

Day 5: HTTP, HTTPS, Cookies – Full Explanation (Hindi + English)

1. HTTP (Hypertext Transfer Protocol)

◇ क्या है?

HTTP एक request-response protocol है जो web browser (client) और web server के बीच डेटा भेजने के लिए इस्तेमाल होता है।

◇ कैसे काम करता है?

जब आप किसी वेबसाइट पर जाते हैं (http://example.com)

- आपका browser HTTP request भेजता है
- Server उसे process करता है और HTTP response भेजता है (HTML, images, etc)

◇ Example:

```
GET /index.html HTTP/1.1
Host: example.com
User-Agent: Mozilla/5.0
```

Server से response:

```
HTTP/1.1 200 OK
Content-Type: text/html
Content-Length: 1234
```

◇ Ethical Hacking में उपयोग:

- Packet sniffing: HTTP data plaintext में होता है – Wireshark, tcpdump से capture किया जा सकता है
- Login credentials capture: अगर HTTP पर login हो रहा है, तो credentials network में दिख जाते हैं

2. HTTPS (HTTP Secure)

◇ क्या है?

HTTPS = HTTP + SSL/TLS — यह encryption provide करता है जिससे data secure रहता है।

◇ कैसे काम करता है?

- SSL Certificate install होता है server पर
- Browser और server SSL handshake करके data encrypt करते हैं

- कोई attacker बीच में data नहीं पढ़ सकता (Man-in-the-middle attack fail)
- ◇ URL:
- https://example.com
- ◇ Ethical Hacking में उपयोग:
- HTTPS encrypt करता है data, इसलिए attacker को sniffing में सिर्फ encrypted data मिलेगा
 - HTTPS को bypass करने के लिए attacker को MITM setup, custom certificate, या proxy (Burp CA install) करना पड़ता है

3. HTTP Methods (Web Actions)

- ◇ Common Methods:

Method	Use	Example
GET	Page retrieve	Search, image fetch
POST	Form submit	Login, Signup
PUT	Update data	API update
DELETE	Delete data	API delete
OPTIONS	Check available methods	Preflight check

- ◇ Example – Login POST Request:

```
POST /login.php HTTP/1.1
Host: example.com
Content-Type: application/x-www-form-urlencoded

username=admin&password=admin123
```

- ◇ Ethical Hacking में उपयोग:
- Burp Suite से request capture करके data modify किया जाता है
 - XSS, SQLi attacks POST/GET में embedded होते हैं

4. Cookies (Client-side storage)

- ◇ क्या है?

Cookie एक छोटी सी data file होती है जो browser के अंदर store होती है।

◇ Use:

- Login session maintain करना
- User preference save करना
- Tracking और analytics

◇ Set-Cookie Example (from Server to Client):

```
Set-Cookie: session_id=abc123; HttpOnly; Secure; Path=/; Expires=Wed, 10 Jul 2025 12:00:00 GMT
```

◇ Browser से दिखना:

F12 (Dev Tools) → Application Tab → Cookies section

◇ Ethical Hacking में उपयोग:

- Cookie theft: अगर XSS मिला तो document.cookie से cookie चुरा सकते हैं
- Session Hijacking: अगर attacker को session_id मिल जाए तो वो उसी user के रूप में act कर सकता है
- Insecure Cookies: अगर cookies पर Secure और HttpOnly flags नहीं लगे तो attacker उसे intercept कर सकता है

🔒 5. Cookie Flags

Flag	Meaning
Secure	केवल HTTPS पर भेजी जाती है
HttpOnly	JavaScript access नहीं कर सकता
SameSite=Strict	Cross-site request नहीं होती
Expires	Expiry date set करता है

🔗 6. Practical Examples You Should Try

🔗 Using Dev Tools:

1. Visit <http://testphp.vulnweb.com>
2. Open Developer Tools → Network → Refresh the page
3. Click on any request:
 - Check Method (GET/POST)
 - Check Headers (Cookies, Host, User-Agent)
 - Check Response (Set-Cookie)

🔗 Burp Suite (for Intercepting HTTP/HTTPS):






1. Set Burp Proxy → Browser → 127.0.0.1:8080

2. Enable Intercept → Login to test site
3. Modify POST data in Burp → Send to Repeater
4. Try fake login, try SQLi payloads in POST data

Cheat Sheet Summary

Term	Use	Tool
HTTP	Web request/response	Browser, Wireshark
HTTPS	Secure HTTP	SSL Certificates
Cookie	Session/Preference	DevTools, Burp
GET	Data fetch	Web page load
POST	Data submit	Forms, login
Session Hijack	Cookie theft	XSS, MitM
Modify Request	Tampering	Burp Suite, Postman

Practice Task for You

-  DevTools में जाओ – cookie और request header analyze करो
-  Burp Suite से HTTP request intercept करो
-  Try to find missing cookie flags: HttpOnly, Secure
-  POST request में username बदलकर देखो क्या होता है
-  XSS injection try करो: `<script>alert(1)</script>` comment box में