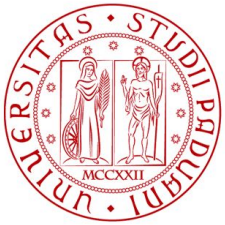
Ethical Hacking

2021-2022

Lesson 8: Ingredients of Web

Teaching Assistants
Luca Pajola
pajola@math.unipd.it.
Pier Paolo Tricomi
tricomi@math.unipd.it



UNIVERSITÀ DEGLI STUDI DI PADOVA

SPRITZ SECURITY & PRIVACY RESEARCH GROUP

DIPARTIMENTO MATEMATICA

Preliminaries

Client - Server Paradigm

- What's behind the magic stuffs that we use today?

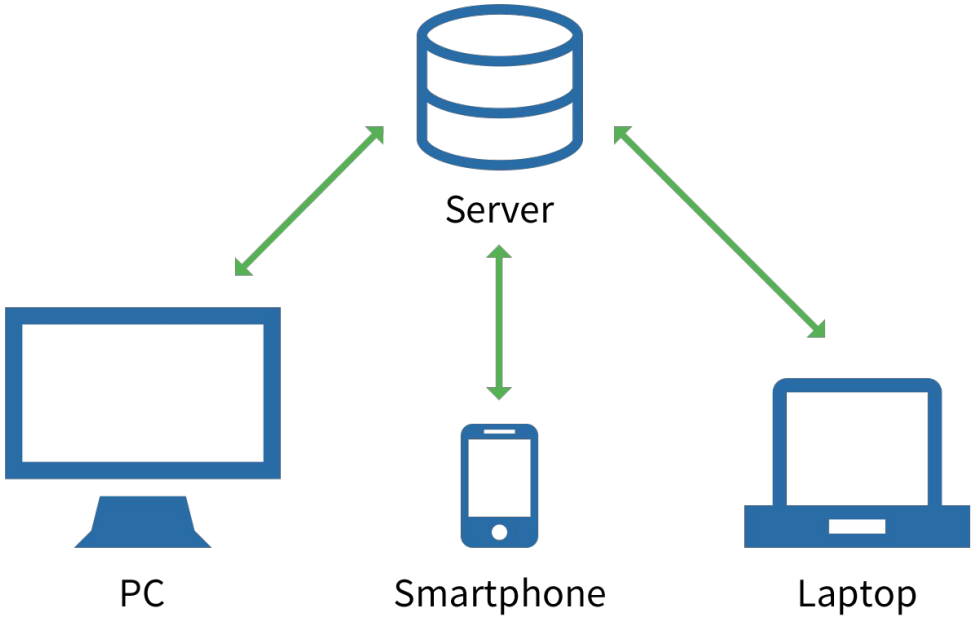
- Classic architecture **client-server**
  - Client: a device that requests for a service
  - Server: who provide a service
  - There are also other paradigms

- Each device has a **unique IP address**
  - Essential for the identification

- A device might have **multiple communications**
  - The role of *PORTS*

- Devices communicates through **protocols**
  - Define some *rules*

Client - Server Paradigm



Client - Server Paradigm

- Netflix is a good example
  - Multiple way to retrieve the service
    - Via browser
    - Via APP

General Notation

- Web application usually are accessed by browsers
  - Chrome, Firefox, Explorer
  - The layout is handled by HTML
  - HTML **is not** a programming language

- Usually some programming languages helps
  - They help for running algorithms / user interactions
  - JavaScript (client side)
  - PHP, Python (server side)

HTTP

***HTTP**: application-layer protocol for transmitting hypermedia documents, such as HTML*

HTTP

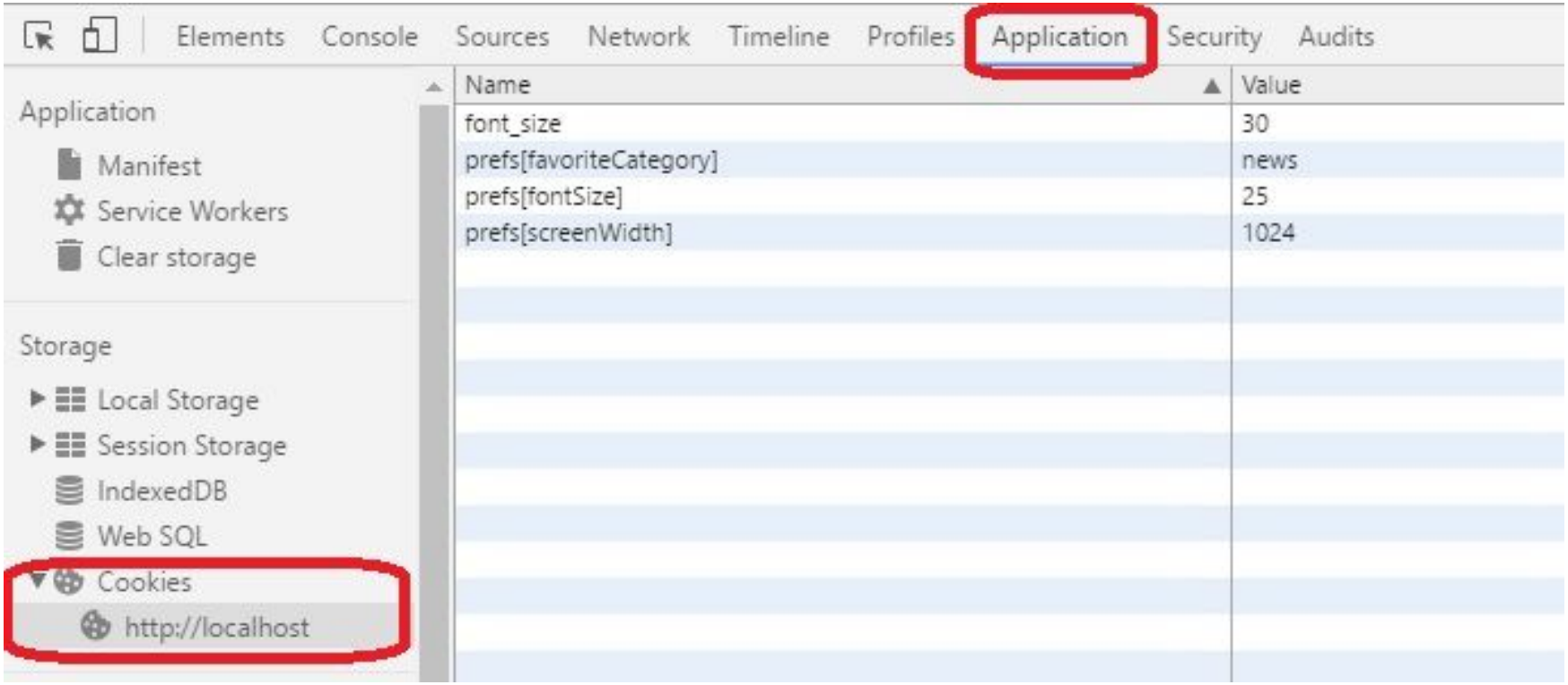
- What about security?
  - **HTTPS**: secure version of http obtained by making http *using a TLS connection between two hosts*

- TLS guarantees confidentiality, data integrity, server authentication, and resistance to several specific attacks while data are being transmitted over the network

Cookies

- “Hey, how can my browser open my Facebook page without the login phase?”
- Since HTTP protocol is stateless, browser stores the so called “cookies”
- Cookies maintain some info (e.g., login sessions)
- A cookie is made by several ingredients, such as:
  Name (mandatory), value (mandatory),
  expiry, path, domain, need for a secure connection
- Cookies can be harmful
  - Session hijacking or cookie hijacking

Cookies


Elements
Console
Sources
Network
Timeline
Profiles
Application
Security
Audits
Application
Manifest
Service Workers
Clear storage
Storage
Local Storage
Session Storage
IndexedDB
Web SQL
Cookies
http://localhost
Name
Value
font_size
30
prefs[favoriteCategory]
news
prefs[fontSize]
25
prefs[screenWidth]
1024

Cookies


1. The browser requests a web page
2. The server sends the page and the cookie
The cookie
Hello World!
3. The browser requests another page from the same server
The cookie
Web browser
Web server

Client Requests

- Clients requests stuffs to the server with some methods
  - e.g., **GET, POST**, PUT, HEAD, DELETE, PATCH, OPTIONS

- GET is used to request data from a specified resource
  - e.g., */test/demo.php?name1=value1&name2=value2*
  - The query string is sent in the URL of a GET request

- POST is used to send data to a server to create/update a resource
  - Data sent is stored in the request body of the HTTP request
- An attacker might play with these fields ...

Input Validation

- Applications usually expected some inputs
  - e.g., a calculator expects some numbers
- We must check the input that our application receives
- This process is called **input validation and sanitization**
  - The app processes only feasible inputs, rejecting non feasible ones
- Where should we put these stuffs?
  - Client side: can be easily bypassed (e.g., if based on JS)
  - Server side: increase the server's overhead

Exercises Session

1. Break the security AJAX authentication mechanism to reach the flag

2. Do you like sweets?

3. I heard you are good at breaking codes, can you crack this vault?

Questions? Feedback? Suggestions?