Perfect, let’s now cover **Testing for Bypassing Authentication Schema** 🔐 — a classic **WSTG** (Web Security Testing Guide) test case.

**🧾 Testing for Bypassing Authentication Schema**

**1. Why It Matters**

Authentication schemas are meant to control **who** can access what.  
If bypassed, attackers can:

* Skip login altogether.
* Access restricted pages directly.
* Impersonate another user.

**2. Common Weaknesses to Test**

**🔸 Direct Page Access (Forced Browsing)**

* Try visiting restricted URLs directly, e.g.:
* /admin
* /dashboard
* /config
* If the app doesn’t re-check authentication, you bypass login.

**🔸 Parameter Manipulation**

* Example:
* login.php?admin=false
* → change to admin=true
* Or cookies like role=user → change to role=admin.

**🔸 Session Fixation / Token Manipulation**

* Capture the **session cookie** in Burp.
* Modify it to see if the server accepts **predictable / weak tokens**.
* Example: sequential PHPSESSID=12345.

**🔸 Default / Hidden Credentials**

* Sometimes apps ship with default logins:
* admin/admin
* root/password
* Hidden “backdoor” accounts may exist.

**🔸 Authentication Logic Flaws**

* Try **skipping login requests**: send only part of the flow (e.g., skip password validation).
* Sometimes **only username check is enforced**.

**🔸 Alternative Authentication Paths**

* Access **mobile APIs**, debug endpoints, or legacy URLs.
* /api/v1/login
* /old/login.jsp
* These might enforce weaker or no checks.

**3. Testing with Burp Suite**

1. **Proxy**
   * Capture login flow.
   * Look at parameters (cookies, tokens, headers).
2. **Repeater**
   * Modify requests (e.g., delete password field, change role).
   * Replay and analyze server response.
3. **Intruder**
   * Brute-force cookies, tokens, or role parameters.
   * Example: fuzz Authorization: Basic <payload>.
4. **Comparer**
   * Compare authenticated vs. unauthenticated responses.
   * Look for differences in headers, content length, or status codes.

**4. Example Attack Scenarios**

* **URL Manipulation**
* /profile?user=123
* → change to /profile?user=124

If it shows another user’s data → broken auth.

* **Weak Cookie Authentication**
* Cookie: user=admin

If accepted → bypass successful.

* **API Misconfiguration**  
  Mobile API allows login without password but web UI requires it.

**5. Mitigations**

✅ Enforce **authentication checks on every request** (not just login).  
✅ Use **strong, unpredictable session tokens**.  
✅ Implement **role-based access control (RBAC)**.  
✅ Disable **default credentials**.  
✅ Harden APIs and legacy endpoints.

✅ **Summary**:  
When testing for **Authentication Schema Bypass**, focus on **direct access, parameter tampering, weak sessions, default creds, and alternate entry points**. Burp Suite makes it easy to intercept, modify, and replay requests to find logic flaws.

👉 Do you want me to make a **step-by-step challenge lab exercise** where you bypass a fake login in Burp Suite (e.g., parameter tampering → gain admin access)?