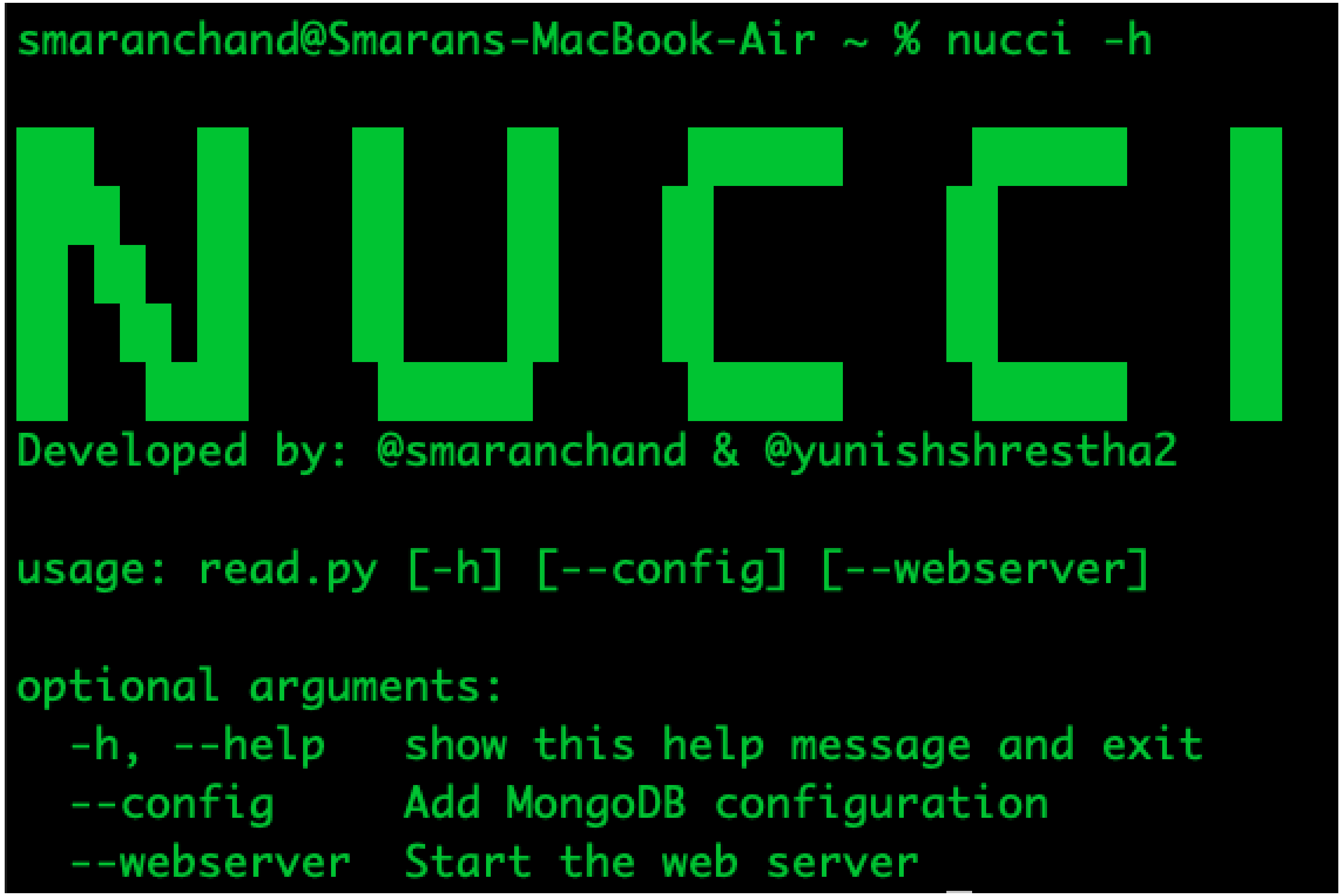
```
Developed by: @smaranchand & @yunishshrestha2
```



```
smaranchand@Smarans-MacBook-Air ~ % cat .zshrc
export GOPATH=$HOME/go
export GOROOT=/usr/local/opt/go/libexec
export PATH=$PATH:$GOPATH/bin
export PATH=$PATH:$GOROOT/bin
alias nucci="python /Users/smaranchand/Documents/Projects/nucci/read.py"
```

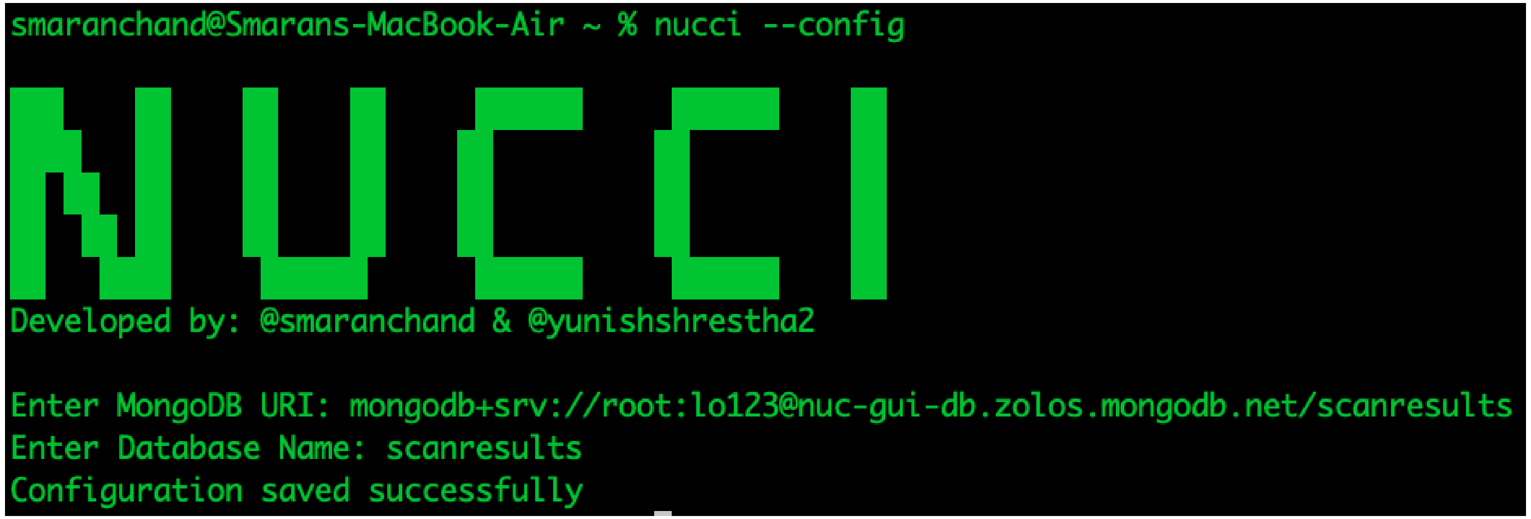
Better Right ?

nucci -h



```
smaranchand@Smarans-MacBook-Air ~ % nucci -h

Developed by: @smaranchand & @yunishshrestha2

usage: read.py [-h] [--config] [--webserver]

optional arguments:
  -h, --help    show this help message and exit
  --config      Add MongoDB configuration
  --webserver   Start the web server
```

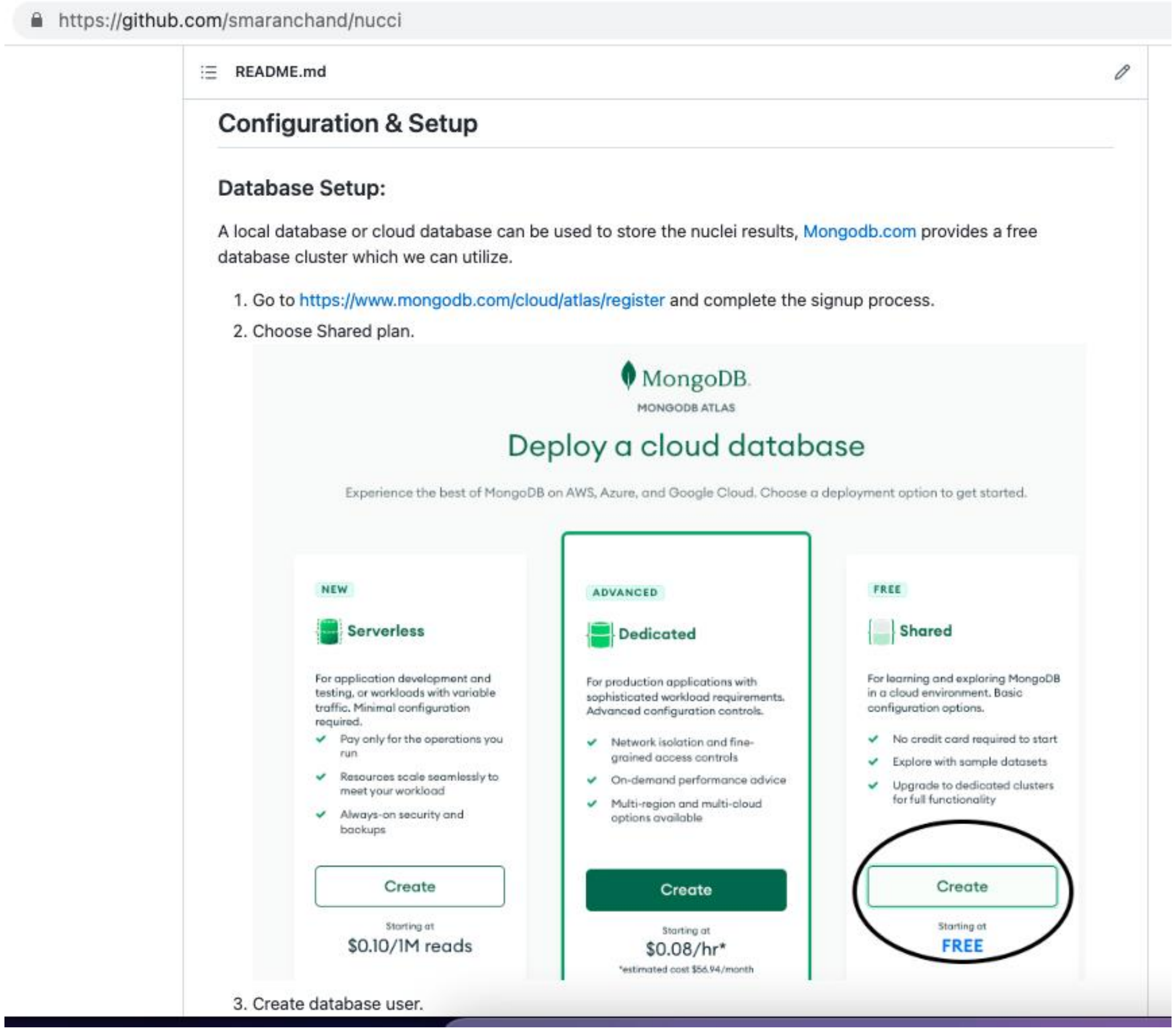
nucci --config



```
smaranchand@Smarans-MacBook-Air ~ % nucci --config

Developed by: @smaranchand & @yunishshrestha2

Enter MongoDB URI: mongodb+srv://root:lo123@nuc-gui-db.zolos.mongodb.net/scanresults
Enter Database Name: scanresults
Configuration saved successfully
```

Saves database connection string at your ~/.nucci/config.yaml

Get a free Mongo DB cluster from mongodb.com



Configuration & Setup

Database Setup:

A local database or cloud database can be used to store the nuclei results, Mongodb.com provides a free database cluster which we can utilize.

1. Go to https://www.mongodb.com/cloud/atlas/register and complete the signup process.
2. Choose Shared plan.

Deploy a cloud database

Experience the best of MongoDB on AWS, Azure, and Google Cloud. Choose a deployment option to get started.

NEW

Serverless

For application development and testing, or workloads with variable traffic. Minimal configuration required.

- ✓ Pay only for the operations you run
- ✓ Resources scale seamlessly to meet your workload
- ✓ Always-on security and backups

Starting at
$0.10/1M reads

ADVANCED

Dedicated

For production applications with sophisticated workload requirements. Advanced configuration controls.

- ✓ Network isolation and fine-grained access controls
- ✓ On-demand performance advice
- ✓ Multi-region and multi-cloud options available

Starting at
$0.08/hr*
*estimated cost $56.94/month

FREE

Shared

For learning and exploring MongoDB in a cloud environment. Basic configuration options.

- ✓ No credit card required to start
- ✓ Explore with sample datasets
- ✓ Upgrade to dedicated clusters for full functionality

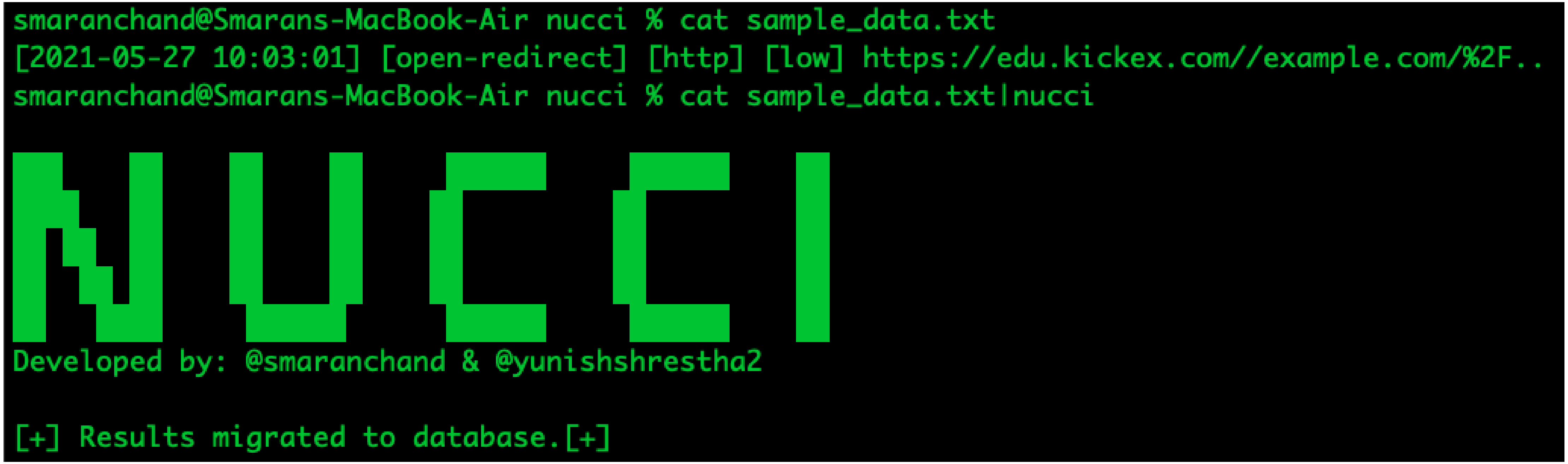
Starting at
FREE

3. Create database user.

Nucci in action

```
subfinder -d smaranchand.com.np|httpx|nuclei -t ~/nuclei-templates|nucci
```

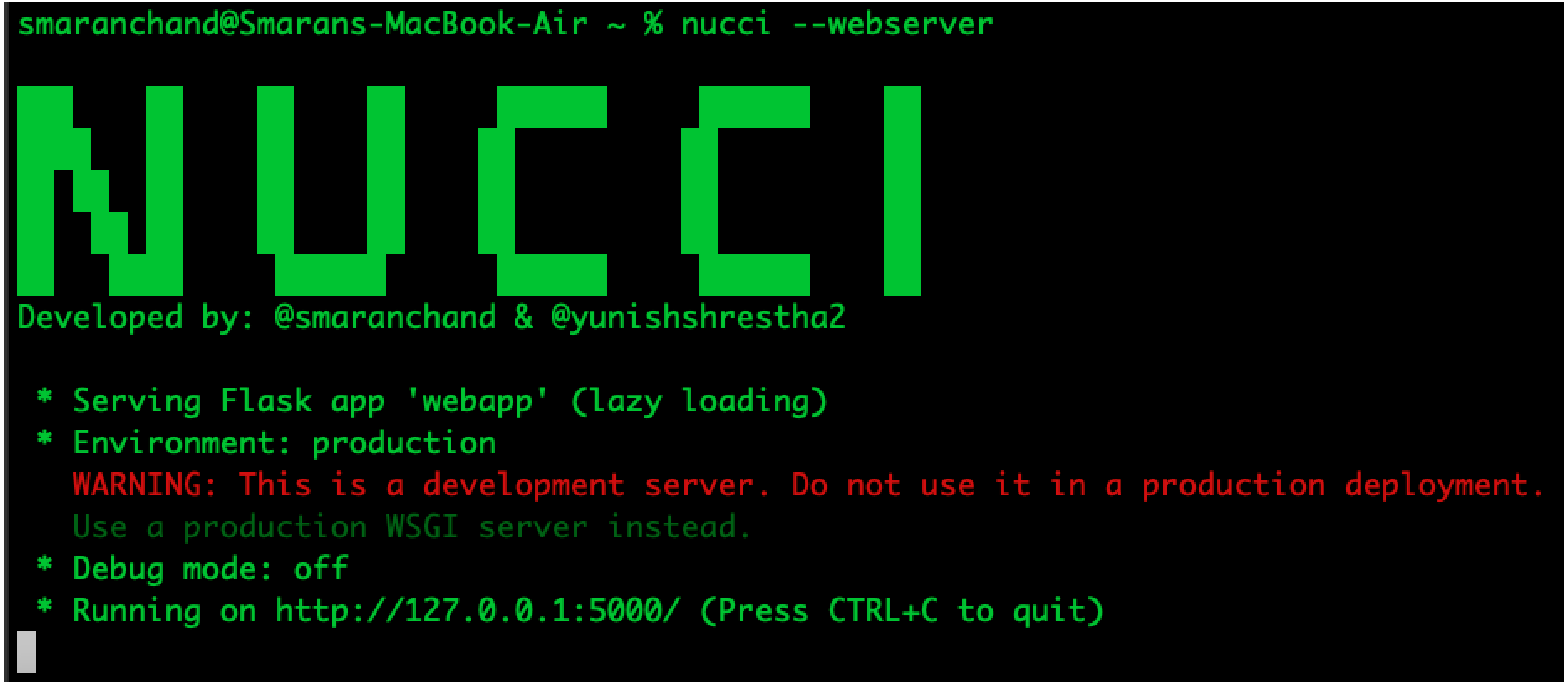
Also simply



```
smaranchand@Smarans-MacBook-Air nucci % cat sample_data.txt
[2021-05-27 10:03:01] [open-redirect] [http] [low] https://edu.kickex.com//example.com/%2F..
smaranchand@Smarans-MacBook-Air nucci % cat sample_data.txt|nucci
```

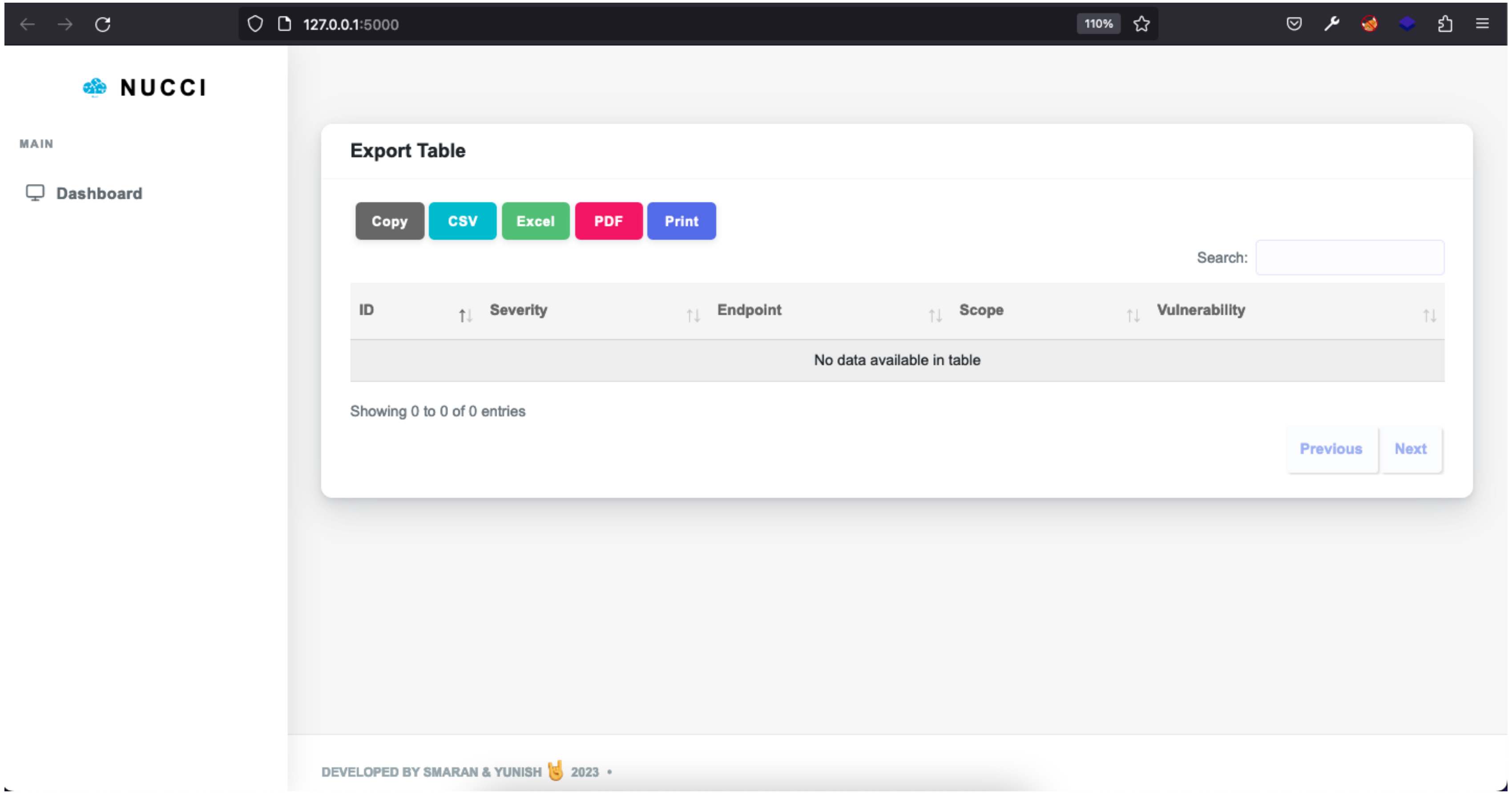
```
Developed by: @smaranchand & @yunishshrestha2

[+] Results migrated to database.[+]
```

View Nucci dashboard



```
smaranchand@Smarans-MacBook-Air ~ % nucci --webserver

NUCCI
Developed by: @smaranchand & @yunishshrestha2

 * Serving Flask app 'webapp' (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: off
 * Running on http://127.0.0.1:5000/ (Press CTRL+C to quit)
```

Nucci Dashboard



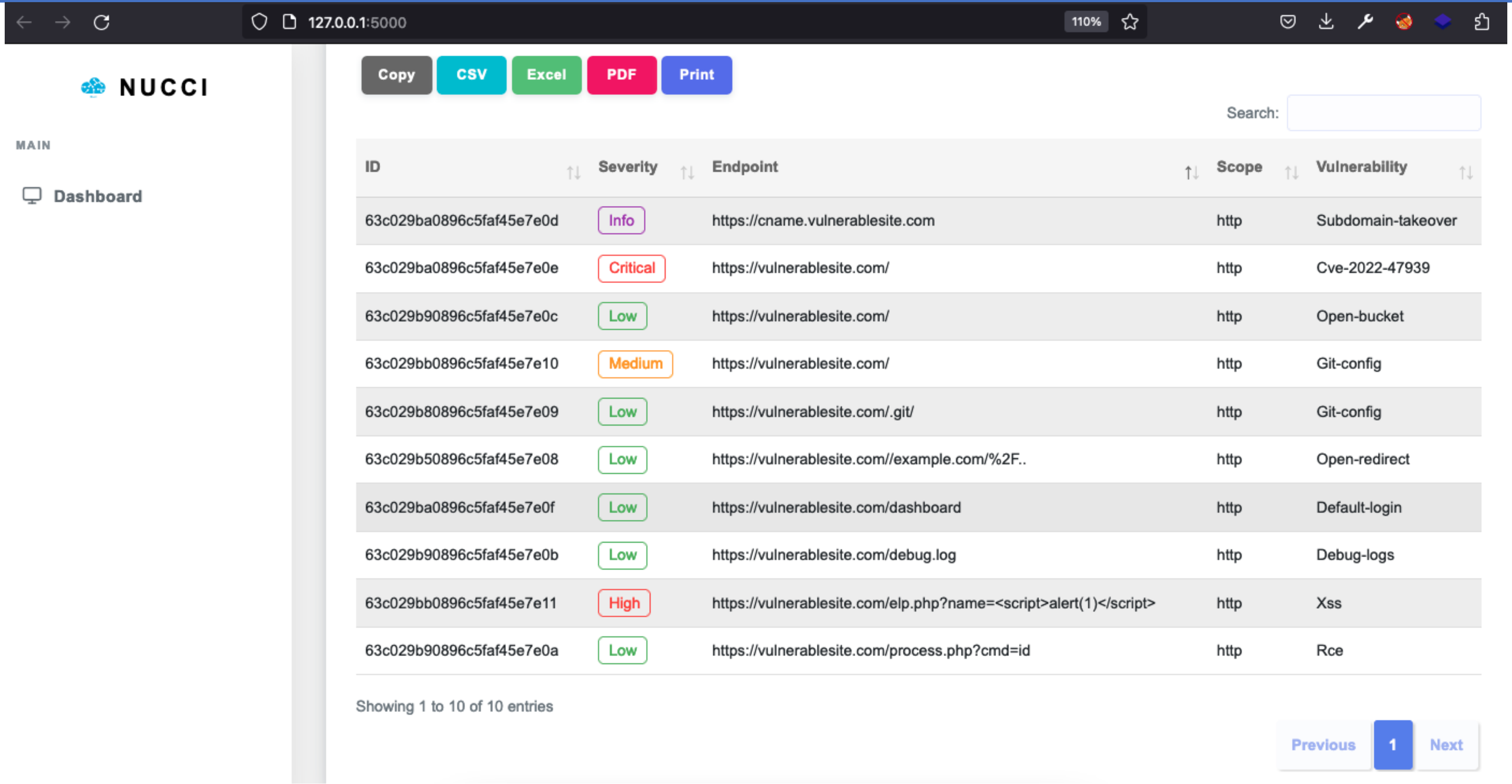
Export Table

ID	Severity	Endpoint	Scope	Vulnerability
No data available in table				

Showing 0 to 0 of 0 entries

DEVELOPED BY SMARAN & YUNISH 🤘 2023 •

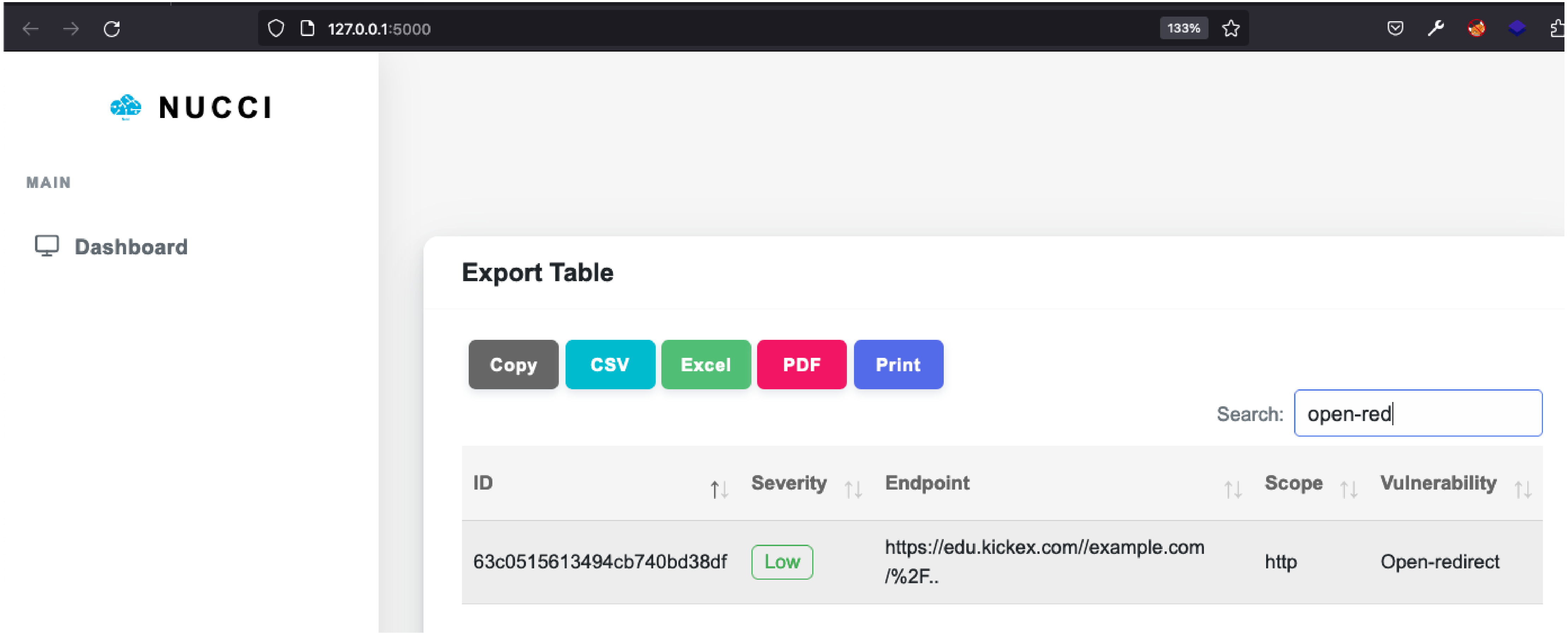
Nucci Dashboard with Nuclei Scan results



ID	Severity	Endpoint	Scope	Vulnerability
63c029ba0896c5faf45e7e0d	Info	https://cname.vulnerablesite.com	http	Subdomain-takeover
63c029ba0896c5faf45e7e0e	Critical	https://vulnerablesite.com/	http	Cve-2022-47939
63c029b90896c5faf45e7e0c	Low	https://vulnerablesite.com/	http	Open-bucket
63c029bb0896c5faf45e7e10	Medium	https://vulnerablesite.com/	http	Git-config
63c029b80896c5faf45e7e09	Low	https://vulnerablesite.com/.git/	http	Git-config
63c029b50896c5faf45e7e08	Low	https://vulnerablesite.com//example.com/%2F..	http	Open-redirect
63c029ba0896c5faf45e7e0f	Low	https://vulnerablesite.com/dashboard	http	Default-login
63c029b90896c5faf45e7e0b	Low	https://vulnerablesite.com/debug.log	http	Debug-logs
63c029bb0896c5faf45e7e11	High	https://vulnerablesite.com/elp.php?name=<script>alert(1)</script>	http	Xss
63c029b90896c5faf45e7e0a	Low	https://vulnerablesite.com/process.php?cmd=id	http	Rce

Showing 1 to 10 of 10 entries

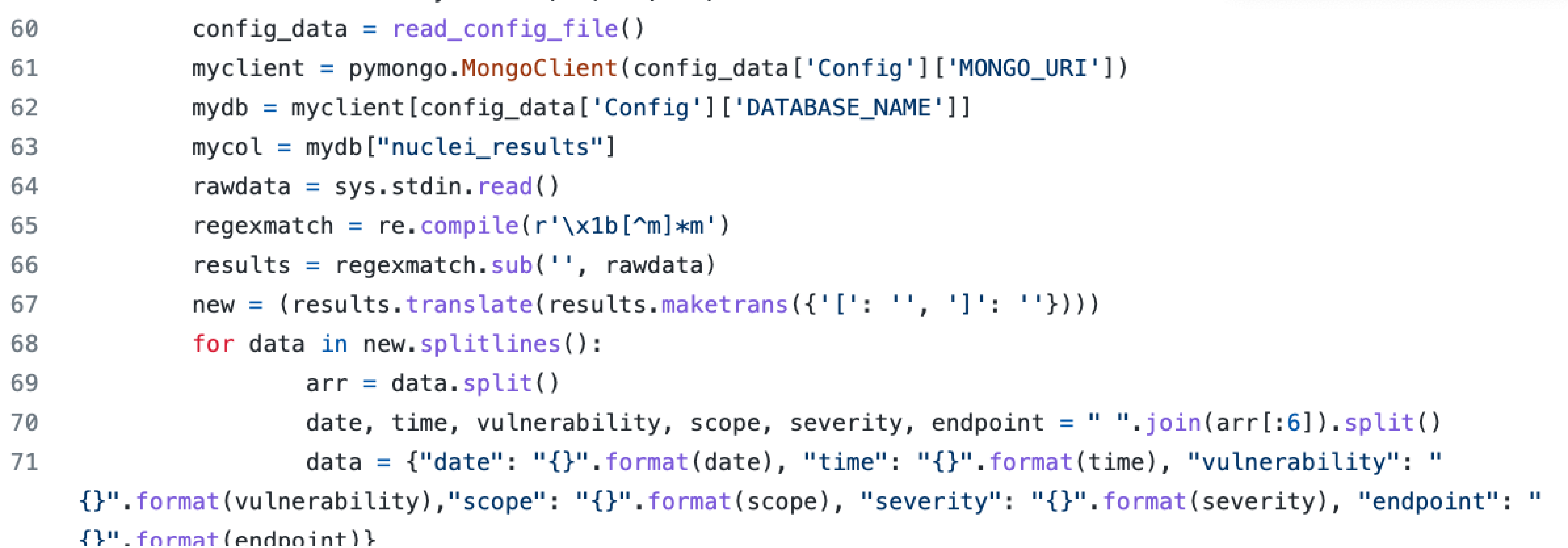
Search Filter



Export Table

Search: open-red

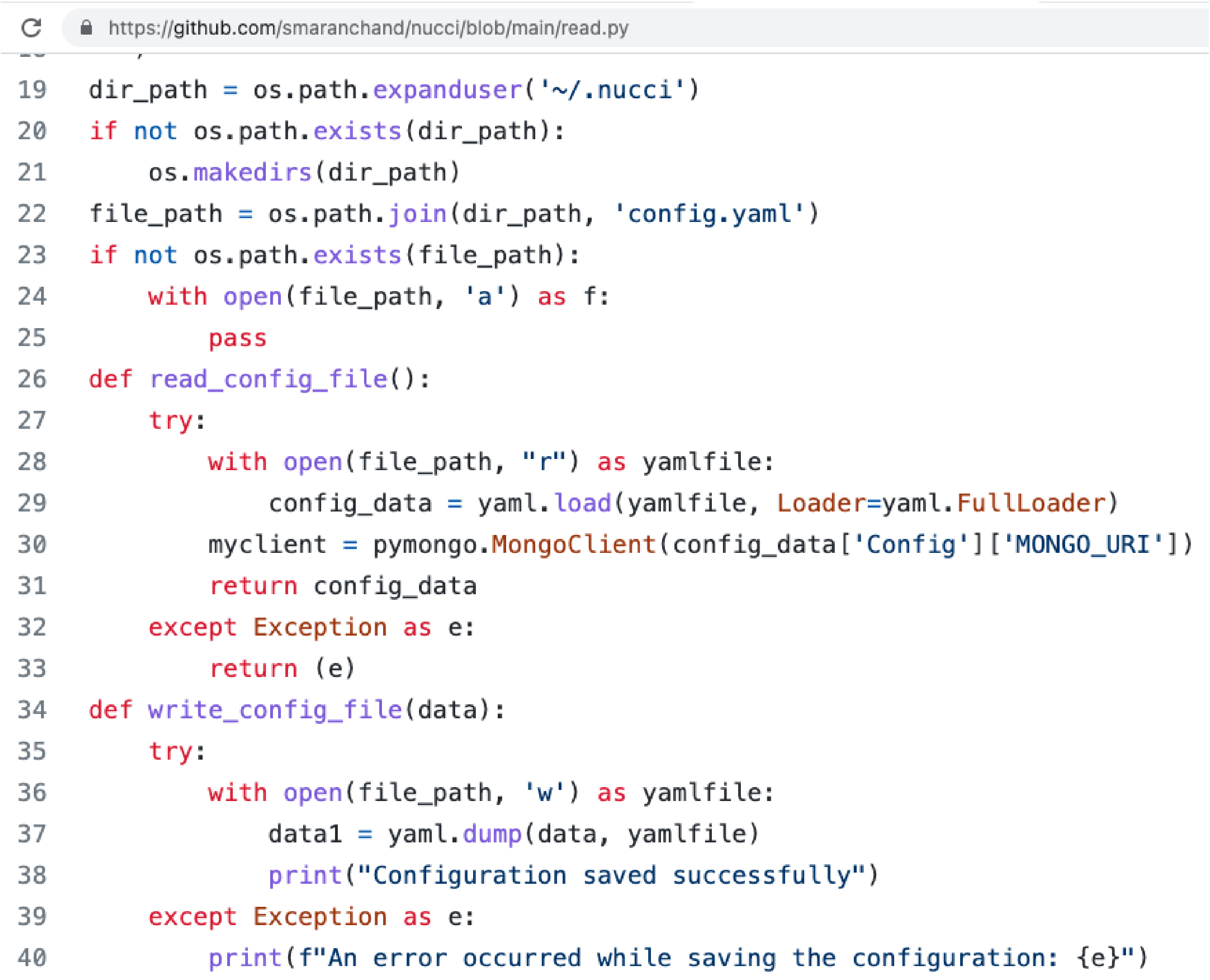
ID	Severity	Endpoint	Scope	Vulnerability
63c0515613494cb740bd38df	Low	https://edu.kickex.com//example.com /%2F..	http	Open-redirect



```
        config_data = read_config_file()
        myclient = pymongo.MongoClient(config_data['Config']['MONGO_URI'])
        mydb = myclient[config_data['Config']['DATABASE_NAME']]
        mycol = mydb["nuclei_results"]
        rawdata = sys.stdin.read()
        regexmatch = re.compile(r'\x1b[^m]*m')
        results = regexmatch.sub('', rawdata)
        new = (results.translate(results.maketrans({'[': '', ']': ''})))
        for data in new.splitlines():
                arr = data.split()
                date, time, vulnerability, scope, severity, endpoint = " ".join(arr[:6]).split()
                data = {"date": "{}".format(date), "time": "{}".format(time), "vulnerability": "{}".format(vulnerability),"scope": "{}".format(scope), "severity": "{}".format(severity), "endpoint": "{}".format(endpoint)}
```

Piece of Code

Piece of Code



```python
dir_path = os.path.expanduser('~/.nucci')
if not os.path.exists(dir_path):
    os.makedirs(dir_path)
file_path = os.path.join(dir_path, 'config.yaml')
if not os.path.exists(file_path):
    with open(file_path, 'a') as f:
        pass
def read_config_file():
    try:
        with open(file_path, "r") as yamlfile:
            config_data = yaml.load(yamlfile, Loader=yaml.FullLoader)
        myclient = pymongo.MongoClient(config_data['Config']['MONGO_URI'])
        return config_data
    except Exception as e:
        return (e)
def write_config_file(data):
    try:
        with open(file_path, 'w') as yamlfile:
            data1 = yaml.dump(data, yamlfile)
            print("Configuration saved successfully")
    except Exception as e:
        print(f"An error occurred while saving the configuration: {e}")
```

Next ?

- Fully Automate Vulnerability Report Generation.
- Create a Pypi package for easy install.
- CVSSv3.1 integration, references from Hackerone hacktivity, disclosed bugbounty reports etc.
- Work on UI/UX .
- Add authentication in dashboard and some security features.
- Expecting Pull Requests from public.

Questions?

Thanks and Shoutouts

- Yunish Shrestha
- Kailash Bohara (@corrupted_brain)
- Ankit Pandey (4_N_K_1_T)
- Rohitash Kumar (Rooks)