Secure APIs: Road to Business Growth

Anupama Natarajan

OWASP
The Open Web Application Security Project
About Me

• Senior Solutions Architect
• 15+ years experience
• Passionate with Data, Integration and Business Intelligence

https://www.linkedin.com/in/anupama-natarajan-516a107/

http://www.anupamanatarajan.com

https://twitter.com/@shantha05
Agenda

- Introduction to APIs
- API Security
- What are Underprotected APIs?
- Impacts of Underprotected APIs
- Examples of Underprotected APIs
- How to detect Underprotected APIs?
- How to protect Underprotected APIs?
- How do we design Secure APIs?
APIs Increase the Attack Surface

Traditional Web site or app

HTTP server

App server

Database

C58D75
F43A87
789A75
9C349E

Granularity boundary

API-based client app

HTTP server

App server

Database

C58D75
F43A87
789A75
9C349E

Granularity boundary

Reference: APIacademy
Underprotected APIs

- Core concern of modern Enterprises
- Increases the Attack Surface
- Breadth and Complexity of APIs makes it difficult to automate effective security testing
- Malicious APIs give attackers internal access to apps
• Technical Impacts
  – Data Theft
  – Data Corruption
  – Data Destruction

• Business Impacts
  – Denial of Service Attack on Critical API
  – Critical data compromised
  – Critical functions compromised
Underprotected APIs Examples

- WordPress REST API
  - Parameter Manipulation
- IoT Devices
  - Clear text data transmission
- Mobile App connecting to API
- Web Application connected to Database using API
Detect Underprotected APIs

- API Gateways (Apigee, Mulesoft, Azure API Management, CA Technologies, Red Hat [3scale])
- Metasploit
- ZAP (Zed Attack Proxy)
- POSTMAN, Insomnia REST Client
- Machine Learning and Analytics
• Not being in a rush
• Documentation
• Developers keeping Security in mind
• Web API tracing/testing tools
  – Fiddler (HTTP Requests)
  – Wireshark (Traffic capture & Analysis)
  – Metasploit Framework (Penetration Testing)
Secure API Design

• Validate Parameters e.g. sanitize incoming data
• Protect against injection of all forms
• Turn on TLS everywhere and enable SSL
• Implement rigorous Authentication and Authorisation Standards
• Separate API security and implementation as separate tiers
• Using Analytics to detect API usage patterns
• https://github.com/shieldfy/API-Security-Checklist
• https://www.owasp.org/index.php/OWASP_Zed_Attack_Proxy_Project
• https://www.metasploit.com/
• https://www.telerik.com/fiddler
• https://insomnia.rest/
• https://www.getpostman.com/
Thanks