Securing APIs with JWT & Backend Security

Faculty of Computing

Overview of API Security

- APIs are vulnerable to unauthorized access and data breaches
- Security is essential for user trust and data protection
- Key focus: authentication, authorization, data integrity

What is JWT (JSON Web Token)?

- Standard for securely transmitting information

- Three parts: Header, Payload, Signature

- Used for stateless authentication

Token-Based Authentication Using JWT

- User logs in -> server returns JWT

- Client stores and sends JWT in Authorization header

- Server verifies JWT to authenticate requests

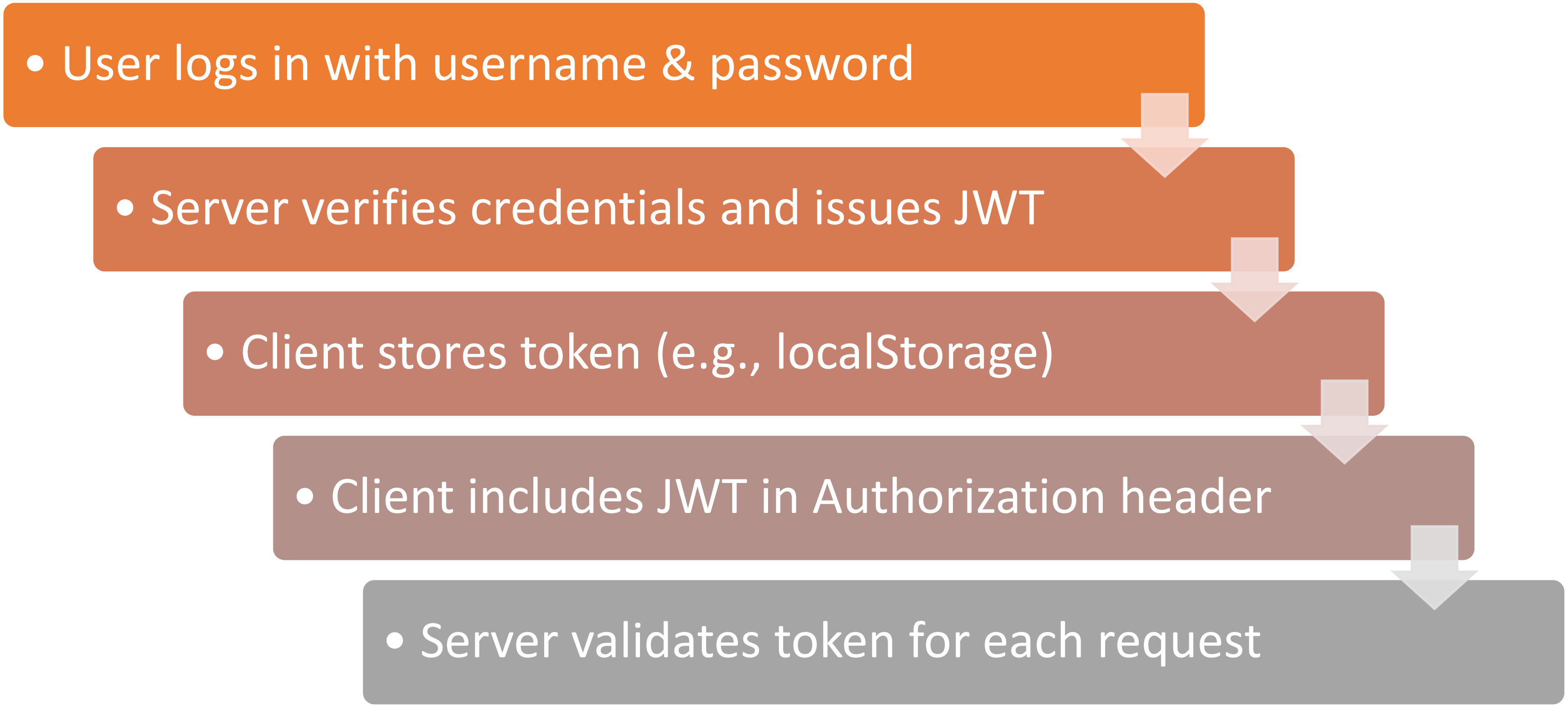
JWT Login & Token Validation

- Validate user credentials at login

- Generate token with secret key

- Use middleware to verify token for protected endpoints

JWT Authentication Flow



- User logs in with username & password
- Server verifies credentials and issues JWT
- Client stores token (e.g., localStorage)
- Client includes JWT in Authorization header
- Server validates token for each request

Protecting API Routes

- Middleware checks token before allowing access

- Invalid or expired token -> access denied

- Ensures only authenticated users access sensitive routes

Common Web Security Threats

- Cross-Site Scripting (XSS)
- SQL Injection
- Query Data Tampering
- Authentication Bypass

Cross-Site Scripting (XSS)

- Malicious scripts injected into trusted websites
- Can steal cookies, session tokens, or redirect users
- Use input sanitization and Content Security Policy (CSP)

SQL Injection & Query Tampering

- Attackers inject SQL via input fields or URLs
- Can read/modify/delete database records
- Use prepared statements and input validation

Authentication vs Authorization

- Authentication: Who are you? (Login)
- Authorization: What are you allowed to do? (Permissions)
- Use role-based access controls (RBAC)