Video on API Hacking by David Bombal: <https://www.youtube.com/watch?v=Clu3-5TFdw0>

A video on APISEC university’s course: <https://www.youtube.com/watch?v=CkVvB5woQRM>

APISEC university’s API security course: <https://university.apisec.ai/apisec-certified-expert>

This is a completely free course. The course has 13 categories. Each has a small quiz at the end:

1. Introduction
2. Lab Setup
3. API Reconnaissance
4. Endpoint Analysis
5. Scanning APIs
6. API Authentication Attacks
7. Exploiting API Authorization
8. Testing for Improper Assets Management
9. Mass Assignment
10. Server-Side Request Forgery
11. Injection Attacks
12. Evasion and Combining Techniques
13. Conclusion
14. **Introduction**

A basic introduction of the course and a bit about the “[Hacking APIs](https://suchitreddi.github.io/Notes/Hacking%20APIs%20book.pdf)” book written by the course instructor Corey Ball.

1. **Lab Setup**

**Note**: Detailed installation procedures will be in videos of this category

The setup category is about installing Kali Linux VM, and important tools used in this course.

Install kali prebuilt image for you VM. Update and upgrade it.

Burp Suite should be already installed. If not, install it. Install Autorize extension for it. It is used to perform automated authorization testing. This extension needs Jython to be installed. (Install the Jython standalone jar, then add it as a python environment in options under the extension tab of Burp Suite.)

We must use proxy in our browser for its traffic to be visible in Burp Suite. We can configure the proxy manually or install an extension like FoxyProxy and add proxies for burp suite and postman to easily switch proxies. (Proxies are, 127.0.0.1 with 8080 for BS, 5555 for PM)

Download CA certificate from <http://burpsuite> and add it to the browser.

Install postman, mitmproxy2swagger, git, docker-compose, docker.io, golang-go, jwt\_tool, kiterunner, Arjun, zaproxy and update OpenAPI support in it.

Install the testing labs crAPI and vAPI.

1. **API Reconnaissance**

The reconnaissance category deals with how to find APIs of an application. Passive Reconnaissance is done using the power of OSINT. We can search for API keys, credentials, .json web tokens left unsecured on the internet. Passive Reconnaissance is done with the help of google and git dorking, API directories, The Wayback Machine and Shodan. You can search for swagger.json files and export them to postman where you can further work on them, or search for unsupported things in an API using Wayback machine.

In Active Reconnaissance, we use tools like Nmap , gobuster, kiterunner, amass and devtools of a browser. You can use crAPI installed in previous module, to do you practice.

**Gobuster** is used to brute force websites for their sub links. We use dev tools(network) to copy a query as cURL from and analyze it further in postman.

1. **Endpoint Analysis**

The Endpoint analysis category is all about reverse engineering an API’s documentation. If an API doesn’t have any documentation and specification file, we will have to reverse engineer it based on our interactions with it.

When trying to document the APIs an application has, it can be done in two ways. We can document the API by manually collecting API requests using Postman. This method takes more time and focus but will give you a greater understanding of that API as you are manually looking into it. We can also build out an automatic API specification using mitmproxy2swagger which we installed before.

1. Using Postman:

There are two ways of using postman to reverse engineer an API.

1. Constructing the API requests manually.

This method can be cumbersome, but it allows you to add only the requests you want.

1. Proxy web traffic through postman and use it to capture requests.

This method is easier, but you will have to delete or ignore the requests you don’t need.

1. Using mitmproxy2swagger:
2. **I**