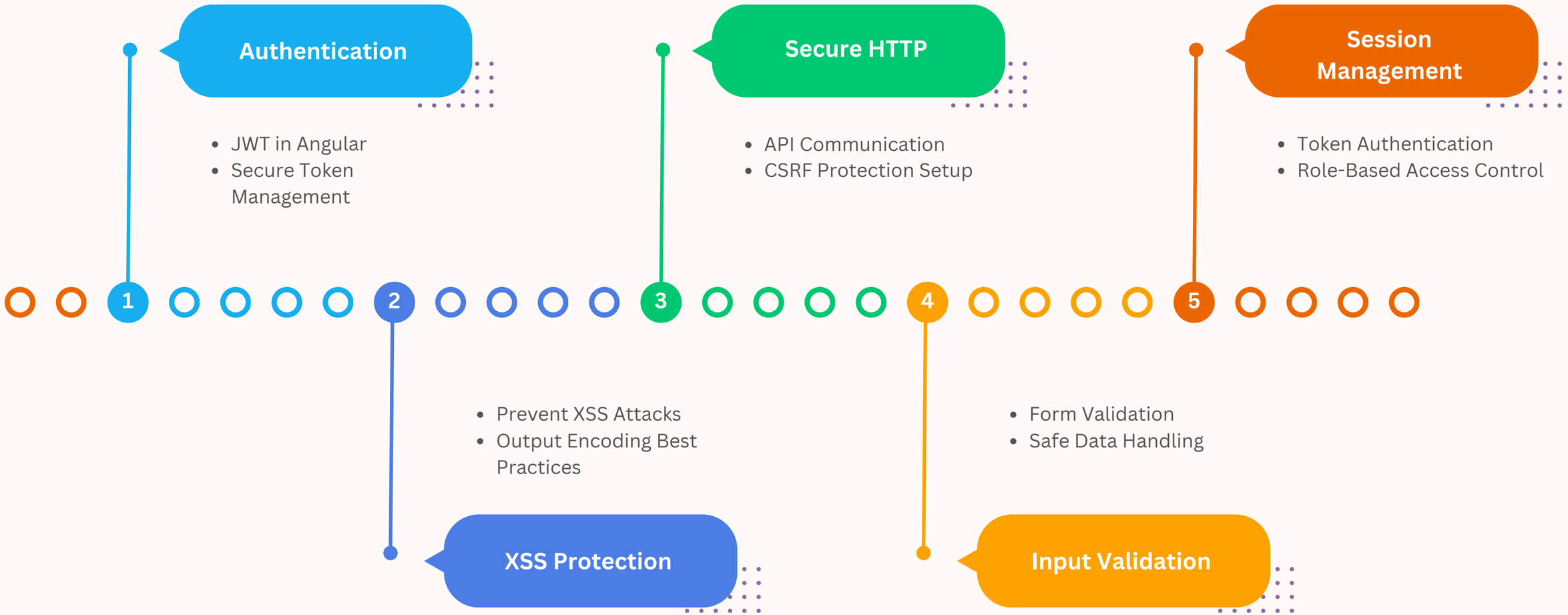
MASTER ANGULAR SECURITY



Master Angular Security

Deployment Security

- Enforce HTTPS
- Secure Build Configurations

6

Library Security

- Auditing Dependencies
- Safe Third-Party Integrations

7

Clickjacking Prevention

- Block Clickjacking
- CSP and HTTP Headers

8

Security Testing

- Penetration Testing
- Code Auditing and Tools

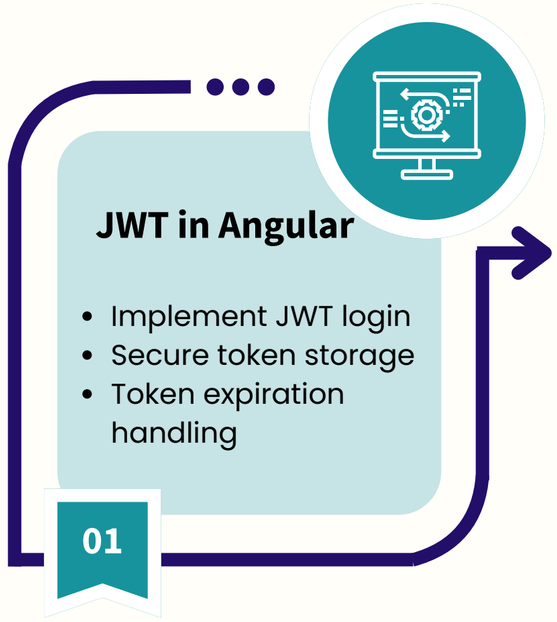
9

Secure App

- Build Secure App
- Implement Best Practices

10

Authentication



JWT in Angular

- Implement JWT login
- Secure token storage
- Token expiration handling

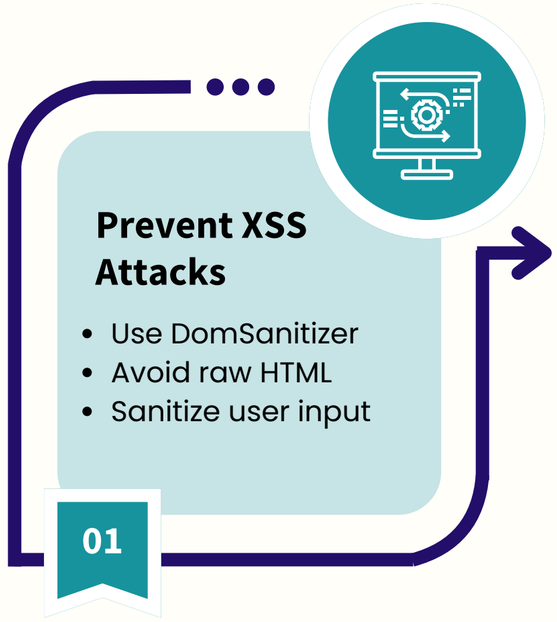
01

Secure Token Management

- Use HttpOnly cookies
- Token refresh logic
- Prevent token leakage

02

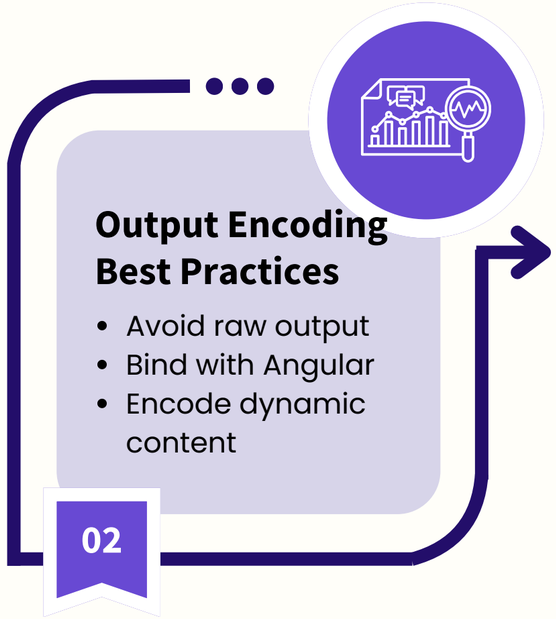
XSS Protection



Prevent XSS Attacks

- Use DomSanitizer
- Avoid raw HTML
- Sanitize user input

01

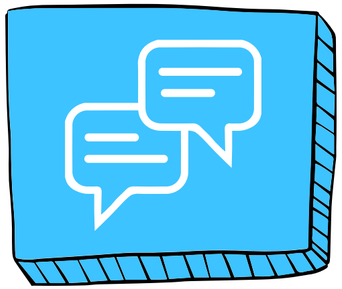


Output Encoding Best Practices

- Avoid raw output
- Bind with Angular
- Encode dynamic content

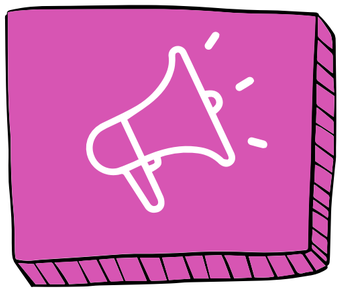
02

SECURE HTTP



API Communication

- Use HttpClient
- Add auth tokens
- Handle errors globally



CSRF Protection Setup

- Include CSRF tokens
- Backend CSRF validation
- Set CSRF headers

Input Validation

Form Validation

- Use Reactive Forms
- Set required validators
- Handle form errors

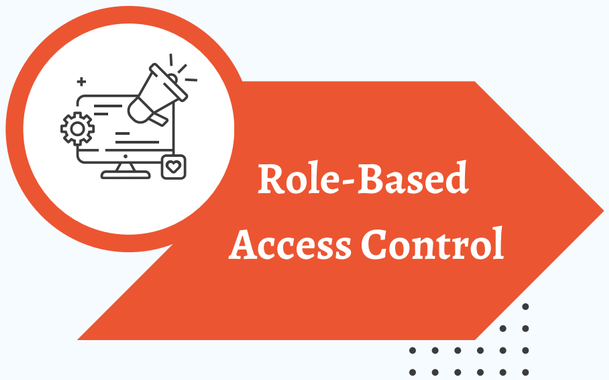
Safe Data Handling

- Sanitize user input
- Validate form data
- Restrict unsafe inputs

Session Management

Token Authentication

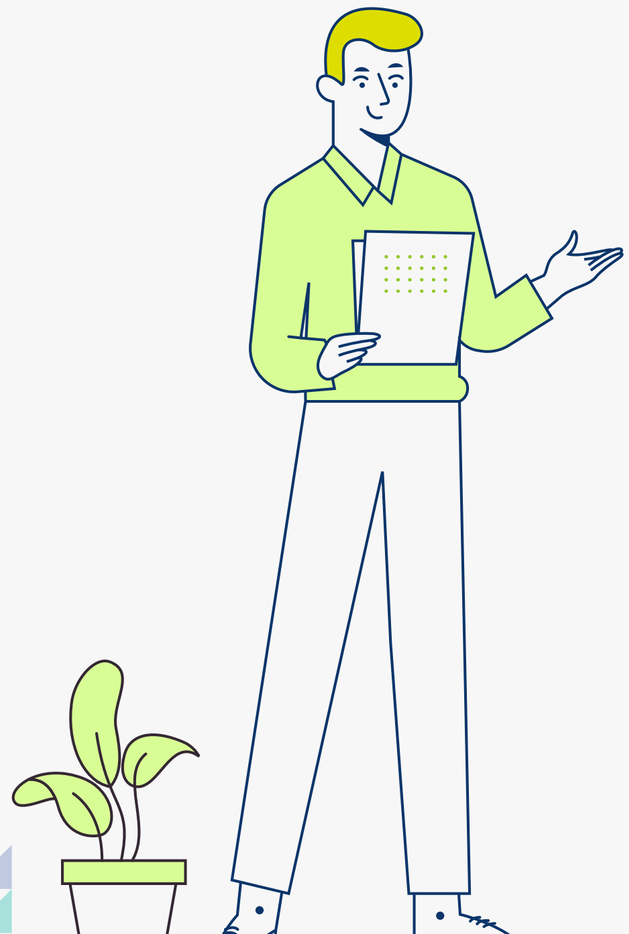
- Implement JWT authentication
- Secure token storage
- Token-based validation



Role-Based Access Control

- Create user roles
- Protect routes with guards
- Implement role validation

Deployment Security



Secure Build Configurations

- Enable production optimizations
- Minify and obfuscate code
- Use environment variables

Enforce HTTPS

- Set up SSL
- Force HTTPS redirection
- Enable HSTS

Library Security

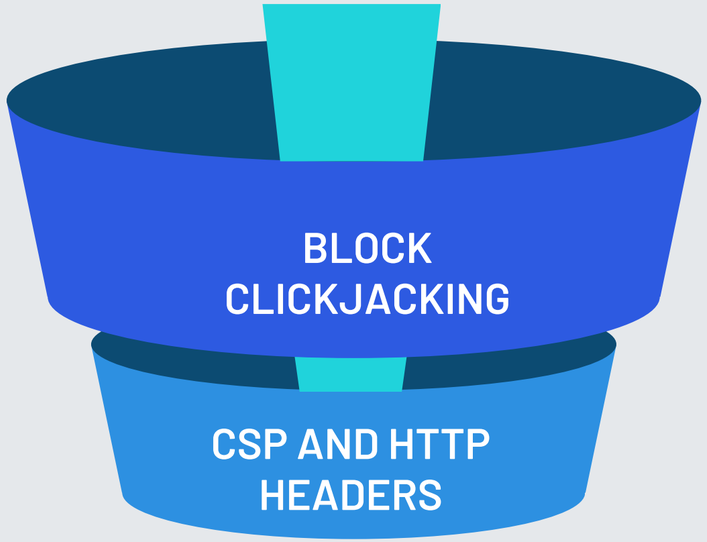
Auditing Dependencies

- Run **npm audit**
- Update outdated packages
- Check for vulnerabilities

Safe Third-Party Integrations

- Use trusted libraries
- Limit third-party features
- Test libraries before use

Clickjacking Prevention

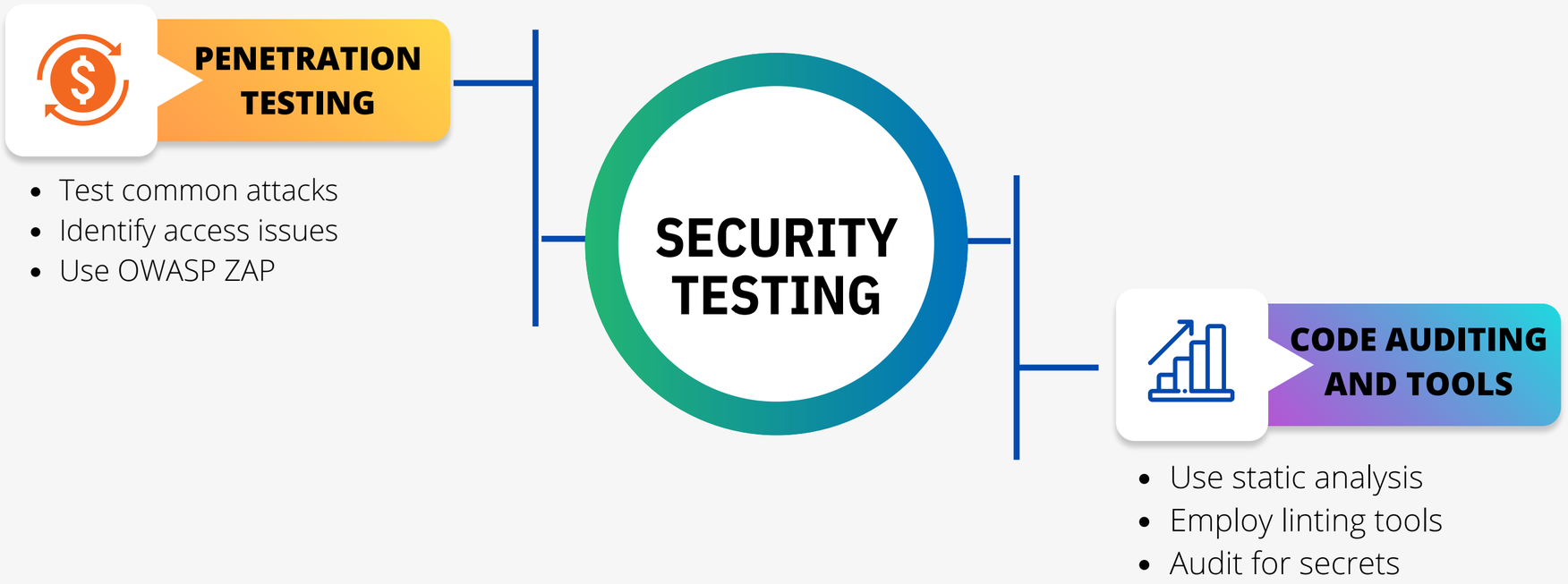


Block Clickjacking

- Set X-Frame-Options
- Disable iframe embedding
- Use Content-Security-Policy

CSP and HTTP Headers

- Configure CSP headers
- Set security headers
- Enable script nonce


PENETRATION TESTING
Test common attacks
Identify access issues
Use OWASP ZAP
SECURITY TESTING
CODE AUDITING AND TOOLS
Use static analysis
Employ linting tools
Audit for secrets

Secure App

01

Build Secure App

- Implement strong authentication
- Secure data encryption
- Use HTTPS communication

02

Implement Best Practices

- Follow OWASP guidelines
- Conduct security audits
- Apply secure coding practices