**API Discovery tools:**

1. Kiterunner
2. Ffuf
3. Arjun
4. Burpsuite intruder
5. OWASP ZAP API explorer

**API Scanning tools:**

1. Vooki

**Kiterunner:**

Kiterunner is a Bruteforcing/ Fuzzing tool for APIs, uses info on how the API was built

We can give our Website as the input.

We can give list of endpoints in the notepad as input.

But when trying it on a peak website, it was giving all the endpoints with 411 response or 404. It is just using the wordlist to append the words in the wordlist on the api endpoints.

Also download the wordlists and extract the file from the following link:

https://ryn0f1sh.medium.com/kiterunner-api-fuzzer-windows-installation-b6e524b93ad9

**Running Kiterunner:**

./kiterunner scan

./kiterunner scan -d=2

./kiterunner scan -d=3 --kitebuilder-full-scan

./kiterunner scan -w /home/sai/Downloads/routes-large.kite

./kiterunner brute -A -d=3

/kiterunner brute -A=apiroutes-210228

/kiterunner scan -A=apiroutes-210228

./kiterunner scan list.txt -w /home/sai/Downloads/routes-large.kite -A=apiroutes-210228

./kiterunner scan -w /home/sai/Downloads/routes-large.kite -A=apiroutes-210328:20000 -x 20 -j 1 --fail-status-codes 400,401,404,403,501,502,426,411

./kiterunner scan -w /home/sai/Downloads/routes-large.kite -A=apiroutes-210228

Here is the output result from kiterunner:

and upon following the redirection, we are receiving the error: 404 Which mean the URL doesn’t exist.

Kiterunner claims that it has a right accurate data set for the OpenAPI/ swagger specifications.

Content discovery tooling currently relies on static txt files as wordlists and it is up to the user to perform bruteforces using different HTTP methods or to have wordlists with parameters and values pre-filled.

For example, If we have an endpoint with the path: /api/v1/notes\_todo with GET, PUT DELETE methods enabled, then in order to reach this specific endpoint, our wordlist should also contain this specific endpoint with the enabled HTTP verbs.

It is very likely that we miss these kinds of endpoints.

**Pros:**

Kiterunner can be used to discover the secret routes in the API. Kiterunner has the most popular and large wordlsists.

Replay requests is the key feature.

With –proxy, traffic can be re routed to any intercepting tools such as burpsuite.

**kr kb replay -w routes.kite "POST 500 [ 40, 5, 1] https://example " --proxy http://localhost:8080**

**Cons:**

Kiterunner uses predefined worlist to bruteforce the API endpoints. But, whereas the kiterunner cant be used for our Peak API as it does the fuzzing based on the wordlists.

We may not find all the URIs of the Peak APIs because of the wordlist approach.

**Ffuf (Fuzz faster U fool)**

Ffuf is also a fuzzing tool similar to the kiterunner.

Ffuf is also a tool fuzzing the API endpoints based on the wordlist provided.

Gives output as JSON, prettyprint, text

Wordlists can be downloaded from the following link:

<https://raw.githubusercontent.com/danielmiessler/SecLists/master/Discovery/Web-Content/api/api-seen-in-wild.txt>

**Running ffuf:**

ffuf -w wordlist -u .com/FUZZ

**Proxy through Burpsuite:**

ffuf -w wordlist -u force.com/FUZZ -x http://127.0.0.1:8080

**Pros:**

Ffuf tool calls can be proxied through burpsuite. Burpsuite can be used to analyze the API calls from the Ffuf.

It can be used for fuzzing the HTTP headers.

It can also be used for Host discovery by the following command:

ffuf -w subdomains.txt -u <http://website.com/> -H “Host: FUZZ.website.com”

**Cons:**

As Ffuf uses predefined wordlist to bruteforce the API endpoints. It cant be used for our Peak API as it does the fuzzing based on the wordlists.

We may not find all the URIs of the Peak APIs because of the wordlist approach.

**Arjun:**

Arjun can find query parameters for URL endpoints. It finds valid HTTP parameters with a huge default dictionary of 10,985 parameter names.

But when trying to trigger arjun with the below command, we couldn’t get any results from the tool.

**Owasp ZAP API Explorator**:

OWASP ZAP Api Exploration is only for standard documented API such as Swagger or Open API standards as per the OWASP API user guide.

OWASP ZAP exploring has addons for

SOAP WSDL

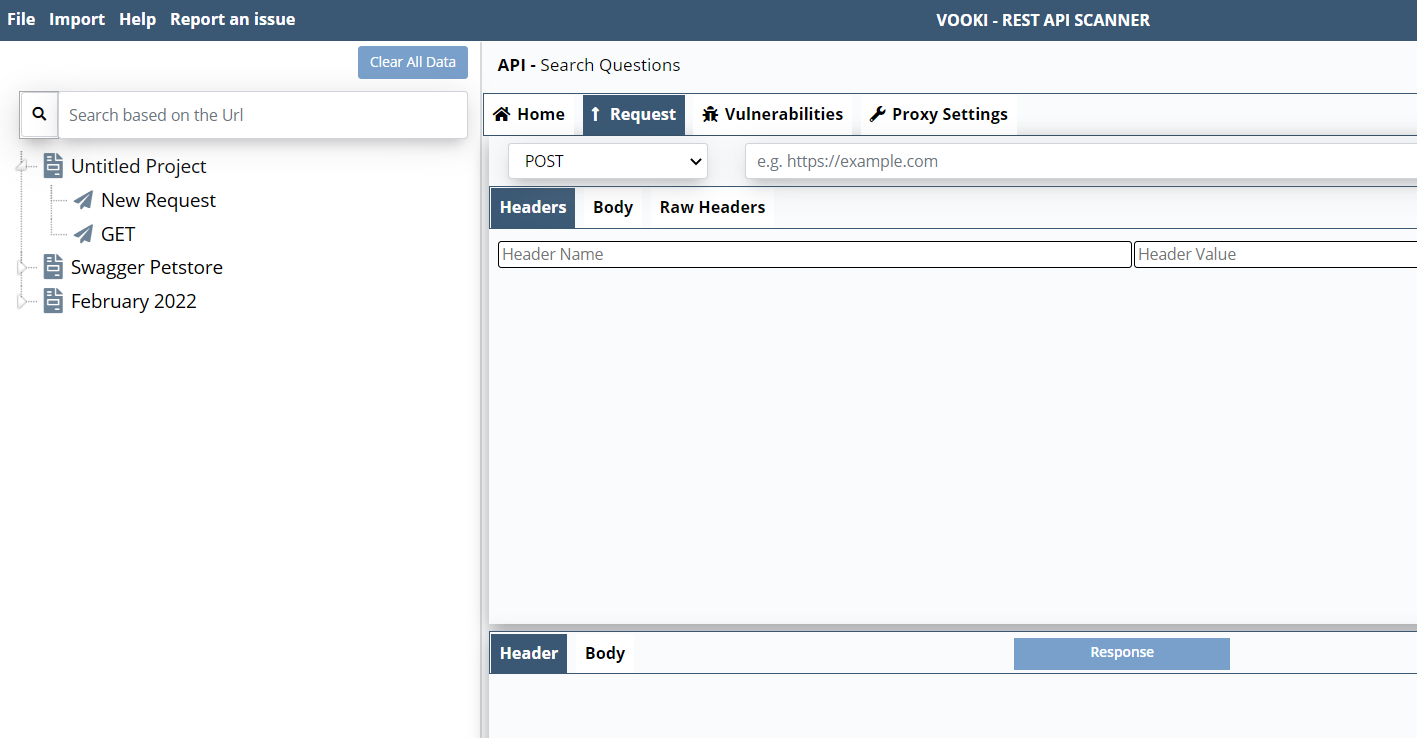
Open API

Swagger APIs

As our Peak APIs are not documented according to the above standard definitions.Therefore, it cannot be used for our Peak APIs discovery.

**Vooki:**

In vooki, we can add the requests manually by creating a separate project.



Vooki can be used to import the Postman collections and Environment variables.

Pros:

Vooki has the option of importing postman collections and environment which is the best import option if we don’t have any API standards.

Cons:

Vooki is giving only the Configuration related vulnerabilities such as security misconfigurations. Doesn’t seems to be working on injection points, Broken object level or rate limiting.

**Links:**

<https://blog.assetnote.io/2021/04/05/contextual-content-discovery/>