INTERMEDIATE LEVEL

Advanced Authentication Vulnerability Concepts

1. Authentication Bypass Techniques

- **Session Management Flaws: Predictable session IDs, session fixation**
- **Token Weaknesses: JWT vulnerabilities, insecure token storage**
- **MFA Bypass Methods: SMS interception, social engineering**
- **Authentication Logic Flaws: Race conditions, parallel authentication attempts**
- **Cookie Manipulation: Insecure cookie handling, missing security flags**

2. Sophisticated Credential Attacks

- **Credential Stuffing with Proxy Rotation: Avoiding IP detection**
- **Password Cracking: Using specialized hardware (GPUs) to break hashed passwords**
- **Pass-the-Hash Attacks: Using stolen password hashes without knowing plaintext**
- **Replay Attacks: Capturing and resending authentication traffic**
- **Man-in-the-Middle: Intercepting authentication communications**

3. Identity Federation Vulnerabilities

- **SAML Implementation Flaws: XML signature wrapping attacks**
- **OAuth Weaknesses: Improper implementation, insecure redirect handling**
- **OpenID Connect Vulnerabilities: Token validation issues**
- **Cross-Domain Authentication Problems: Origin validation failures**
- **Trust Relationship Exploitation: Identity provider compromise**

Complex Real-World Examples

1. JWT Authentication Flaws

- **None Algorithm Attacks: Setting algorithm to "none" to bypass signature validation**
- **Key Confusion: Using the wrong key type for verification**
- **Weak Secret Keys: Using guessable or brute-forceable secrets**
- **Missing Signature Verification: Not validating signatures at all**
- **Information Disclosure: Sensitive data in JWT payload**

2. Single Sign-On Vulnerabilities

- **Improper Audience Validation: Accepting tokens meant for different services**
- **XML External Entity (XXE) in SAML: XML parsing vulnerabilities**
- **Insufficient Token Validation: Missing checks on issuer, expiration, or signature**

- **Cross-Site Request Forgery in OAuth: Forced authorization through CSRF**
- **Open Redirectors: Redirect_uri validation issues**

3. MFA Implementation Weaknesses

- **SIM Swapping Vulnerabilities: Mobile carrier account takeover**
- **Time-Based OTP Synchronization Issues: Clock drift, replays**
- **Account Recovery Bypassing MFA: Weak recovery flows negating MFA benefits**
- **Social Engineering Against MFA: Phishing for OTP codes**
- **Push Notification Fatigue: Users approving unwanted authentication attempts**

Intermediate Detection Methods

1. Enhanced Monitoring and Analytics

- **User Behavior Analytics (UBA): Detecting anomalous login patterns**
- **Risk-Based Authentication Scoring: Evaluating login risk factors**
- **Geographic Impossible Travel Detection: Identifying physically impossible login locations**
- **Device Fingerprinting: Recognizing new or suspicious devices**
- **Login Velocity Analysis: Detecting automation based on timing patterns**

2. Advanced Testing Techniques

- **Authentication Fuzzing: Manipulating authentication parameters**
- **Session Handling Tests: Session ID analysis, cookie inspection**
- **MFA Bypass Testing: Attempting to circumvent second factors**
- **Federation Security Testing: SAML/OAuth implementation testing**
- **Password Recovery Flow Analysis: Examining recovery mechanisms**

3. Code and Configuration Review

- **Framework-Specific Authentication Reviews: Looking for known vulnerabilities**
- **Custom Authentication Analysis: Reviewing bespoke implementations**
- **Password Hashing Verification: Checking for proper algorithms (Argon2, bcrypt)**
- **Token Handling Inspection: Verifying secure token management**
- **Cross-Service Authentication Flows: Examining trust relationships**

Intermediate Prevention Strategies

1. Enhanced Authentication Architecture

- **Defense in Depth: Multiple authentication checks**
- **Adaptive Authentication: Risk-based additional factors**

- **Centralized Authentication Services: Single source of truth**
- **Certificate-Based Authentication: Using client certificates**
- **API Security Tokens: Properly implemented OAuth/JWT**

2. Secure Development Practices

- **Authentication Design Patterns: Following established models**
- **Authentication Libraries: Using vetted implementations**
- **Input Validation: Strict parameter checking**
- **Secure Password Recovery: Time-limited, single-use tokens**
- **Rate Limiting: Preventing brute force attempts**

3. Advanced MFA Implementation

- **FIDO2/WebAuthn: Phishing-resistant authentication**
- **Hardware Security Keys: YubiKey, Google Titan**
- **Biometric Authentication: When implemented securely**
- **Time-based One-Time Passwords (TOTP): Authenticator apps**
- **Context-Aware Authentication: Location, device, behavior factors**