1. Introduction

- This presentation aims to
    - Present an overview of APIs
    - Discuss why APIs are a unique security concern
    - Propose research into a framework for automated reconnaissance during API vulnerability testing

2. API Characteristics and Security

- Basic definition of APIs

- RESTful vs GraphQL APIs
    - How the APIs differ in structure, data, and endpoints
    - How each transfers info between apps

3. APIs as a Security Concern

APIs are vulnerable to several types of attacks:

-- Legacy Attacks
    - XSS
    - SQL injection
    - XML injection
    - CSRF
    - Command injection
    - HTTP verb tampering

- API Key defeat
    - API Gateway
    - HMAC
    - SSL/TLS Certificate Pinning
    - Restrictive Controls
    - Conditional Access
    - IP restrictions
    - admin access to IAM system
    - MFA

- Security Token defeat
    - OAuth 2.0
    - JWT

- Other Serious Attacks

- Dependency and namespace confusion
- IDORs
- Lack of rate limiting
- Broken access control
- Broken authentication
- Injection attacks
- Excessive Data Exposure

- API security is often neglected given the nature of API use / upkeep
  - because they are unseen, they can be neglected
    - not often updated
  - traditional IDS systems and penetration testing does not pick up on API-specific vulnerabilities

4. API Reconnaissance Automation

To aid in API-specific vulnerability testing, automatic reconnaissance can shorten / simplify finding the various aspects needed to perform API vulnerability testing alongside regular web application testing

- Passive scanning
  - possibilities to automate
    - google dorking
    - Nmap
    - web crawling

- Active scanning
  - possibilities to automate
    - endpoint testing
    - scanning for API characteristics
    - Targeted scanning
    - Robots.txt location
    - API validation
    - Crawling URLs
    - Brute forcing URIs
    - Content discovery

- the artefact would be
  - master program in Bash to be used with Kali Linux
  - supplementary programs will be in Python
  - third party open source apps may be utilised, if possible

- validation
  - will be tested on
    - a customized API vulnerability lab

- open source vulnerability web apps online (if applicable)
- websites which accept bug bounty hunting (if allowed)

5. Ethical Concerns

- When dealing with web app vulnerabilities, often research can be used for prevention or utilized by malicious actors for unintended purposes
  - Reconnaissance is a part of this, and an automation automation would be designed to make vulnerability testing faster – could do the same for malicious hacking

- Possible conflicts about how to test the web application
  - though web apps may be open to bug bounty hunting, it may not be ethical to access the sites for research

- Third Party Applications
  - Though third party apps for vulnerability testing are overwhelmingly open source, may still be unlawful usage to include them in an automation framework for vulnerability testing

6. Conclusions

- Tie the whole thing together

7. Sample References

Arnold, T & Seitz, J. (2021) Blackhat Python: Python Programming for Hackers and Pentesters. 2nd Ed. San Francisco, USA: No Starch Press.

Ball, C. J. (2022) Hacking APIS: Breaking Web Application Programming Interfaces. San Francisco, USA: No Starch Press.

Crawley, K., Wylie, P. L. (2021) The Pentester Blueprint: Starting a Career as an Ethical Hacker. Indianapolis, USA: Wiley.

Graham, D. G. (2021) Ethical Hacking: A Hands-On Introduction to Breaking In. San Francisco, USA: No Starch Press.

Li, V. (2021) Bug Bounty Bootcamp: The Guide to Finding and Reporting Web Vulnerabilities. San Francisco, USA: No Starch Press.

Khawaja, G. (2021) Kali Linux Penetration Testing Bible. Indianapolis, USA: Wiley.

Negus, C. (2020) Linux Bible: The Comprehensive Tutorial Servic. Indianapolis, USA: Wiley.

Pinto, M. & Stuttard, D. (2011) The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws. 2nd Ed. Indianapolis, USA: Wiley.

Shostack, A. (2014) Threat Modeling: Designing for Security. Indianapolis, USA: Wiley.