Authentication Vulnerabilities Testing Checklist & Methodology

Eyad Islam El-Taher

February 2, 2026

Abstract

This document provides a comprehensive checklist and methodology for testing authentication vulnerabilities. It serves as a practical guide for security professionals, penetration testers, and developers to systematically identify and remediate authentication-related security flaws.

Testing Methodology Overview

Phase 1: Reconnaissance

Information Gathering

1. **Identify authentication endpoints**
   - Login pages (`/login`, `/signin`)
   - Registration pages (`/register`, `/signup`)
   - Password reset (`/forgot-password`, `/reset-password`)
   - Password change (`/change-password`)
   - Multi-factor authentication endpoints
2. **Analyze authentication mechanisms**
   - Form-based authentication
   - Basic authentication
   - Token-based authentication (JWT, OAuth)
   - Multi-factor authentication
3. **Identify user input parameters**
   - Username/email parameters
   - Password parameters
   - Session tokens
   - CSRF tokens
   - Verification codes
4. **Gather user information**
   - Public profiles for username enumeration
   - Error messages revealing information
   - Email addresses in HTTP responses

Phase 2: Username Enumeration Testing

Username Enumeration Checklist

1. **Error message analysis**
   - Submit invalid username + invalid password
   - Submit valid username + invalid password
   - Compare error messages for differences
   - Check HTTP status codes (should be identical)
2. **Response time analysis**
   - Test with invalid usernames (measure response time)
   - Test with valid usernames (measure response time)
   - Use long passwords to amplify timing differences
   - Use Burp Intruder `Response received` & `Response completed`
3. **Registration form testing**
   - Test if "username already exists" message appears
   - Check email registration for enumeration
   - Test account recovery for user existence disclosure
4. **Public information sources**
   - Check user profiles for usernames
   - Search for email addresses in source code
   - Check API responses for user information
   - Review comments in HTML/JavaScript

Phase 3: Password Brute-Force Testing

Brute-Force Testing Checklist

1. **Basic brute-force testing**
   - Test with common password lists
   - Test default credentials (`admin:admin`, `root:root`)
   - Test password patterns (season+year, companyname+123)
   - Test password variations (`Password1!`, `Password2!`)
2. **Protection mechanism testing**
   - Test account lockout mechanisms
     - Number of attempts before lockout
     - Lockout duration
     - Automatic vs manual unlock
   - Test IP-based rate limiting
     - Attempts per time window
     - Block duration
     - CAPTCHA implementation
3. **Protection bypass testing**
   - Test if successful login resets counter
   - Test credential interspersing attack
   - Test X-Forwarded-For header for IP spoofing
   - Test array parameter attacks (`password[]=pass1&password[]=pass2`)
   - Test race condition attacks
4. **Advanced techniques**
   - Use Burp Intruder with macros for multi-step attacks
   - Use Turbo Intruder for high-speed attacks
   - Test with pitchfork attacks for multi-parameter brute-force
   - Configure resource pools for sequential execution

Phase 4: Multi-Factor Authentication Testing

2FA/MFA Testing Checklist

1. **2FA bypass testing**
   - Test if 2FA can be skipped entirely
   - Access protected pages after first factor only
   - Check session state after first authentication
   - Test if 2FA verification is enforced on all endpoints
2. **2FA code testing**
   - Test default codes (`000000`, `123456`)
   - Test null/empty codes
   - Test code reuse (previously used codes)
   - Test cross-account code reuse
   - Test rate limiting on 2FA attempts
3. **2FA logic flaws**
   - Test cookie manipulation between steps
   - Test if user can be changed between 1st and 2nd factor
   - Test response manipulation (`false` $\rightarrow$ `true`)
   - Test if 2FA tokens are validated on submission
4. **Integration testing**
   - Test if password reset bypasses 2FA
   - Test if OAuth login bypasses 2FA
   - Test if 2FA enrollment requires verification
   - Test session persistence after 2FA enablement

Phase 5: Additional Authentication Functionality

Supplementary Features Checklist

1. **Password reset testing**
   - Test token predictability
   - Test token validation on form submission
   - Test token expiration
   - Test password reset poisoning via headers
   - Test if username can be changed in reset form
   - Test rate limiting on reset requests
2. **Password change testing**
   - Test if username in hidden field can be modified
   - Test error message differences for password enumeration
   - Test if current password is properly verified
   - Test if old sessions are invalidated
3. **"Remember me" functionality**
   - Analyze remember me token structure
   - Test token predictability
   - Test token brute-forceability
   - Test if token contains password/hash
   - Test cookie security flags
4. **Session management**
   - Test session fixation
   - Test concurrent session handling
   - Test session timeout
   - Test logout functionality
   - Test session invalidation on password change

Wordlists & Resources

- **Username lists:**
  - SecLists/Usernames
  - Common admin usernames
  - Company-specific patterns
- **Password lists:**
  - rockyou.txt
  - SecLists/Passwords
  - CrackStation wordlist
  - Custom generated lists
- **Default credentials:**
  - Default vendor credentials
  - Common admin:admin combinations
  - Default device credentials