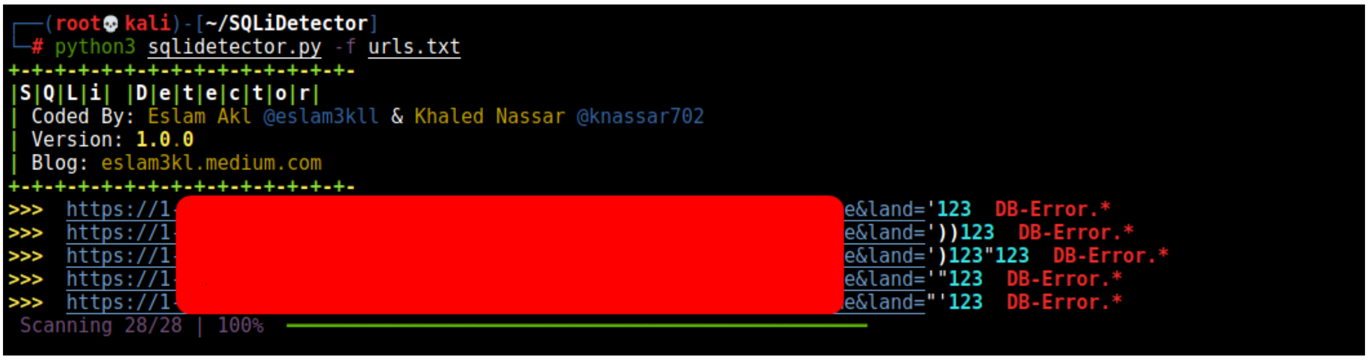
SQLiDetector – Helps You To Detect SQL Injection «Error Based» By Sending Multiple Requests With 14 Payloads And Checking For 152 Regex Patterns For Different Databases

[Download SQLiDetector]

Simple python script supported with BurpBouty profile that helps you to detect SQL injection «Error based» by sending multiple requests with 14 payloads and checking for 152 regex patterns for different databases.

```
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
| S|Q|L|i| |D|e|t|e|c|t|o|r|
| Coded By: Eslam Akl @eslam3kll & Khaled Nassar @knassar702
| Version: 1.0.0
| Blog: eslam3kl.medium.com
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-
```



Description

The main idea for the tool is scanning for Error Based SQL Injection by using different payloads like

```
'123
''123
123
")123
"))123
)123
`))123
'))123
')123"123
[]123
""123
'"123
"'123
\123
```

And match for 152 error regex patterns for different databases.
Source: https://github.com/sqlmapproject/sqlmap/blob/master/data/xml/errors.xml

How does it work?

It’s very simple, just organize your steps as follows

1. Use your subdomain grabber script or tools.
2. Pass all collected subdomains to httpx or httprobe to get only live subs.
3. Use your links and URLs tools to grab all waybackurls like waybackurls, gau, gauplus, etc.
4. Use URO tool to filter them and reduce the noise.
5. Grep to get all the links that contain parameters only. You can use Grep or GF tool.
6. Pass the final URLs file to the tool, and it will test them.

The final schema of URLs that you will pass to the tool must be like this one

```
https://aykalam.com?x=test&y=fortest
http://test.com?parameter=ayhaga
```

Installation and Usage

Just run the following command to install the required libraries.

```
~/eslam3kl/SQLiDetector# pip3 install -r requirements.txt
```

To run the tool itself.

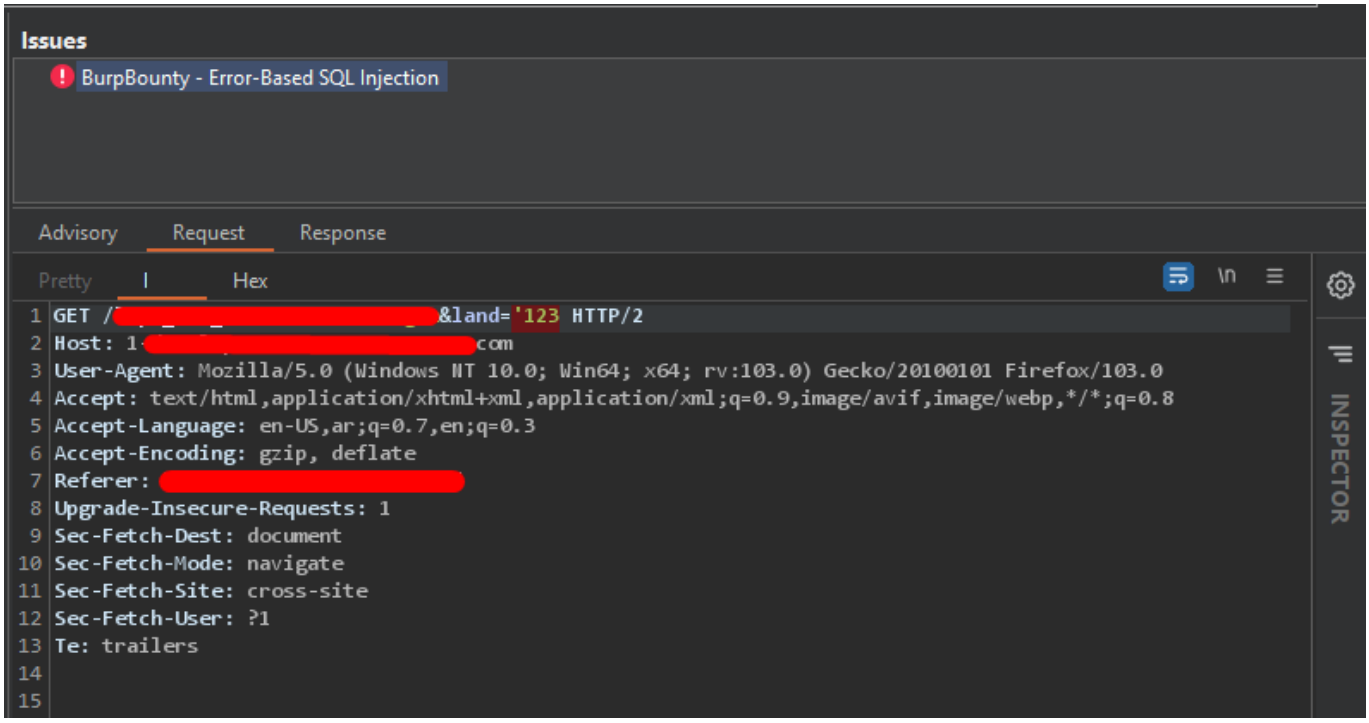
```
# cat urls.txt
http://testphp.vulnweb.com/artists.php?artist=1

# python3 sqlidetector.py -h
usage: sqlidetector.py [-h] -f FILE [-w WORKERS] [-p PROXY] [-
t TIMEOUT] [-o OUTPUT]
A simple tool to detect SQL errors
optional arguments:
  -h, --help            show this help message and exit]
  -f FILE, --file FILE  [File of the urls]
  -w WORKERS, --workers [WORKERS Number of threads]
  -p PROXY, --proxy [PROXY Proxy host]
  -t TIMEOUT, --timeout [TIMEOUT Connection timeout]
  -o OUTPUT, --output [OUTPUT [Output file]

# python3 sqlidetector.py -f urls.txt -w 50 -o output.txt -t
10
```

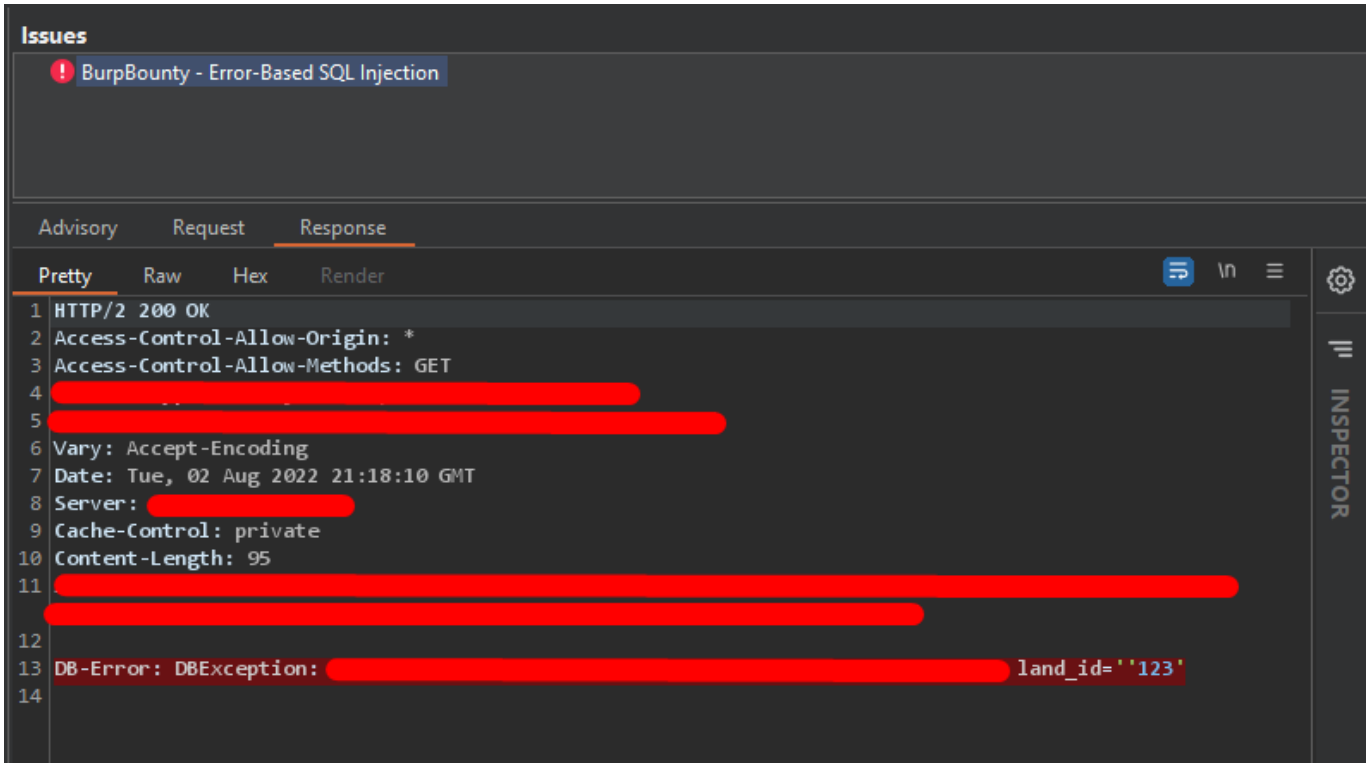
BurpBounty Module

I’ve created a burpbounty profile that uses the same payloads add injecting them at multiple positions like

- Parameter name
- Parameter value
- Headers
- Paths

I think it’s more effective and will helpful for POST request that you can’t test them using the Python script.





How does it test the parameter?

What’s the difference between this tool and any other one? If we have a link like this one https://example.com?file=aykalam&username=eslam3kl so we have 2 parameters. It creates 2 possible vulnerable URLs.

1. It will work for every payload like the following

https://example.com?file=123'&username=eslam3kl
https://example.com?file=aykalam&username=123'

2. It will send a request for every link and check if one of the patterns is existing using regex.
3. For any vulnerable link, it will save it at a separate file for every process.

Upcoming updates

- Output json option.
- Adding proxy option.
- Adding threads to increase the speed.
- Adding progress bar.
- Adding more payloads.
- Adding BurpBounty Profile.
- Inject the payloads in the parameter name itself.

If you want to contribute, feel free to do that. You’re welcome :)

Thanks to

Thanks to Mohamed El-Khayat and Orwa for the amazing paylaods and ideas. Follow them and you will learn more

https://twitter.com/Mohamed87Khayat
https://twitter.com/GodfatherOrwa