API SecurityPentest, a knowledge base

API(Application Programming Interface)
one application to communicate with another
without having to have direct interaction.
Ex – User of Application A communicate
& use data of Application B

Types?
REST (Representational State transfer) & SOAP(Simple object access protocol)
REST is an architecture where when a request is made through HTTP, RESTfull Api response is variety of formats such as HTML, XML JSON

SOAP – has more rules & constraints. Built with diffrent labguages & on diff platforms which leads to longer load times.

API Attack
Lets discuss how to find out Vulnerablities in API's

- Always have a wordlist(even ur own)
- Api Versioning
  - api/v3/users/1/edit exists then check for
  -/api/**v1/users/1/**edit also exists

Vulnerabltites to find out during attacking API
- Information Disclosure
- Authorization Issues
- Business Logic
- Insecure direct object references
 XXS and CSRF

Tools Used

Burp Intuder – Burp is a great tool which helps us test different api endpoints, Intruder allows us to add our own wordlist during attack
ffuf – If wordlist is lengthy
arjun – if you go blank with the api endpoint,Use It. it finds valid HTTP parameters with a huge default dictionary of 10,985 parameter names.
Postman – Postman is tailored to test API. We can import the api files for testing. Connect with Burp & both makes a best of testing API

Lab 1
Since we know what is an api call, what all the attacks been made on it, lets explore how to perform an attack. Here i have built a python application on flask f/w which stores defence related information of a country

lets run through the code

Here just imagine these were the secrete army missions conducted by a country, whose data must never be leaked for for what so ever purposes. But while the attacker gets to know he can sniff into an application which has all the defense data, what all can be done?

Basics – Versioning, /v1 /v2

He might check the burp suit and check the data there by /app /v1

attack1 - http://localhost:5001/api/v1/resources/opt/all

attack 2 - http://localhost:5001/api/v2/resources/opt?id=3

Mitigation for IDOR can be, Rather giving id as 1,2,3 we can provide non guessable random numbers
As we can see the vulnerablities here are **IDOR/BAC**(broken access control) We are able to see the data which even though we are not suppose to see it without logging in.

And for broken access control we can use an firewall which i'll give a demo on.

Lab 2 – Information Disclosure
Consider the user who enters a shop accidently sees the billing section and checks the url of the billing desktop browser & get the credit card details of all the customers.

Mitigation – Enforce message mediation policy at API gateway level to filter any data that is being returned from API call, which ensures no sensitive data is leaked.

Lab 3 – Mass Assignment
haha!! more of a guess work but ill keep it minimal.
Ma is sending a data (eg . JSON) to data model. This is done by guessing an object, reading docs, allows attackers to modify object properties they are not supposed to.

Usually only the following fields in your account settings:
Name + lastname
Username
Adress

but here we can see "accountType"

lets Run the application, wait whats this? How to change the method. Here comes the goldmine – POSTMAN

lets use postman to exploit the vulnerablity

Use the path , may give us leads,

we get the error but we have the parameters outlined there for us using which we can find something,
accountType seems to be fishy, lets remove the accountType and check for response

Now we get a different message that is saying that the account type can only be certain types but what if we try admin?

Whats wrong? The object "user" supports feild accountType. The developers should not allow the client UI to change this but it does which leads to privilege escalation from mass assignment bugs.

Mitigation – Whitelist the required words for the user and block the remaining.

Api Firewall

What ? Api firewall is a light-weight program to protect your end api endpoints in cloud-native environments with api schema validation, It has a security model which has a set of rules which allows calls that match api specifications while rejecting everything else. Efficient usage in Cloud native environments.

Built on?
Api firewall is built-in OpenAPI, written in Go.

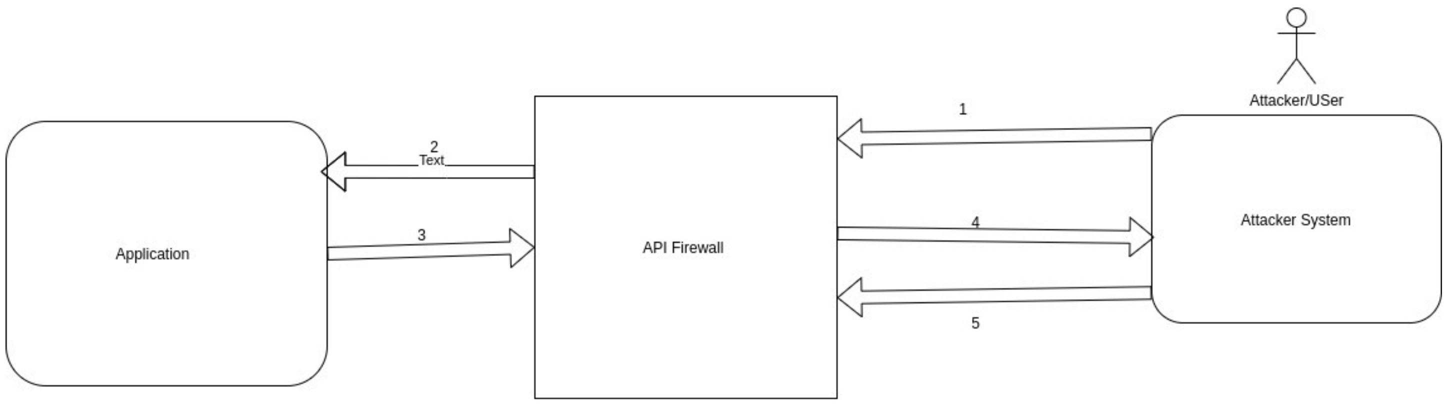
Pre req – Docker, Git, pip

- API firewall - https://hub.docker.com/r/wallarm/api-firewall
- We shall use a demo python application built on Connexion framework that auto handles HTTP requests based on OpenApi Sepcs written in YAML format

Lets Get our Hands Dirty!!

I have the initial set up done for us, installed the api-firewall in docker, have the python program, cd /tmp

- git clone https://github.com/zalando/connexion
- Install the firewall - docker pull wallarm/api-firewall



1 Sends API request

2 Forwards the request if it complies to the rules
3 Response by the Application

4 Response reaches the user
5 The malicious request gets blocked

Step 1 Run the api firewall with

```
docker run -d -v /tmp/connexion/examples/openapi3/methodresolver/openapi/:/tmp -e
APIFW_SERVER_URL=http://192.168.1.14:9090/v1.0/ -e APIFW_API_SPECS=/tmp/pets-api.yaml -e
APIFW_REQUEST_VALIDATION=BLOCK -e APIFW_RESPONSE_VALIDATION=BLOCK -p
8282:8282 wallarm/api-firewall
```

-d – run docker in background
-v mounts the volume of the YAML file in the open API
-e are req environment variables

This will run the docker container in the background and give us container id and in the last line we can see which YML file is specified

Step 2 - We shall run the python application

Step 3 - From the VM (attacker machine)we shall try to do the PUT request

curl -X PUT -H 'content-type: application/json' -d '{"name":"homyak-2", "tag":"aa"}' http://10.0.2.15:8282/v1.0/pets/3 gives us proper response result

lets deviate and think like attacker,

curl -X PUT -H 'content-type: application/json' -d '{"name":123123, "tag":"aa"}' http://10.0.2.15:8282/v1.0/pets/3 where the name parameter is integer , now lets check the result in the docker logs,

Step 4 – Lets check the YAML file for the rules being written
Here we can see that

According to the specified rules our API firewall will either be blocking or logging the requests being made
There's not much on the front end , lets check whats happening to docker logs

error for /name must be string , schema being used to request. Also since we need to defend our application .

So from these we can protect our application from api attacks by creating a simple and free api-firewall. Main advantage is we can prevent DoS attacks by creating and defining rules

Mitigation/Best Practice to prevent API attack

- Always Have a Threat Model
- Implement Oauth – a token based auth f/w that allows info to be access by 3rd party services without exposing user creds.
- Encrypt Data – Sensitive data such as user id, credit card details must be encrypted
- Use Rate Limiting & Throttling – To prevent DdoS
- Use API Firewall