

Forum:[Hack Forums](#)

Thread: [How to build a web scraper for hidden endpoints?](#)

FlexagonV2 - August 6, 2018, 12:10 pm

Hey, I remember someone telling me to make a scraper to find hidden endpoints. I know I have to use Scrapy, but what else do I need? Also, how do I find those hidden endpoints?

c3ypt1c - August 7, 2018, 3:32 pm

What are you talking about?

What endpoints?

Also, I recommend Selenium for Python to scrape.

FlexagonV2 - August 16, 2018, 3:21 pm

(Aug 14, 2018 04:55 PM)PornFlakes Wrote: [\[CITE\]](#) Hello,

You can find 'hidden' endpoint by debugging the http traffic between your computer and the server, I recommend Fiddler for this or if you're willing to pay money I recommend Charles proxy.

If you need additional help I can help you out just send me a message. Hope it helped!
Can you PM me? I don't have the required 25 posts lol

vorticalbox - August 18, 2018, 2:34 am

(Aug 7, 2018 03:32 PM)c3ypt1c Wrote: [\[CITE\]](#) What are you talking about?

What endpoints?

Also, I recommend Selenium for Python to scrape.

Why? Unless you need JavaScript, the using requests / beautifulsoup is massively faster.

I would recommend robobrowser and learning to multi process.

v0dka - August 18, 2018, 7:00 am

i would recommend selenium as its alot less code to write much simplier, scrapy is another option but i find scrapy alot of work to get a end result with selenium you can see you code working and easily debug whats going wrong.

edit: forgot to mention with selenium you dont need to do any cookie management as its handled within the browser itself, however with scrapy,requests etc if your needing to manage cookies then you will have to write that in yourself

Nomad.Null - August 18, 2018, 10:19 am

Why would you use selenium? It will just slow down the requests. I never use selenium except in cases where you need to mimic browser-like behavior like mouse clicks in order to avoid some kind of WAF that detects bots. If you need to perform rate limiting, that's easy enough to do with threading and request management.

Finding endpoints in APIs isn't an exact science. It actually involves quite a bit of bruteforcing in

many cases, and simply looking for anything other than a typical 404 error. At that point, you have to parse errors and trial-and-error your way through the parameters. Just takes a lot of patience. To find the endpoints themselves, try using a dictionary-builder like cewl and dirbuster or gobuster to brute-force the endpoints. Make sure you are specifically searching for non-404 status codes.

vorticalbox - August 18, 2018, 1:24 pm

(Aug 18, 2018 10:19 AM)True Demon Wrote: [\[CITE\]](#) Why would you use selenium? It will just slow down the requests. I never use selenium except in cases where you need to mimic browser-like behavior like mouse clicks in order to avoid some kind of WAF that detects bots. If you need to perform rate limiting, that's easy enough to do with threading and request management.

Finding endpoints in APIs isn't an exact science. It actually involves quite a bit of bruteforcing in many cases, and simply looking for anything other than a typical 404 error. At that point, you have to parse errors and trial-and-error your way through the parameters. Just takes a lot of patience. To find the endpoints themselves, try using a dictionary-builder like cewl and dirbuster or gobuster to brute-force the endpoints. Make sure you are specifically searching for non-404 status codes.

The way I used to do it was to install MITTP (man in the middle proxy), point my browser to it and just use the target website, I comes with certs so that it can monitor HTTPS connections too. Super useful if you need to find out some API calls.

You can also do the same thing in chrome web tools.

Nomad.Null - August 18, 2018, 1:44 pm

(Aug 18, 2018 01:24 PM)vorticalbox Wrote: [\[CITE\]](#) The way I used to do it was to install MITTP (man in the middle proxy), point my browser to it and just use the target website, I comes with certs so that it can monitor HTTPS connections too. Super useful if you need to find out some API calls.

You can also do the same thing in chrome web tools.

You might benefit from learning Burp Suite. There is a free version (pro is over \$1,000 but only really worth it for dedicated pentesters) that has all the features necessary to analyze and manipulate http traffic very easily, including HTTPS connections. The pro has additional features and special plugins, but it's mostly reserved for brute-forcing post and form parameters, using macros to dynamically interact with pages that change based on form-only input rather than URI parameters, and also other handy stuff like SQL injection, XSS, CSRF, clickjacking, etc. If all you're doing is trying to work your way through a web API, the web proxy functionality is bar-none. ?

FlexagonV2 - August 21, 2018, 1:28 pm

(Aug 18, 2018 01:44 PM)True Demon Wrote: [\[CITE\]](#) You might benefit from learning Burp Suite. There is a free version (pro is over \$1,000 but only really worth it for dedicated pentesters) that has all the features necessary to analyze and manipulate http traffic very easily, including HTTPS connections. The pro has additional features and special plugins, but it's mostly reserved for brute-forcing post and form parameters, using macros to dynamically interact with pages that change based on form-only input rather than URI parameters, and also other handy stuff like SQL injection, XSS, CSRF, clickjacking, etc. If all you're doing is trying to work your way through a web API, the web proxy functionality is bar-none. ?

What's the difference between fiddler and burp suite?

wheelcog - August 23, 2018, 1:07 am

I'd use mitmproxy. It will work with both desktop and mobile, and supports several protocols (HTTP, HTTPS, Web Sockets). You can capture and replay your traffic to/from files, and edit/resend your requests. It also has a Python API that will allow you to write plugins or acting on traffic on the fly. The documentation even has a tutorial on how to cheat at mobile games.

wtftrick - September 11, 2018, 1:23 am

(Aug 18, 2018 07:00 AM)v0dka Wrote: [\[CITE\]](#) i would recommend selenium as its alot less code to write much simplier, scrapy is another option but i find scrapy alot of work to get a end result with selenium you can see you code working and easily debug whats going wrong.

edit: forgot to mention with selenium you dont need to do any cookie management as its handled within the browser itself, however with scrapy,requests etc if your needing to manage cookies then you will have to write that in yourself

Saving a session and keeping cookies across requests is literally one line of code with the requests library.

```
Code session = requests.Session()
```

Let's not mention how much less overhead there is (no chromedriver.exe, less memory and CPU usage). You should always use it over Selenium when you can.