

CSE480: Web Security

2. Security & Web Programming Basics

Seongil Wi

Class on 9/15: Online Class

- There will be **NO** offline class on September 15
- Instead, a recorded lecture video will be provided (at Blackboard)
 - This video will be available **ONLY** on September 15

Sep 8 Tue	Security & Web Programming Basic	Slides	NEXT
Sep 10 Thu	Lab #1 HTML, JavaScript, and PHP	Lab Homepage	
Sep 15 Tue	SQL Injection	Online Slides	No offline class A recorded lecture video will be provided
Sep 17 Thu	Lab #2 SQL Injection	Lab Homepage	Assignment 1 out (Optional) Lab #3: SQL 101
Sep 22 Tue	Lab #4 Advanced SQL Injection	Lab Homepage	

Class on 9/17 & 9/22: Lab Sessions

3

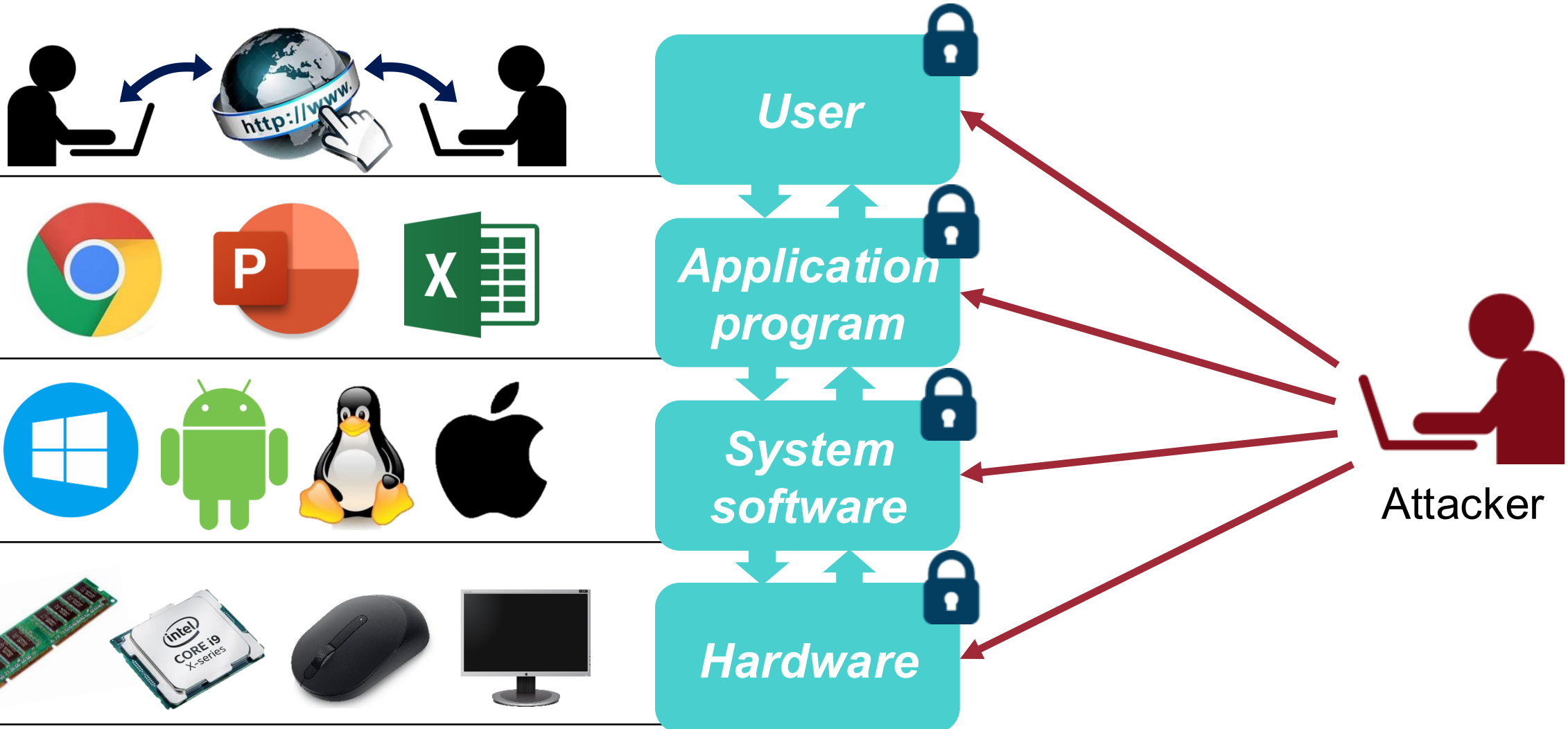
- There will be **two consecutive lab sessions** on SQL Injection on September 17 (Thu.) and September 22 (Tue.)

Sep 8 Tue	Security & Web Programming Basic	Slides	NEXT
Sep 10 Thu	Lab #1 HTML, JavaScript, and PHP	Lab Homepage	
Sep 15 Tue	SQL Injection	Online Slides	No offline class A recorded lecture video will be provided
Sep 17 Thu	Lab #2 SQL Injection	Lab Homepage	Assignment 1 out (Optional) Lab #3: SQL 101
Sep 22 Tue	Lab #4 Advanced SQL Injection	Lab Homepage	

Security Basics

Computer Security

The protection of **computer systems** from unauthorized access



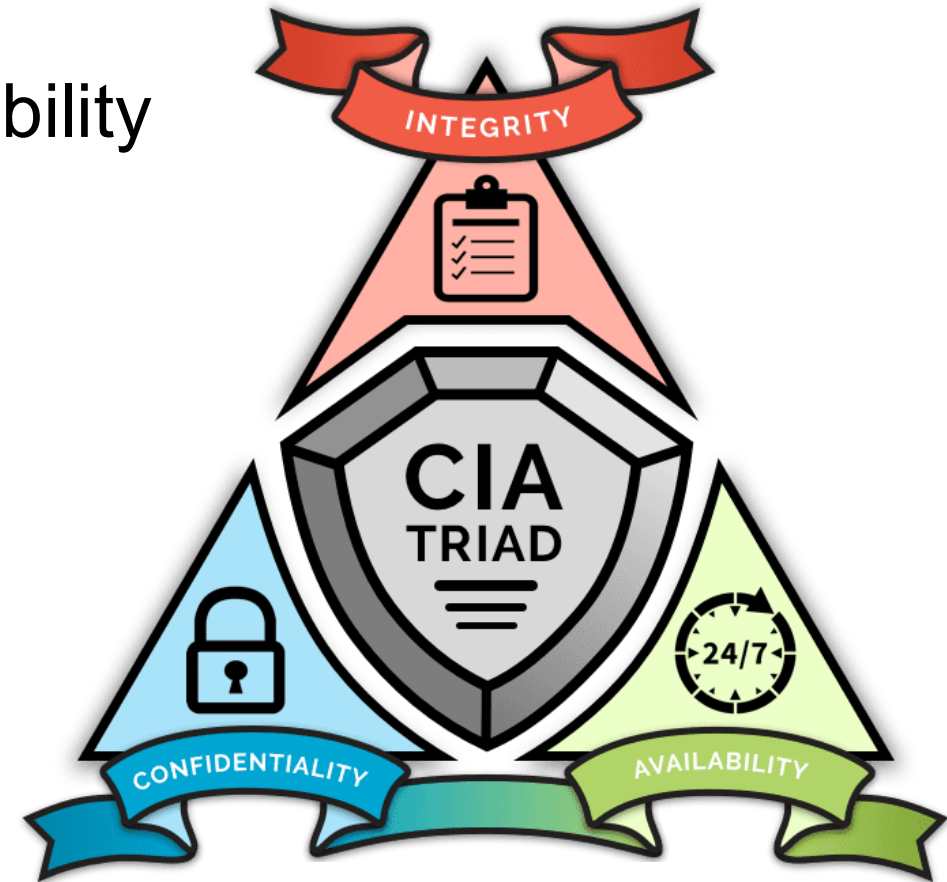
Q. Is Your Computer Secure?



Under what conditions can you say your computer is secure?

Secure Systems Satisfy the CIA Properties⁷

- Three most important **properties** of computer security
- **CIA**: Confidentiality, Integrity, and Availability



CIA



- Confidentiality
- Integrity
- Availability

CIA (1): Confidentiality (기밀성)

- Confidentiality: information is not made available to unauthorized parties
- Integrity
- Availability

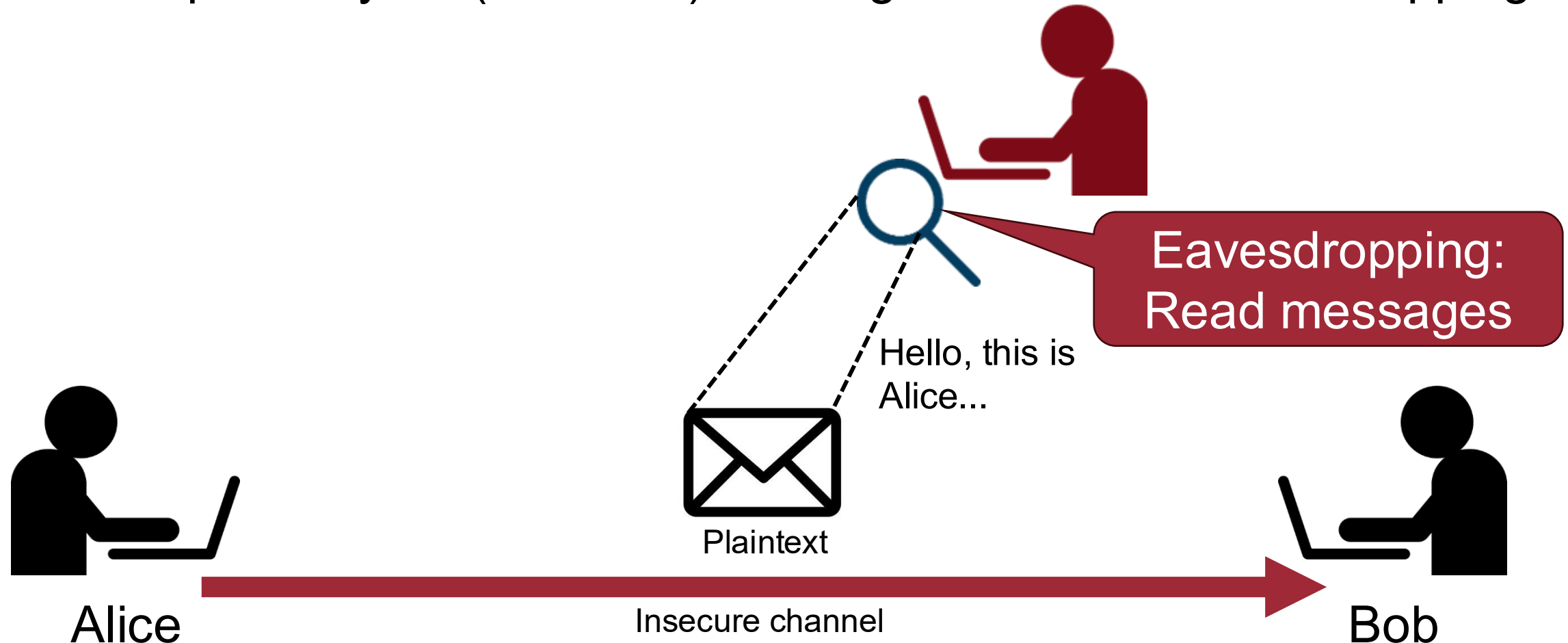
CIA (1): Confidentiality

- Information is not made available to unauthorized parties
- Avoidance of the unauthorized disclosure of information
 - Protection of data
 - Provide access for those who are allowed to see the data
 - Disallow others from learning anything about the data
- How to achieve confidentiality?
 - **Cryptography** (암호화, 암호학): transformation of information
 - **Access control** (접근제어): gatekeeper

CIA (1): Confidentiality – Cryptography

11

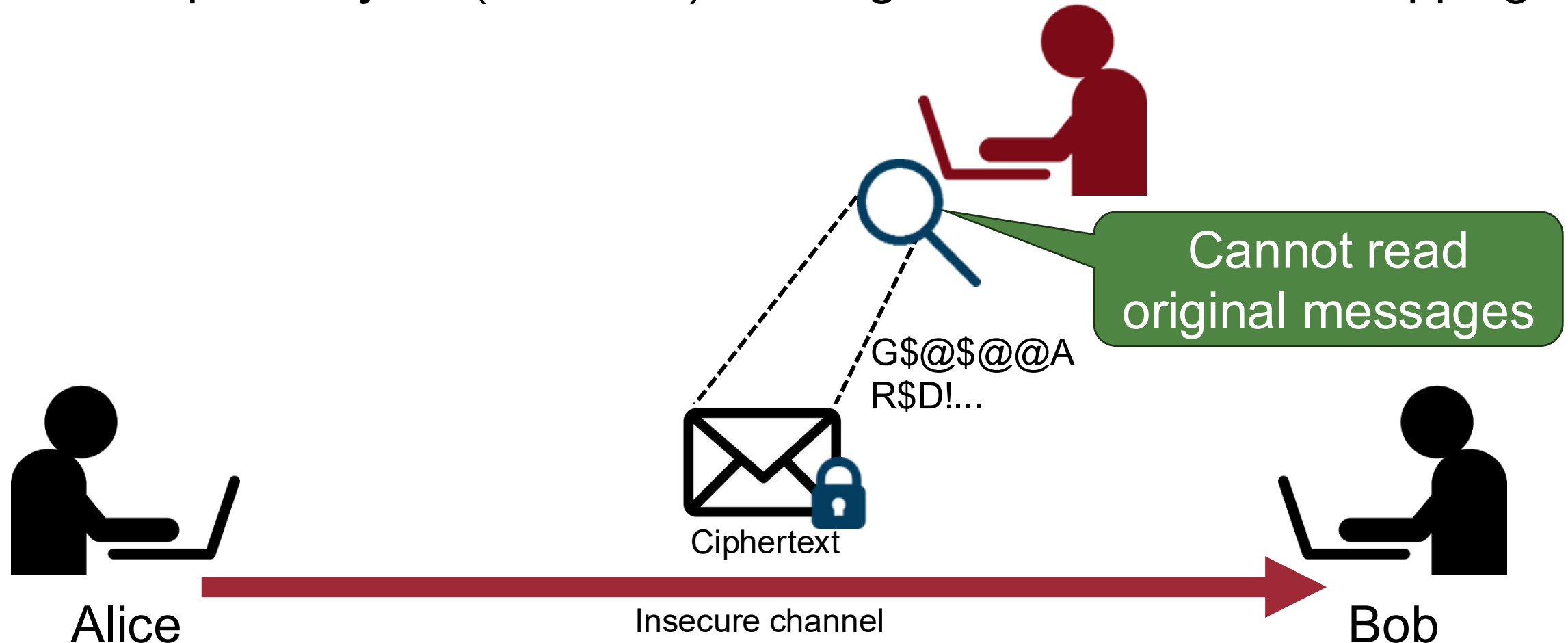
- “Secret writing” in Greek
- Goal: protect your (sensitive) messages/data from eavesdropping



CIA (1): Confidentiality – Cryptography

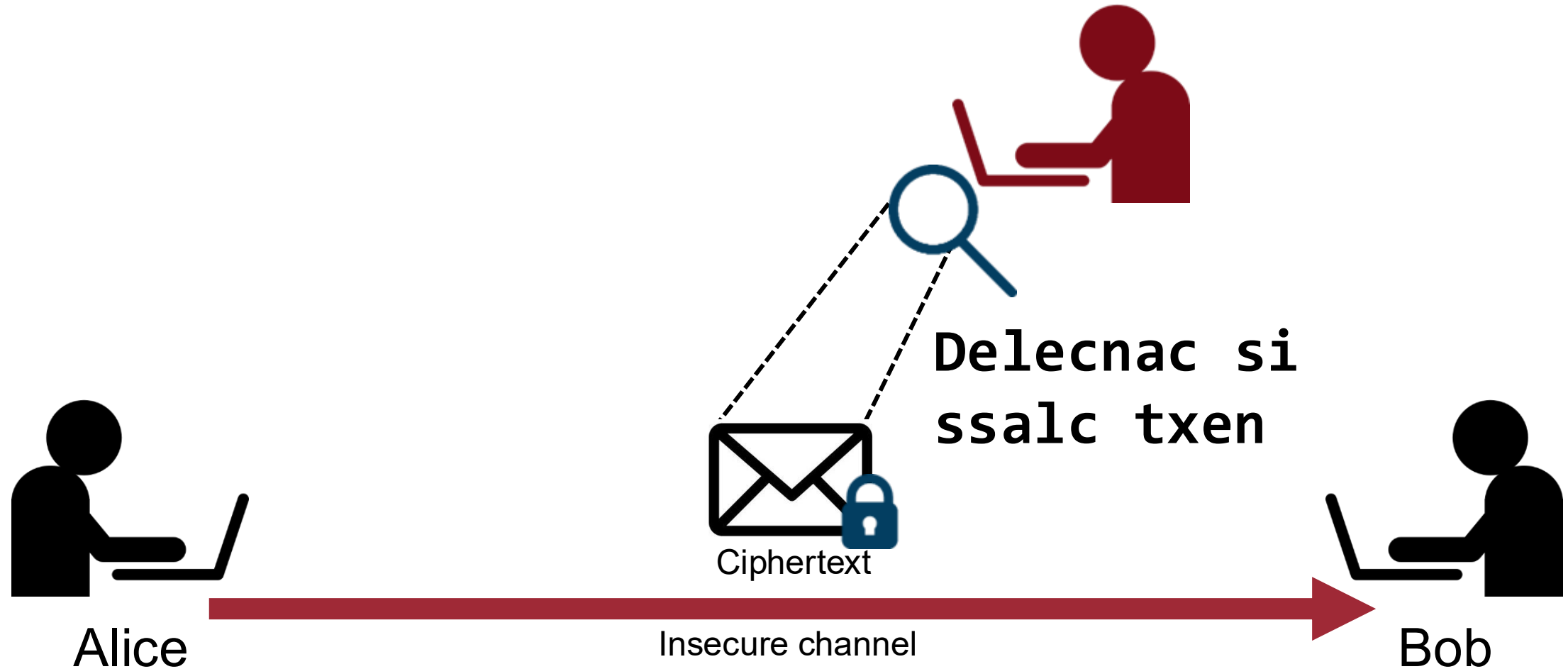
12

- “Secret writing” in Greek
- Goal: protect your (sensitive) messages/data from eavesdropping



Quiz: Can you Decrypt this One?

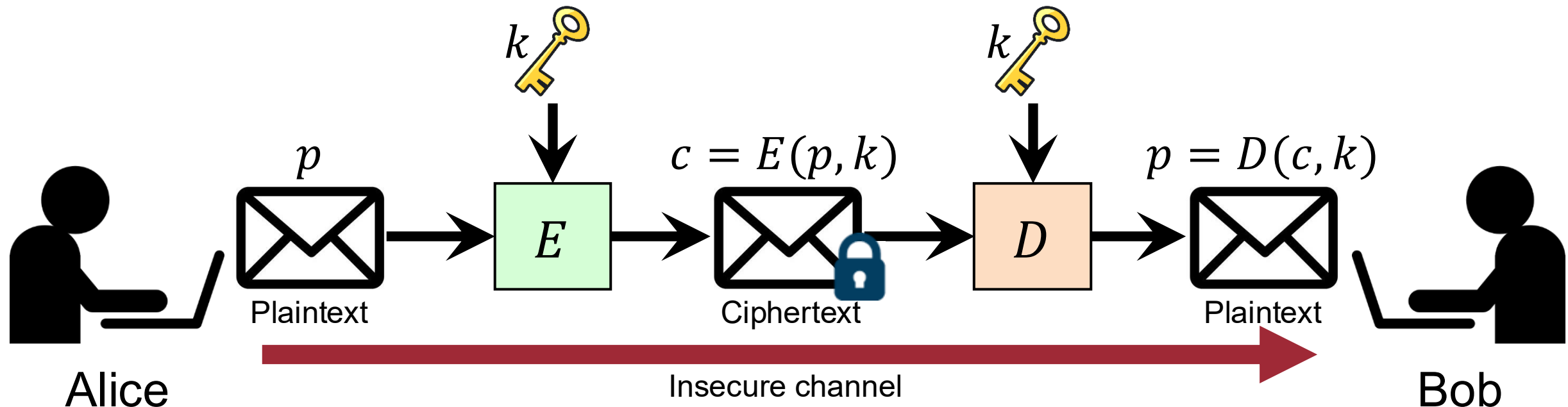
13



CIA (1): Confidentiality – Cryptography

15

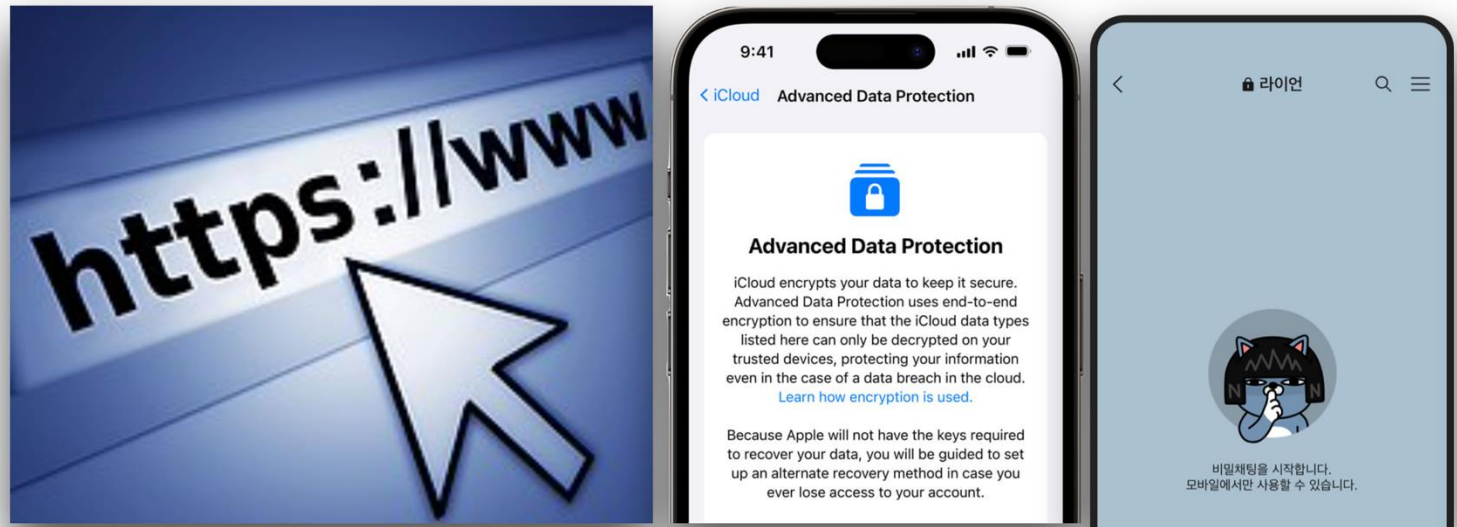
- To be secure: make it **extremely difficult** to decrypt the data without the decryption key
- Two functions: encryption (E) and decryption (D) parameterized by a plaintext (p), ciphertext (c), and cryptographic key (k)



CIA (1): Confidentiality – Cryptography

16

- Transformation of information using an **encryption key**
- Only be read by another user who has the **decryption key**
- Schemes: symmetric-key encryption, public-key encryption, etc
- Example:



CIA (1): Confidentiality – Access Control

17

- **Rules** and **policies** that limit access to confidential information
- Determine what users have permission to do
- Permission is determined by identity (e.g., name, serial) or role (e.g., professor, TA, student)
- Example: Linux file system

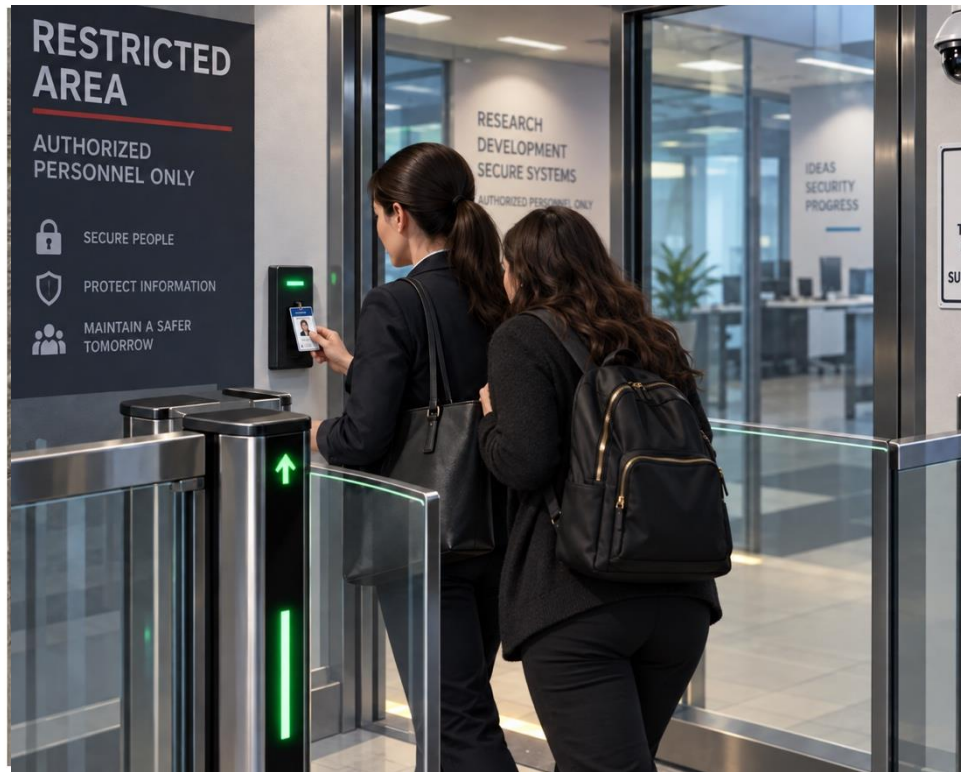
	/etc/passwd	/usr/bin	/home/prof/exam_problem/
root	rw	rwX	rwX
professor	r	rX	rwX
ta	r	rX	r
student1	r	rX	-
student2	r	rX	-

Students 1 and 2 are
unable to read the
exam problem!

CIA (1): Confidentiality – Access Control

18

Access Control Failure



[헤럴드경제=박로명 기자] “입주민이 카드를 찍으면 뒤에 외부인이 붙어서 커뮤니티 시설에 입장하는 장면을 자주 목격한다. 가구 수가 많은 대단지다보니 외부인이 입주민 카드를 남용해 커뮤니티 시설을 사용하는 일이 빈번한데, 커뮤니티 시설에선 주민만 만나고 싶다”(강남구 개포동 아파트 주민 A씨)

An unauthorized person enters a restricted facility by tailgating an authorized person

Exercise: Internet Banking

- What mechanism is used to achieve confidentiality?
 - Visit the bank website and login
 - ID and PW are sent to the server by your web browser using HTTPS
 - The server allows you to access only your account

CIA (2): Integrity (무결성)

- Confidentiality: information is not made available to unauthorized parties
- Integrity
- Availability

CIA (2): Integrity (무결성)

- Confidentiality: information is not made available to unauthorized parties
- Integrity: information is not modified in an unauthorized manner
- Availability

CIA (2): Integrity



Information has not been altered in an unauthorized way

- **Benign compromise:** information altered by accident
 - E.g., bit flips in memory due to cosmic ray

CIA (2): Integrity – Benign Compromise

23



CIA (2): Integrity

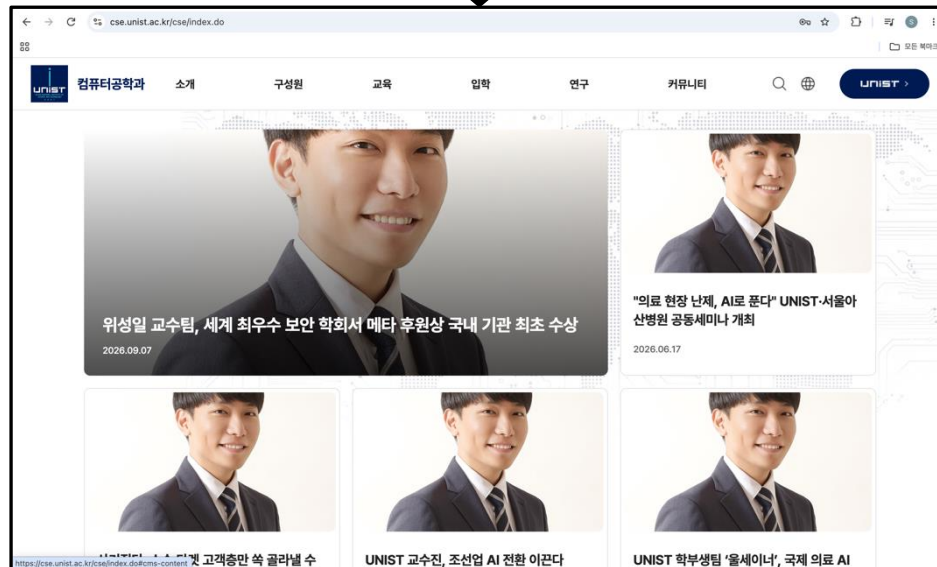
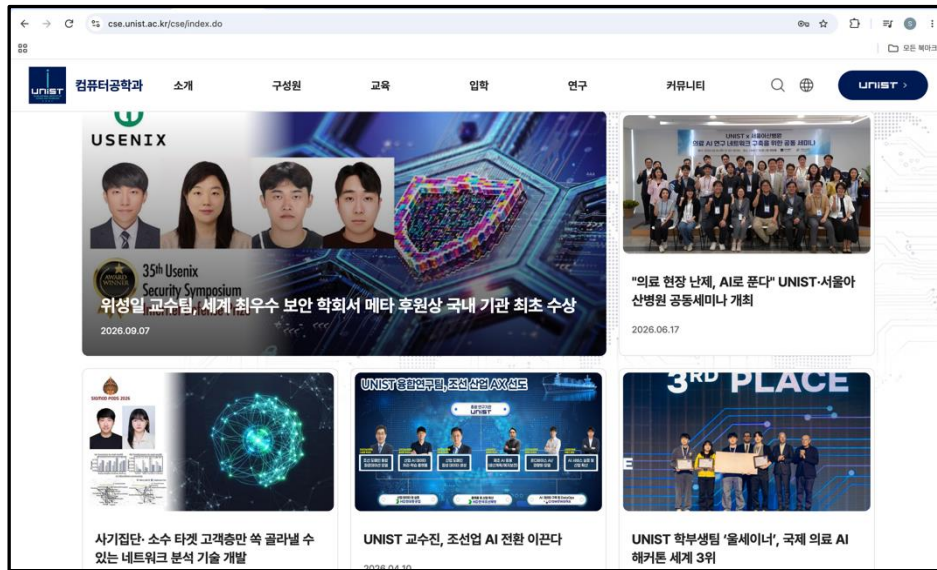


Information has not been altered in an unauthorized way

- **Benign compromise:** information altered by accident
 - E.g., bit flips in memory due to cosmic ray
- **Malicious compromise:** information altered by attackers
 - E.g., malicious code that changes some files in a system

CIA (2): Integrity – Malicious Compromise

25



수면위로 떠오른 UNIST(유니스트) '해킹 사건'

입력 2017-04-29 09:28

'룸메이트 바꾸려고' 전산망 뚫어...2013년부터 총 4건
유니스트 보안 정보 둔감 '지적'

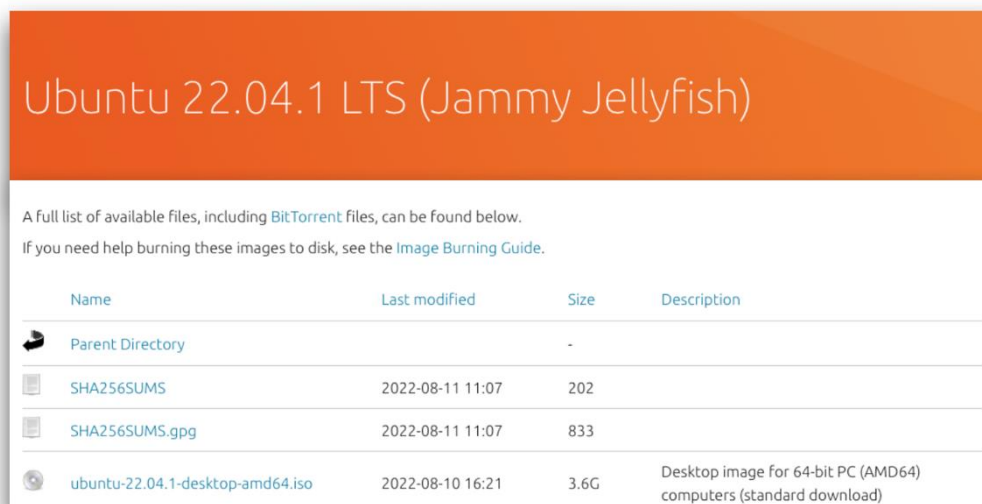


지난 2014년 3월 유니스트 해킹동아리가 학교전산망에 침입, 자신들의 해커 모집광고를 종합전산망 홈페이지에 게재해 놓은 배너 모습.



Ensuring Integrity

- How to ensure the integrity of computer systems?
- **Backups**: periodic archiving of data
- **Checksums**: computation of a function that maps the data to a numerical value



Ubuntu 22.04.1 LTS (Jammy Jellyfish)

A full list of available files, including [BitTorrent](#) files, can be found below.
If you need help burning these images to disk, see the [Image Burning Guide](#).

Name	Last modified	Size	Description
 Parent Directory		-	
 SHA256SUMS	2022-08-11 11:07	202	
 SHA256SUMS.gpg	2022-08-11 11:07	833	
 ubuntu-22.04.1-desktop-amd64.iso	2022-08-10 16:21	3.6G	Desktop image for 64-bit PC (AMD64) computers (standard download)

CIA (3): Availability (가용성)

- Confidentiality: information is not made available to unauthorized parties
- Integrity: information is not modified in an unauthorized manner
- Availability

CIA (3): Availability (가용성)

- Confidentiality: information is not made available to unauthorized parties
- Integrity: information is not modified in an unauthorized manner
- Availability: information should be readily available when it is needed

CIA (3): Availability



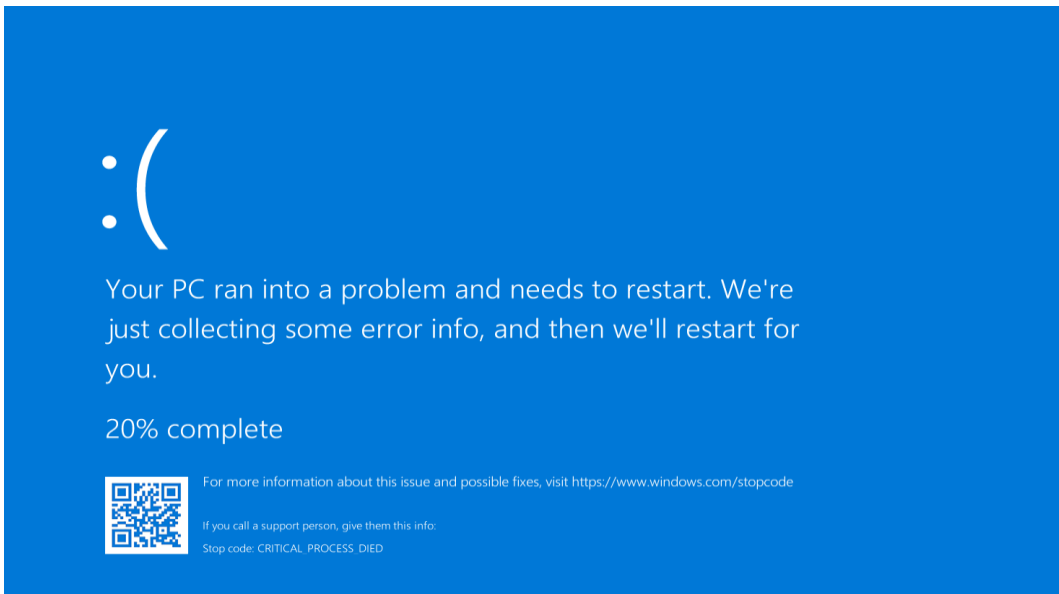
- Information should be ***accessible*** and ***modifiable*** in a timely fashion
- Imagine a unbreakable and unopenable vault. Is it useful?



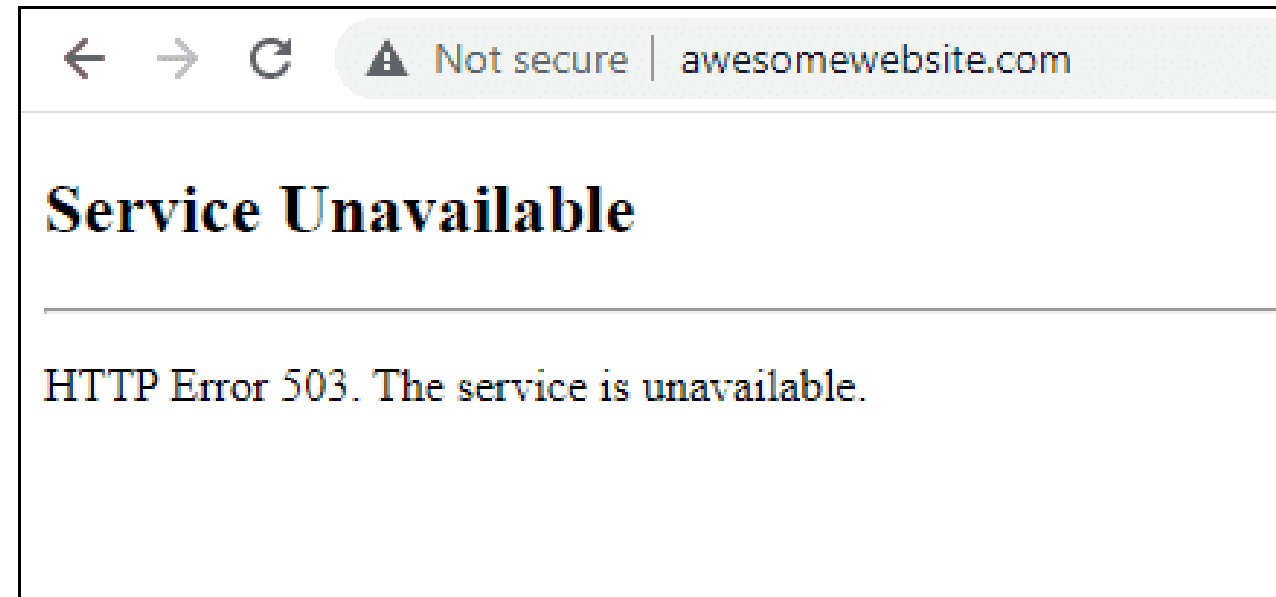
CIA (3): Availability



- Information should be **accessible** and **modifiable** in a timely fashion
- Imagine a unbreakable and unopenable vault. Is it useful?



Blue Screen of Death



503 Error

CIA (3): Availability



- Information should be ***accessible*** and ***modifiable*** in a timely fashion
- Imagine a unbreakable and unopenable vault. Is it useful?

Kakao's meltdown raises big questions about its management



“President office said
KaKao’s network disturbance could even
be **a threat to national security**”

CIA (3): Availability



- Information should be **accessible** and **modifiable** in a timely fashion
- Imagine a unbreakable and unopenable vault. Is it useful?
- How to achieve availability?
 - **Physical protections**: keep information available even in physical challenges (e.g., storms, earthquakes, or power outages)
 - **Computational redundancies**: computers that serve as fallbacks in the case of failure

Other properties?



- **Confidentiality**
- **Integrity**
- **Availability**

Other properties?



- Confidentiality
- Integrity
- Availability

+ **Authentication:** the ability of a computer system to *confirm the sender's identity*

Authentication (인증)

- Determination of the **identity** or **role**
- Typical method
 - Something you are (Fingerprint, iris pattern, ...)
 - Something you know (Password, PIN, ...)
 - Something you have (Smart card, key, ...)



UNIST | 로그인

계정생성 아이디찾기 비밀번호 초기화

ID

PW

로그인



UNIST

joon@unist.ac.kr

ID 확인

+XX XXXXXXXXXX79에 문자 메시지

+XX XXXXXXXXXX79에 전화

추가 정보

확인 방법이 최신 상태입니까? <https://aka.ms/mfasetup>에서 확인하세요.

취소

Other properties?



- **Confidentiality**
 - **Integrity**
 - **Availability**
-
- + **Authentication:** the ability of a computer system to *confirm the sender's identity*
 - + **Non-repudiation:** the ability of a computer system to *confirm that the sender can not deny about something sent*

Non-repudiation (부인방지)

- What happens if the suspected attacker says that...
 - “I did not make commitment. Maybe someone pretended to be me!
(오리발 내밀기, repudiation)”

✦ AI 개요

'오리발을 내민다'는 잘못을 저지르고도 엉뚱한 변명을 하며 모른 척 딱 잡아떼는 태도를 뜻하는 속담입니다. y 영남일보 +1

- How to determine that statements, policies, and permissions are genuine?
- **Non-repudiation** by secure authentication: authentic statement cannot be denied
 - E.g., digital signature



Web Programming Basic

Web Infrastructure

39



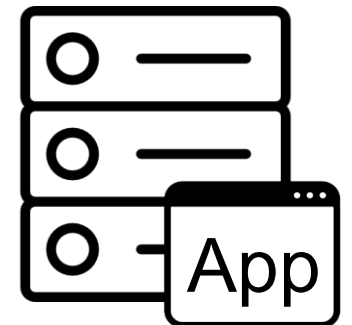
Hypertext Markup Language
(HTML)



HTTP Request



HTTP Response websec-lab.com
web server



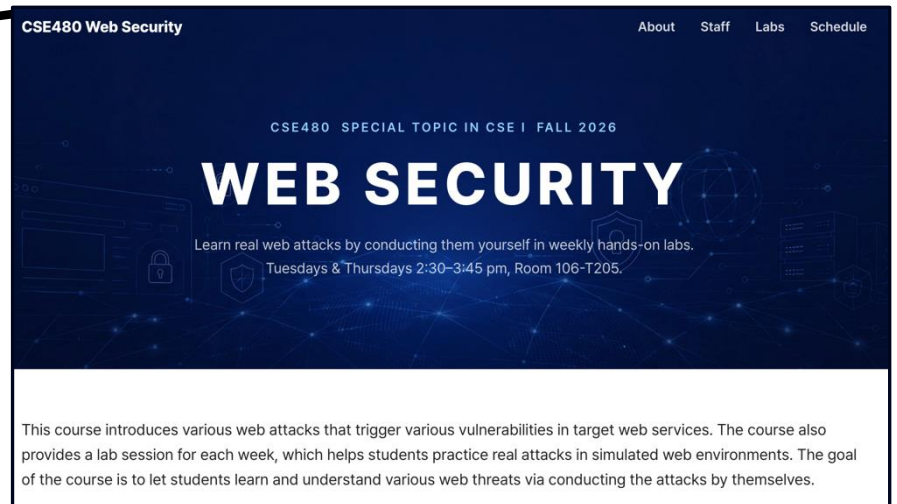
Hypertext Markup Language (HTML)

40



- Markup language for web page layout
 - NOT programming language (i.e., for computation)!
- A web page (document) is written in HTML using markup tags
 - E.g., `<p>`, `<img>`

```
<html>
  <body>
    ...
  </body>
</html>
```



Hypertext Markup Language (HTML)

41



- Markup language for web page layout
 - NOT programming language (i.e., for computation)!
- A web page (document) is written in HTML using markup tags
 - E.g., `<p>`, `<img>`
- A **browser** interprets a web page when rendering the page
- Describes a hyper-text document
 - E.g., image, audio, video

What if we need
computation?
⇒ JavaScript!

```
<html>
  <body>
    ...
  </body>
</html>
```



Uniform Resource Locators (URLs)

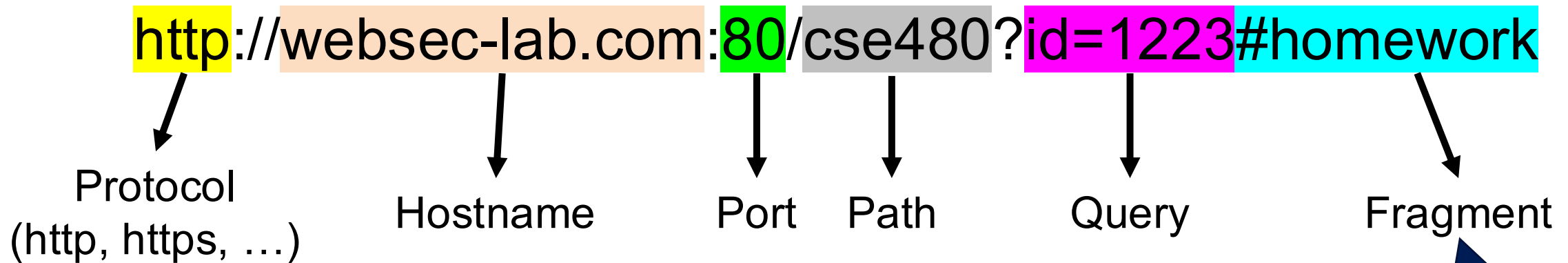
- Global identifiers of network-retrievable documents
- Example

`http://websec-lab.com:80/cse480?id=1223#homework`



Uniform Resource Locators (URLs)

- Global identifiers of network-retrievable documents
- Example

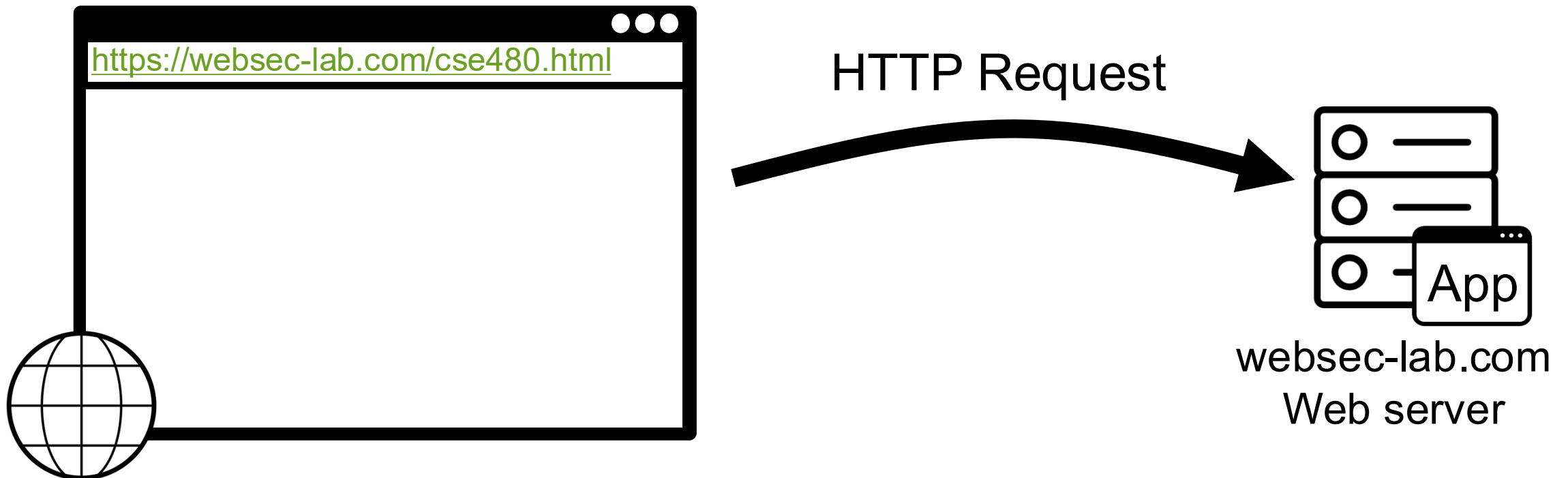


- Special characters are encoded as hex:
 - New line → %0A
 - Space → %20
 - + → %2B

Fragments are not sent to the server

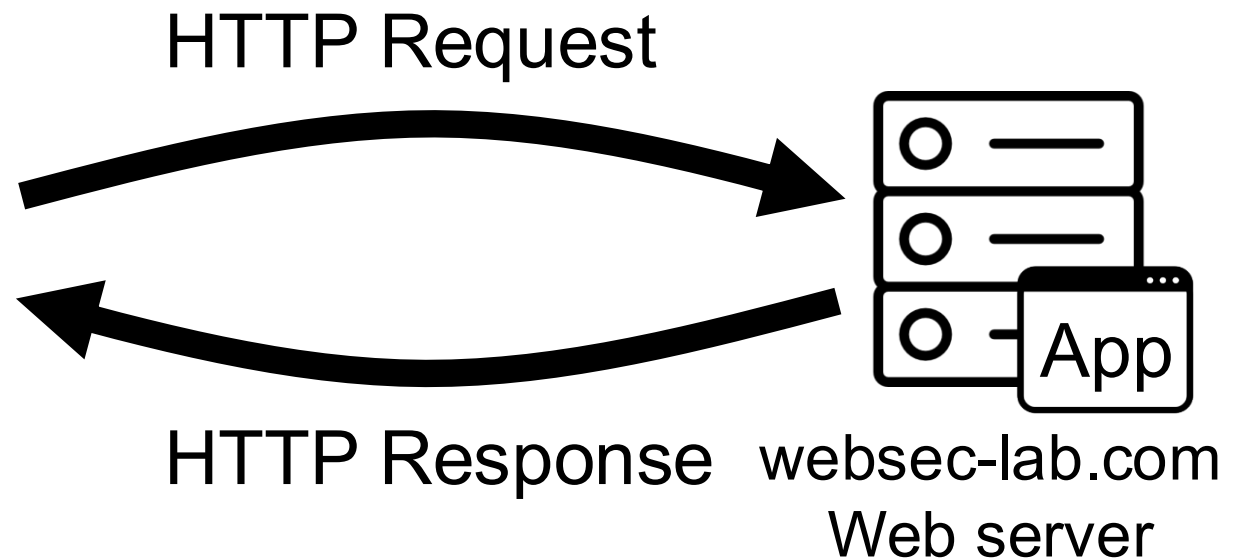
Hyper Text Transfer Protocol (HTTP)

- The primary protocol for data transfer between web browsers and servers



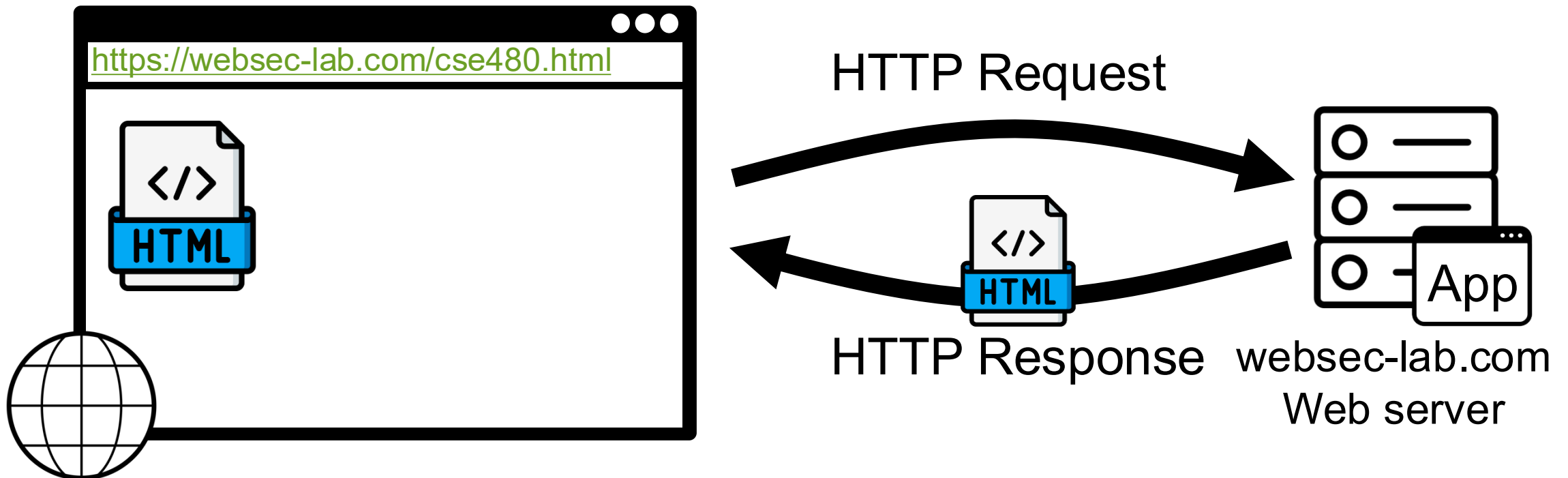
Hyper Text Transfer Protocol (HTTP)

- The primary protocol for data transfer between web browsers and servers



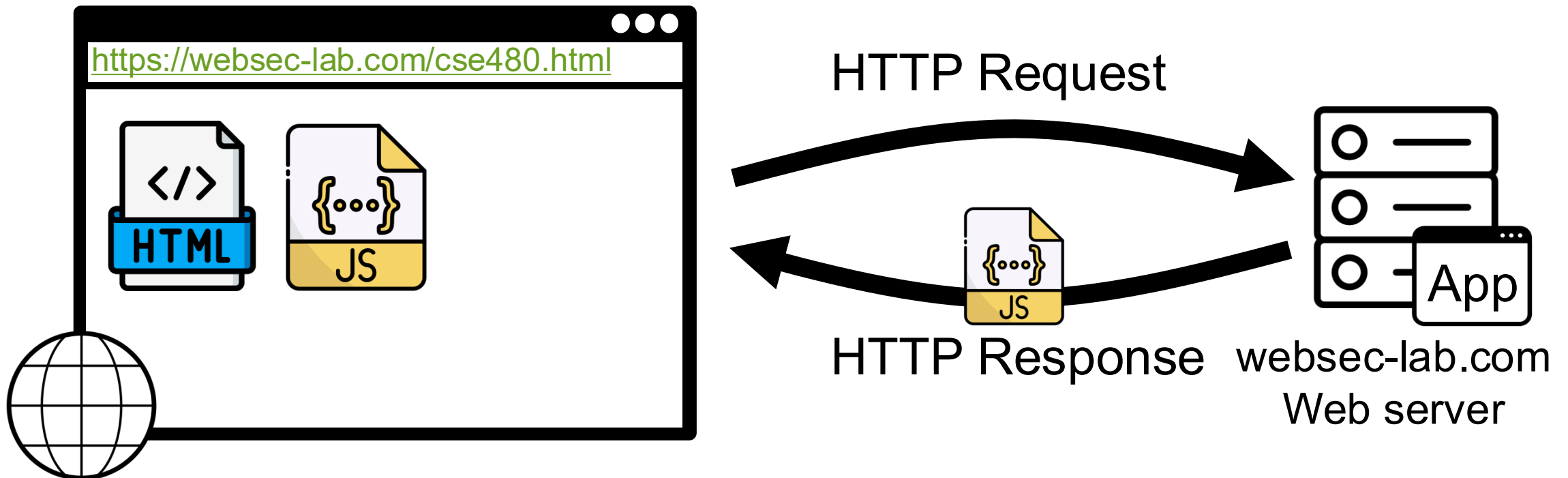
Hyper Text Transfer Protocol (HTTP)

- The primary protocol for data transfer between web browsers and servers



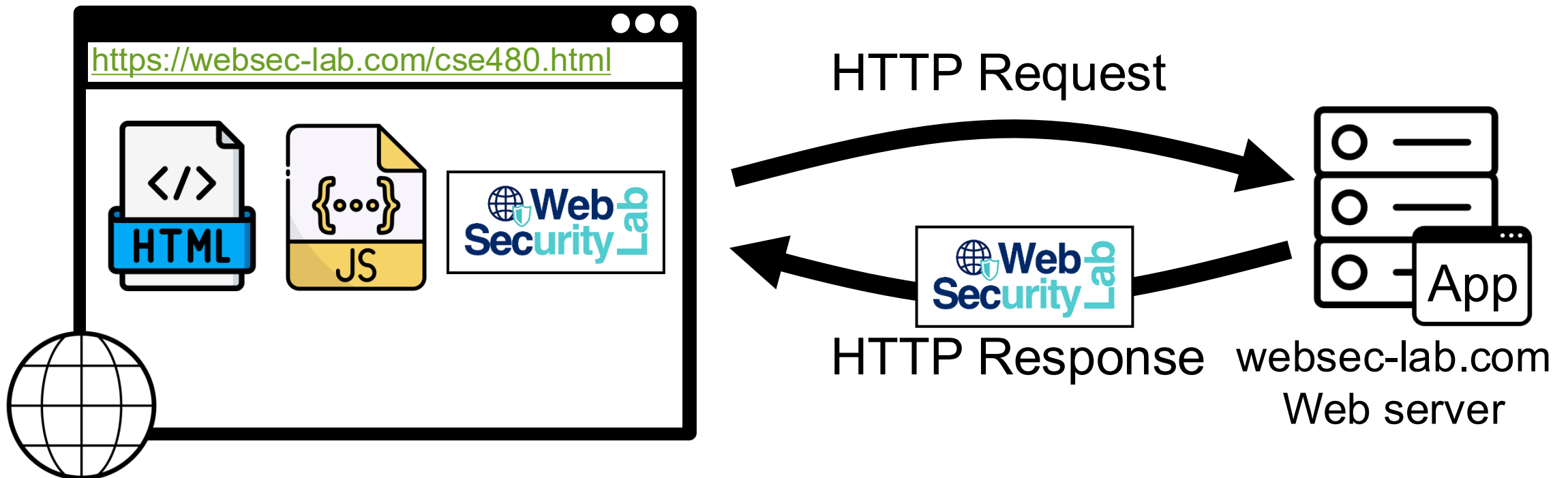
Hyper Text Transfer Protocol (HTTP)

- The primary protocol for data transfer between web browsers and servers



Hyper Text Transfer Protocol (HTTP)

- The primary protocol for data transfer between web browsers and servers



Hyper Text Transfer Protocol (HTTP)



- The primary protocol for data transfer between web browsers and servers
- Application layer protocol
 - A request is sent over a TCP connection on port:80
- Stateless request/response protocol
 - Each request is independent to previous requests

HTTP Request



```
GET /cse467.html HTTP/1.1
Host: websec-lab.com
Accept-Language: en
Connection: keep-alive
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;)
Referer: http://google.com
```

HTTP Request



Method

File path

Protocol

Request
Line {

GET /cse467.html HTTP/1.1

Host: websec-lab.com

Accept-Language: en

Connection: keep-alive

User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64;)

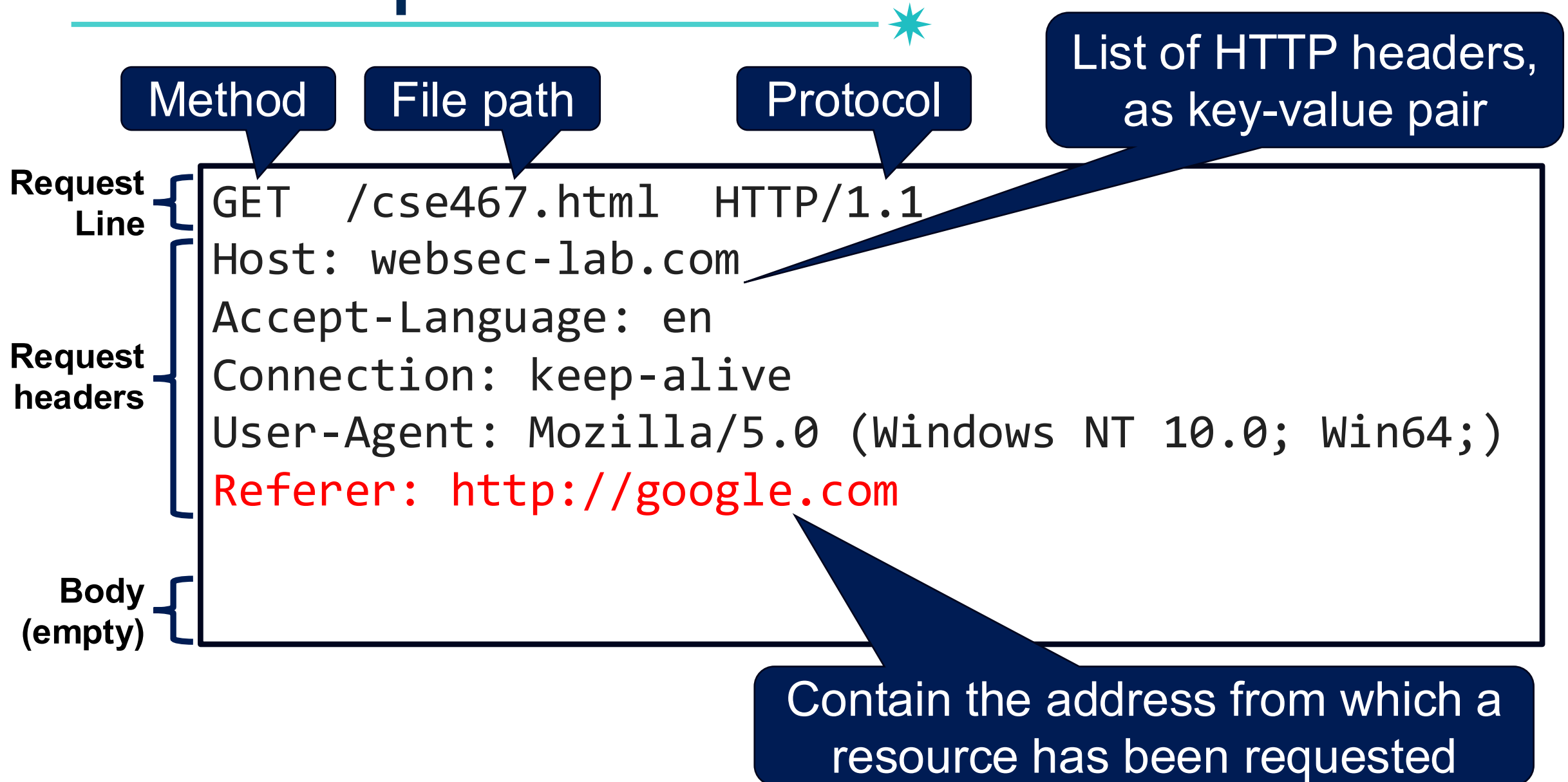
Referer: <http://google.com>

Many HTTP Methods



- **GET:** Get the resource at the specified URL
- **POST:** Create new resource at URL with payload
- **PUT:** Replace current representation of the target resource with request payload
- **PATCH:** Update part of the resource
- **DELETE:** Delete the specified URL

HTTP Request



Referrer Header

54



The screenshot shows a Google search interface. The address bar contains <http://google.com>. The search bar has the text 'CSE480'. Below the search bar, there are tabs for '전체' (All), '이미지' (Images), '지도' (Maps), '뉴스' (News), and '쇼핑' (Shopping). The search results show '검색결과 약 5개 (0.28초)' (About 5 search results in 0.28 seconds). The first result is from GitHub, titled 'CSE480: Web Security', with a URL of <https://websec-lab.github.io> and a snippet mentioning 'Course Information. Instructor: Seongil Wi; Time: Tuesd... T202; TA: Dongyeon Yu (유동연, dy3199@unist.ac.kr). Gr...

The screenshot shows the 'CSE480 Web Security' course page. The address bar contains <https://websec-lab.com/cse480.html>. The page has a dark blue header with the title 'CSE480 Web Security' and navigation links for 'About', 'Staff', 'Labs', and 'Schedule'. The main content area features the text 'CSE480 SPECIAL TOPIC IN CSE I FALL 2026' and a large heading 'WEB SECURITY'. Below this, it says 'Learn real web attacks by conducting them yourself in weekly hands-on labs. Tuesdays & Thursdays 2:30-3:45 pm, Room 106-T205.' The footer contains a paragraph: 'This course introduces various web attacks that trigger various vulnerabilities in target web services. The course also provides a lab session for each week, which helps students practice real attacks in simulated web environments. The goal of the course is to let students learn and understand various web threats via conducting the attacks by themselves.'

Referrer Header

55

The diagram illustrates the Referrer Header concept. It shows two browser windows. The left window displays a Google search for 'CSE480'. The right window displays the 'CSE480 Web Security' page on 'websec-lab.com'. A green arrow points from the search result on Google to the web page, indicating the source of the traffic. A hand icon points to the search result, and a blue callout box contains the text 'Referrer header: http://google.com'.

http://google.com

Google

CSE480

전체 이미지 지도 뉴스 쇼핑

검색결과 약 5개 (0.28초)

GitHub

https://websec-lab.github.io > courses > 2026f-cse480

CSE480: Web Security

Course Information. Instructor: Seongil W. T202; TA: Dongyeon Yu (dy3199@websec-lab.com)

https://websec-lab.com/cse480.html

CSE480 Web Security

About Staff Labs Schedule

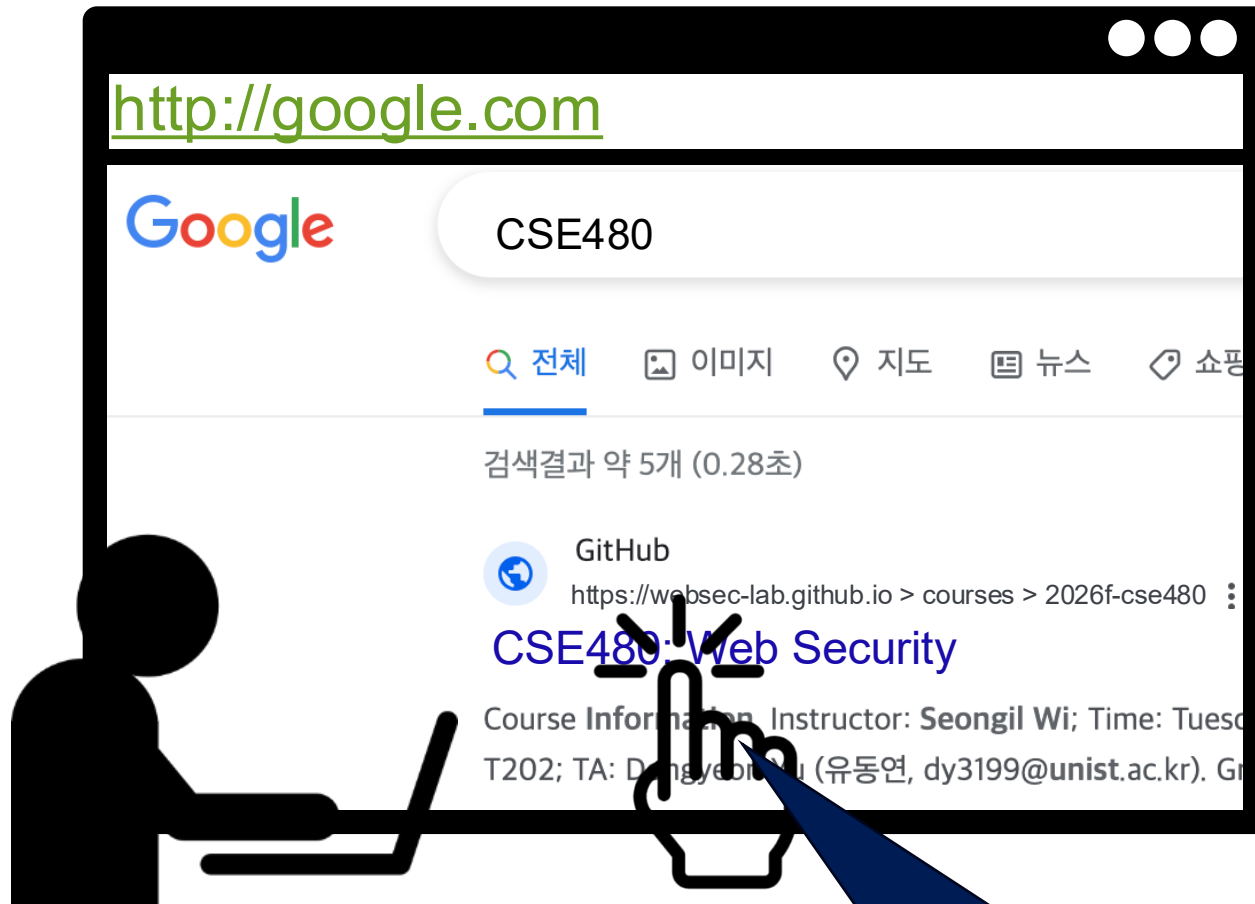
CSE480 SPECIAL TOPIC IN CSE I FALL 2026

WEB SECURITY

Learn real web attacks by conducting them yourself in weekly hands-on labs. Tuesdays & Thursdays 2:30-3:45 pm, Room 106-T205.

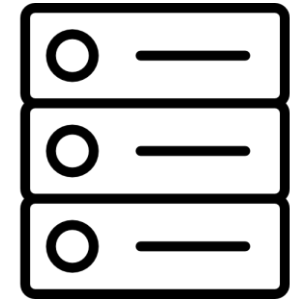
Referrer header: http://google.com

Referrer Header in Detail



(1) Click the link

(2) Send HTTP request
(with referrer header)



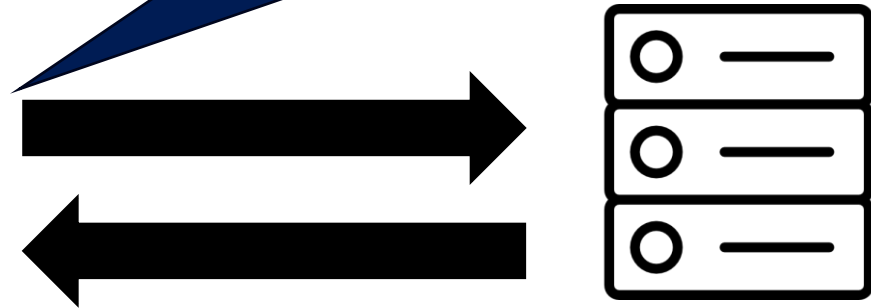
Web server
websec-lab.github.io

Referrer Header in Detail

57



(2) Send HTTP request
(with referrer header)



Web server
websec-lab.github.io

The server can analyze where
the request originated



Are there any security issues
with the referrer header?

HTTP Response



Status
Line

HTTP/1.1 200 OK

Response
headers

Date: Sat, 21 Oct 2023 07:58:24 GMT

Connection: Keep-alive

Content-Type: text/html

Content-Length: 2543

Response
body

<html>

<body>

some data...

</body>

</html>

HTTP Response

HTTP version

Status code

Status text

Status Line

HTTP/1.1 200 OK

Response headers

Date: Sat, 21 Oct 2023 07:58:24 G
Connection: Keep-alive
Content-Type: text/html
Content-Length: 2543

Response body

```
<html>
  <body>
    some data...
  </body>
</html>
```

HTTP STATUS CODES

2xx Success

200

Success / OK

3xx Redirection

301

Permanent Redirect

302

Temporary Redirect

304

Not Modified

4xx Client Error

401

Unauthorized Error

403

Forbidden

404

Not Found

405

Method Not Allowed

5xx Server Error

501

Not Implemented

502

Bad Gateway

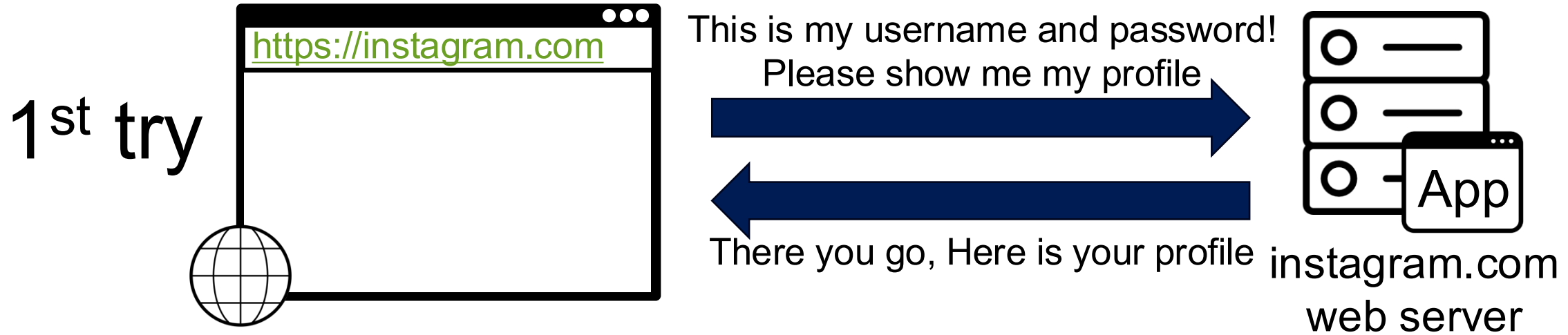
503

Service Unavailable

504

Gateway Timeout

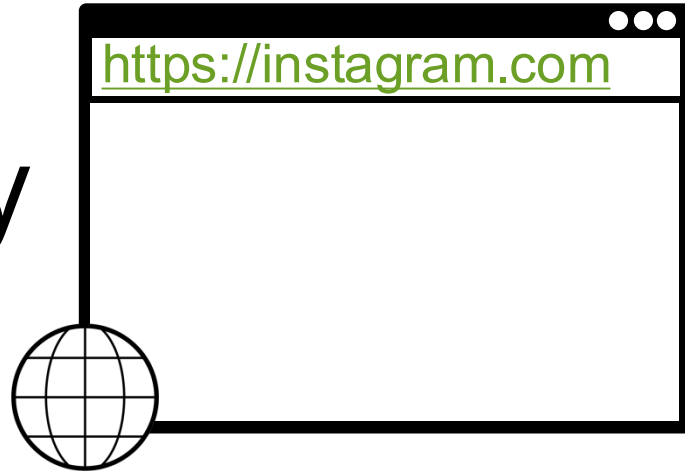
HTTP is a Stateless Protocol



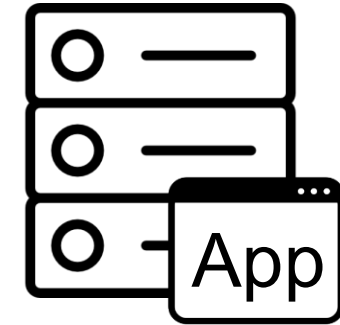
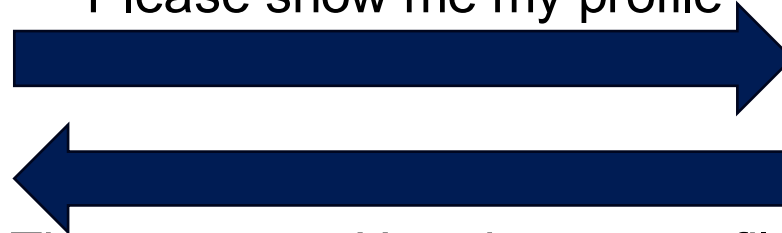
HTTP is a Stateless Protocol



1st try



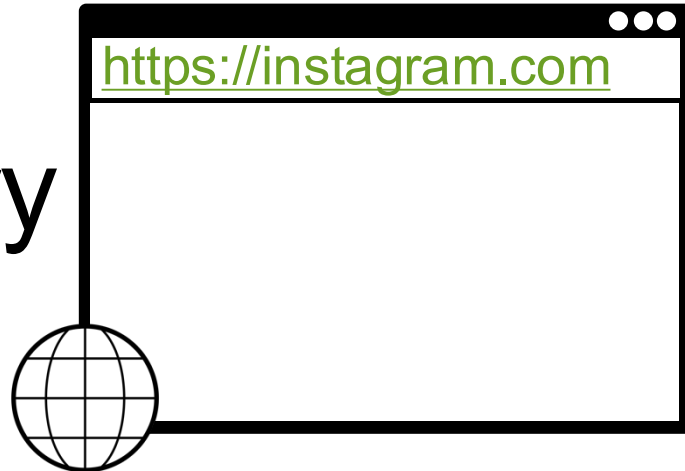
This is my username and password!
Please show me my profile



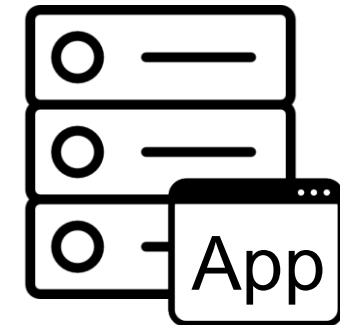
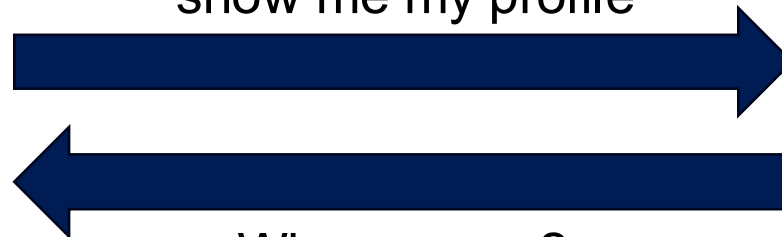
There you go, Here is your profile

instagram.com
web server

2nd try



Hey Instagram,
show me my profile



Who are you?
Please login and verify who you are...

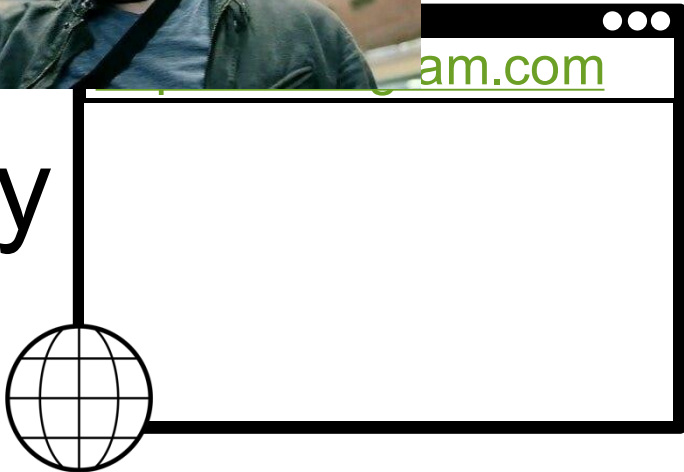
instagram.com
web server

HTTP is a Stateless Protocol

Stateless protocol:
Each request is independent
to previous request



2nd try

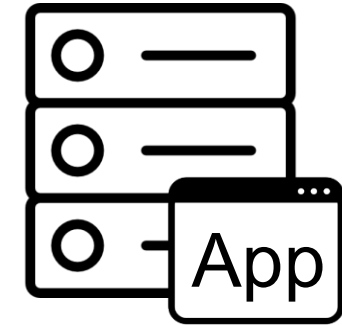


Here is my username and password!
Please show me my profile



There you go, Here is your profile

instagram.com
web server

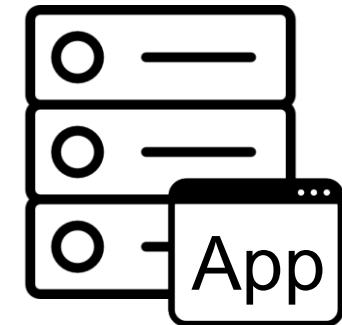


Hey Instagram,
show me my profile



Who are you?
Please login and verify who you are...

instagram.com
web server





How to make HTTP “act”
stateful?

Adding State to HTTP



- Recall: no inherent state in HTTP
 - Server does not keep any state after the connection is closed
- For static content sites, no problem
 - Developing “applications” is impossible though
 - E.g., shopping cart on Amazon
- Need to introduce state in HTTP
 - in the form of “cookies”

Cookie: Making HTTP Stateful



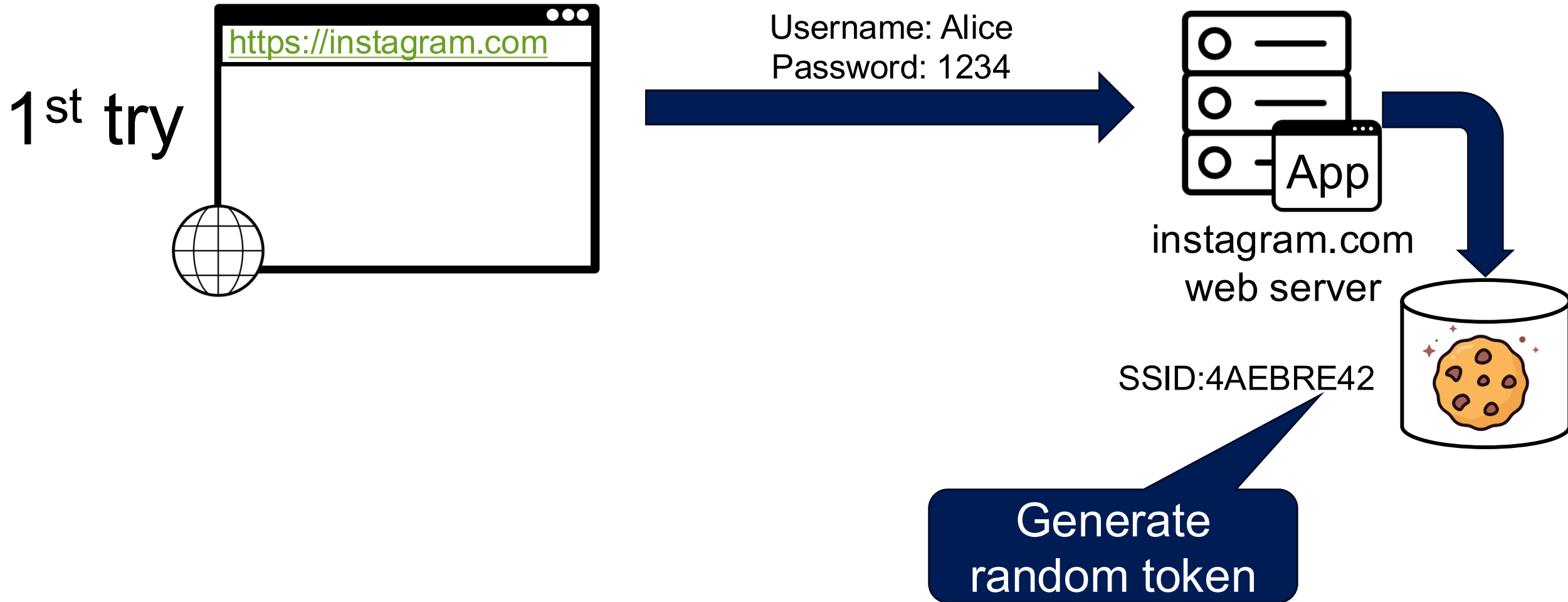
69

- HTTP cookie: small piece of data that a server sends to the browser, who stores it and sends it back with subsequent requests

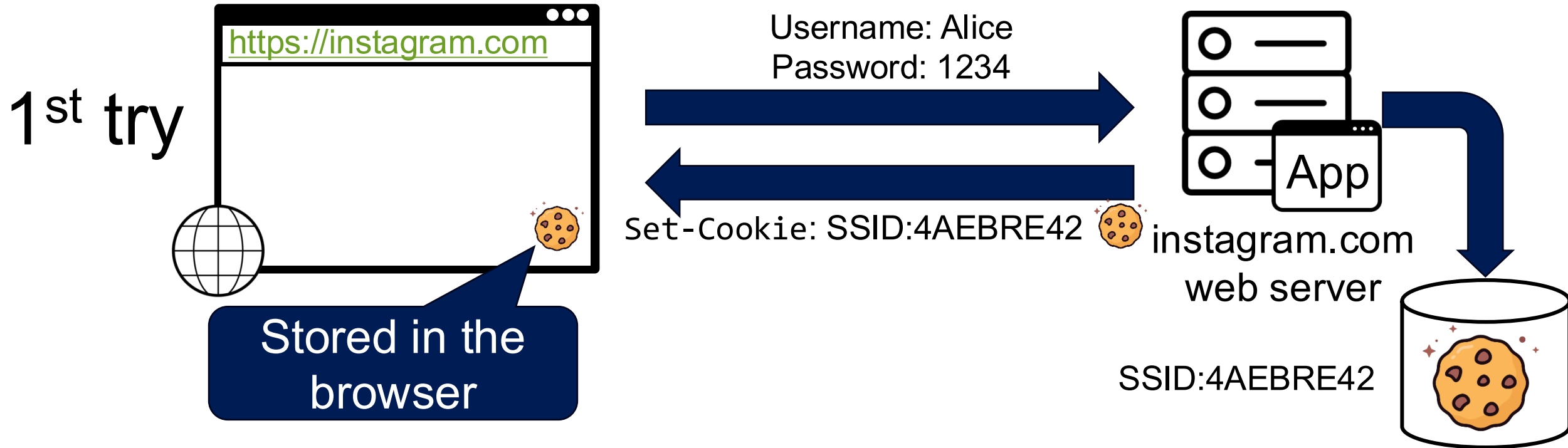


Cookie: Making HTTP Stateful

70



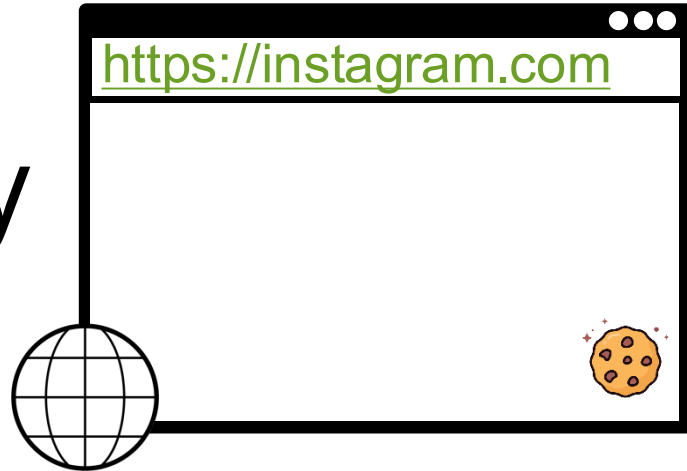
Cookie: Making HTTP Stateful



Cookie: Making HTTP Stateful

72

1st try



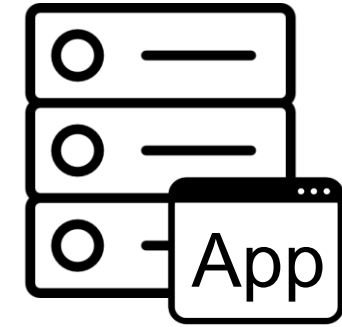
Username: Alice
Password: 1234



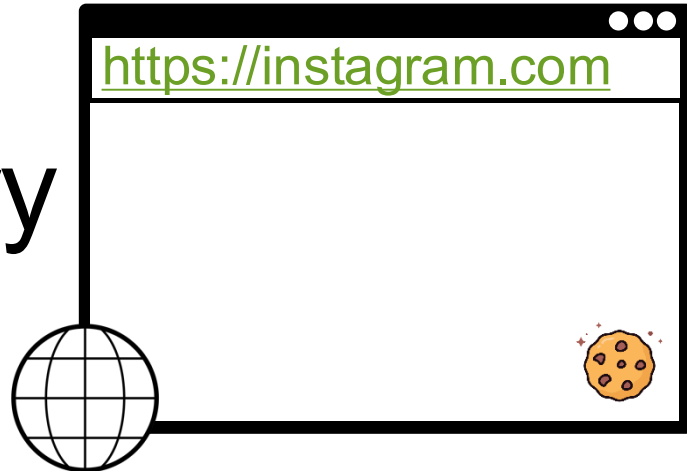
Set-Cookie: SSID:4AEBRE42



instagram.com
web server



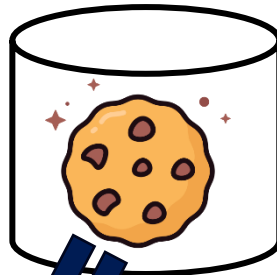
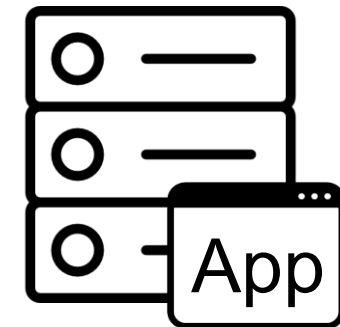
2nd try



Hey Instagram,
show me my profile
Cookie: SSID:4AEBRE42



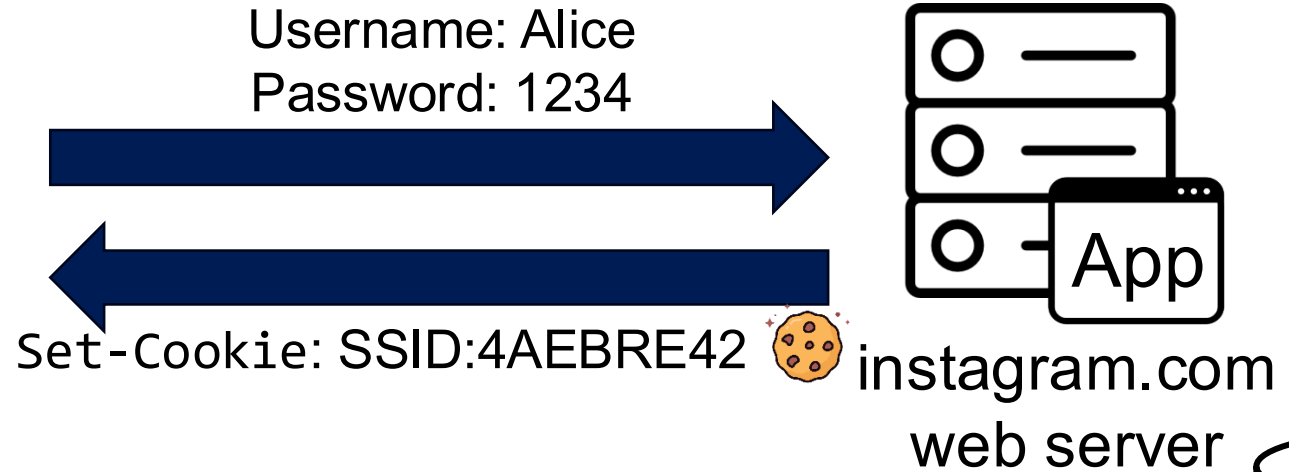
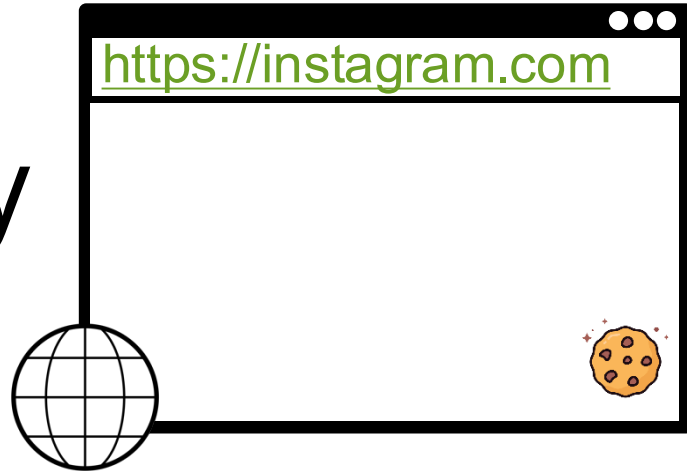
instagram.com
web server



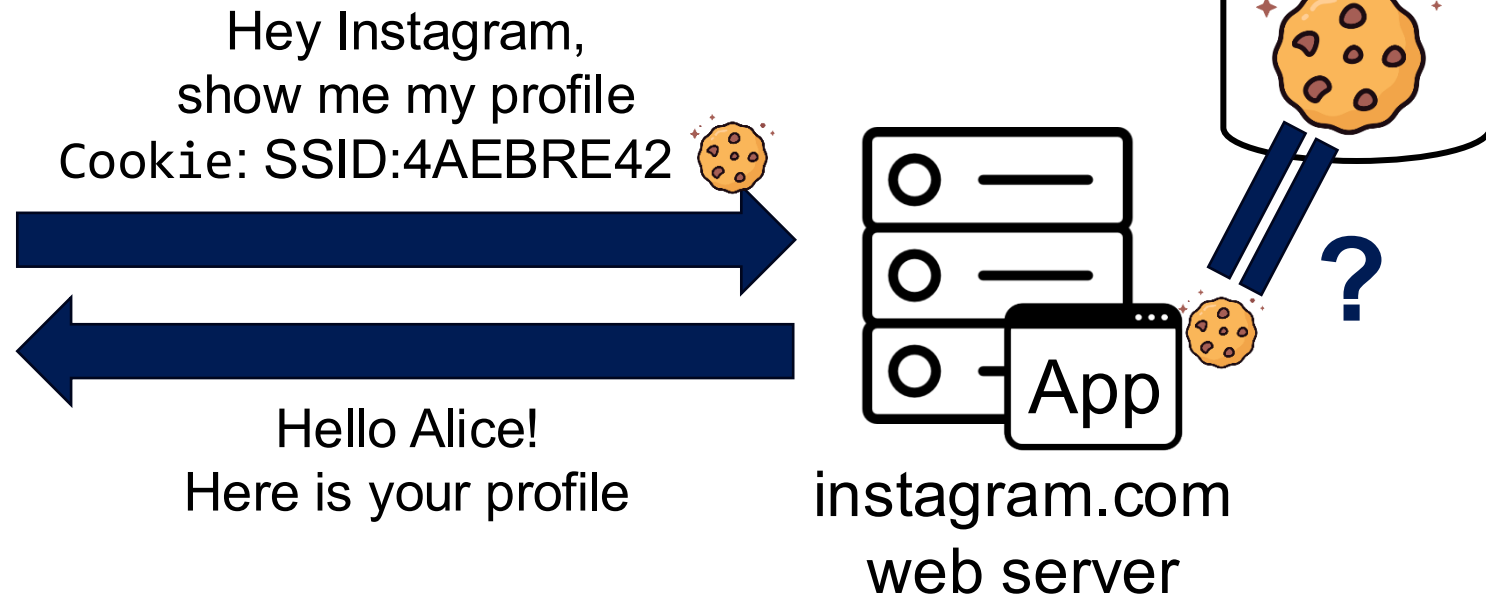
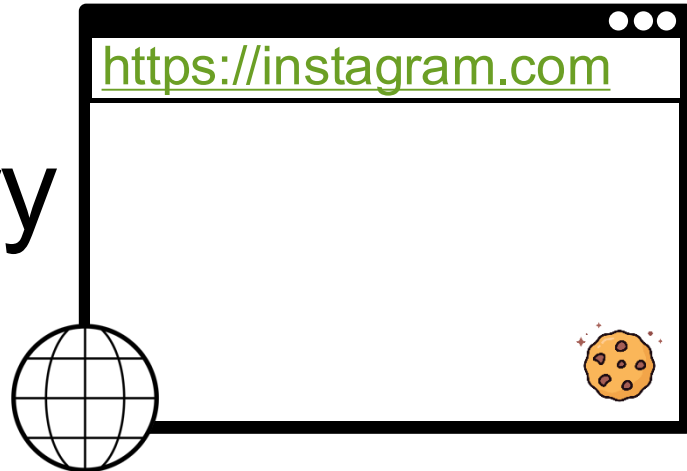
Cookie: Making HTTP Stateful

73

1st try



2nd try



Cookie: Making HTTP Stateful



74

- Generate random token on first page visit
- Sent to client via Set-Cookie header
- Client always sends along cookies in every request to the server
- Cookies are persisted in the browser
 - Controllable by Expires option in cookie

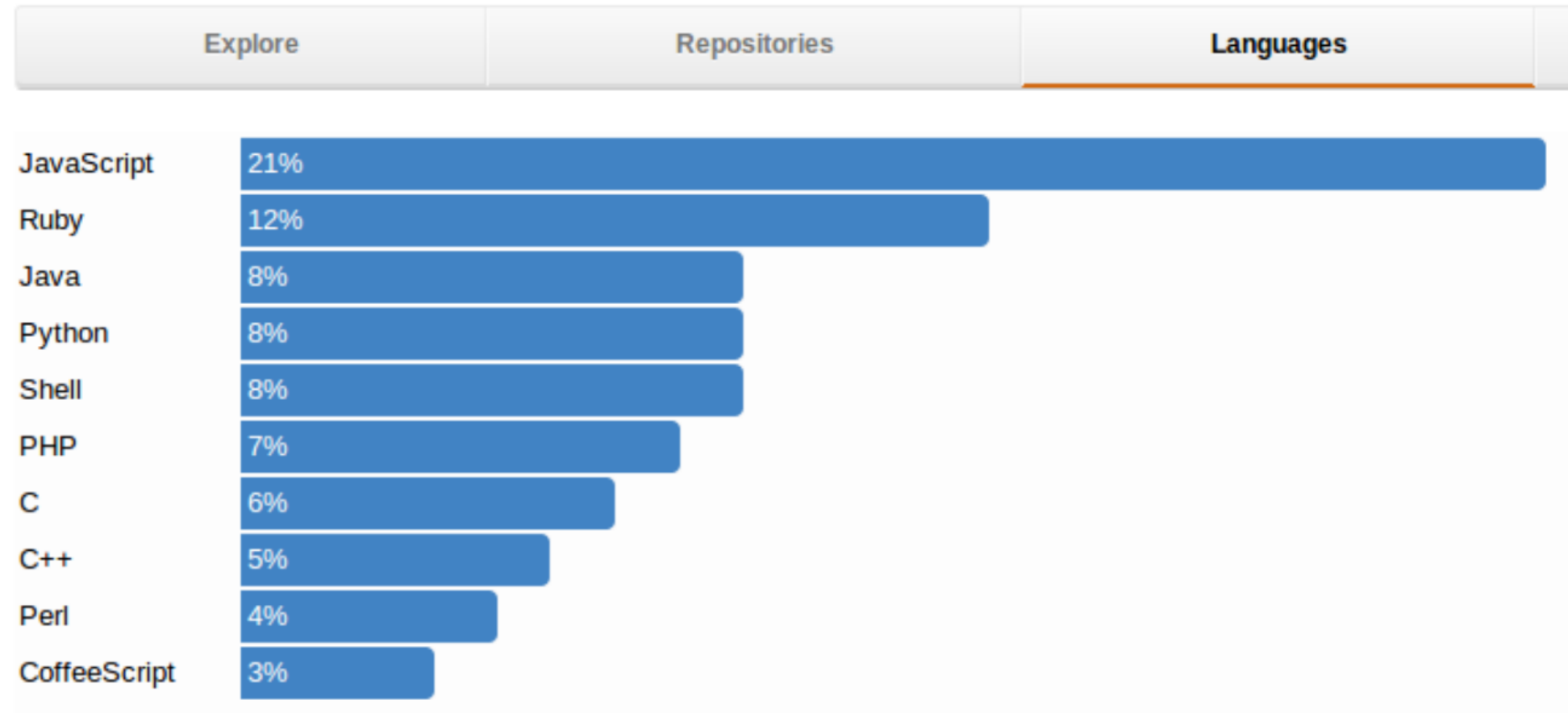


JavaScript (JS)



- Most popular language in the world!

Top Languages



JavaScript (JS)



- Developed by Brendan Eich at Netscape
- Later standardized for browser compatibility
 - ECMAScript Edition 3 (a.k.a., JavaScript 1.5)



Netscape



- HTML may contain JS program code to make web pages more dynamic

JS Example (1)



```
<html>
  <p id="demo"></p>
  <script>
    document.getElementById("demo").
      innerHTML = 5 + 6;

  </script>
</html>
```

JS Example (1)



Inline script with script tag

```
<html>  
  <p id="demo"></p>  
  <script>  
    document.getElementById("demo").  
      innerHTML = 5 + 6;  
  </script>  
</html>
```



JS Example (2)



```
<html>  
  <button type="button" onclick="document.write(5 + 6)">  
    Try it  
  </button>  
</html>
```

JS Example (2)

When the button is clicked,
overwrite whole document with 11

```
<html>  
  <button type="button" onclick="document.write(5 + 6)">  
    Try it  
  </button>  
</html>
```

Inline script with
onclick event handler

JS Example (3)



index.html

```
<html>  
  <script src="write.js">  
  </script>  
</html>
```

write.js

```
document.write(5 + 6)
```

JS Example (3)



Overwrite whole document with 11

index.html

```
<html>
{
  <script src="write.js">
  </script>
  <html>
```

write.js

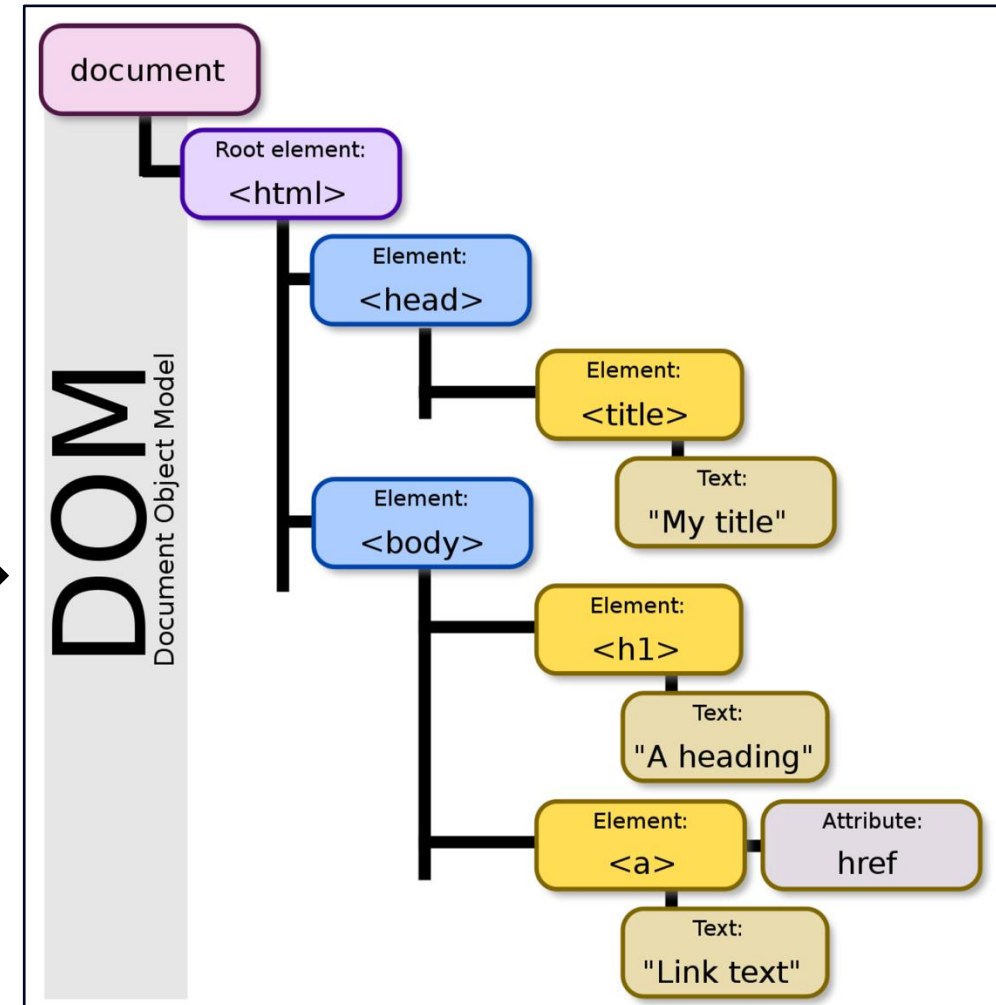
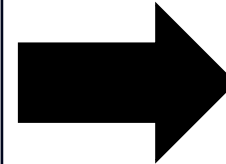
```
document.write(5 + 6)
```

External script
with src attribute

Document Object Model (DOM)

- An HTML document: structured data

```
<html>
  <head>
    <title>
      My title
    </title>
  </head>
  <body>
    <h1>A heading</h1>
    <a href="cse480.com">Link text</a>
  </body>
</body>
```



DOM and JS APIs



- Exposed to JavaScript through global objects
 - document: Access to the document (e.g., cookies, head/body)
 - navigator: Information about the browser (e.g., UA, plugins)
 - screen: Information about the screen (e.g., dimension, color depth)
 - location: Access to the URL (read and modify)
 - history: Navigation

Changing HTML DOM using JS



- JavaScript can change all the HTML DOM components in the page!
- using several APIs
 - `createElement(elementName)`
 - `createTextNode(text)`
 - `appendChild(newChild)`
 - `removeChild(node)`

Changing HTML DOM using JS (Example)⁸⁶



```
<html>
  <body>
    <ul id="t1">
      <li>Item 1</li>
    </ul>
  ...
  </body>
</html>
```

- Item 1

Changing HTML DOM using JS (Example)⁸⁷

- Item 1

```
<html>
  <body>
    <ul id="t1">
      <li>Item 1</li>
    </ul>
```

...

```
</body>
</html>
```

```
<script>
  var list = document.getElementById('t1')
  var newitem = document.createElement('li')
  var newtext = document.createTextNode('Item 2')
  list.appendChild(newitem)
  newitem.appendChild(newtext)
</script>
```

Changing HTML DOM using JS (Example)⁸⁸

```
<html>
  <body>
    <ul id="t1">
      <li>Item 1</li>
    </ul>
```

- Item 1
- Item 2

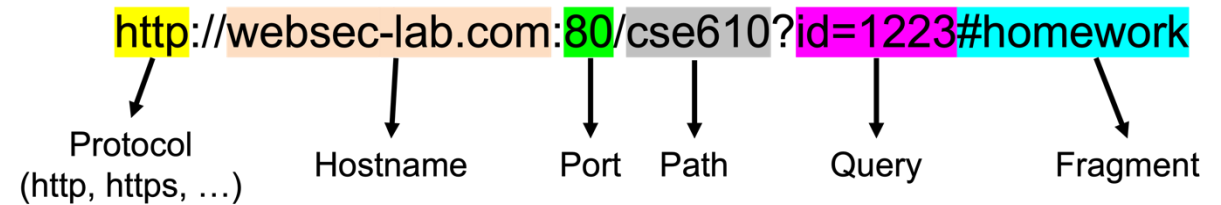
```
...
  </body>
</html>

<script>
  var list = document.getElementById('t1')
  var newitem = document.createElement('li')
  var newtext = document.createTextNode('Item 2')
  list.appendChild(newitem)
  newitem.appendChild(newtext)
</script>
```

A blue arrow originates from the JavaScript code block, specifically pointing towards the `list` variable which is assigned the value of `document.getElementById('t1')`. The arrow curves upwards and to the left, pointing directly at the `<li>Item 1</li>` line within the `<ul id="t1">` block of the HTML code above.

Accessing HTML DOM using JS (Example)⁸⁹

- `location.protocol`: protocol
- `location.hostname`: only HTTP host
- `location.port`: only the port
- `location.pathname`: path

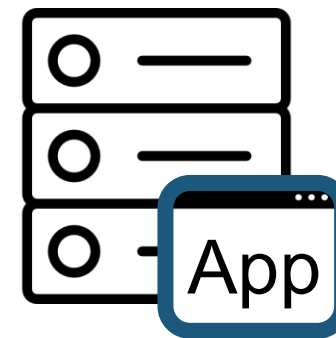


- We can display all cookies for current document by `alert(document.cookie)`



Server-side Web Application

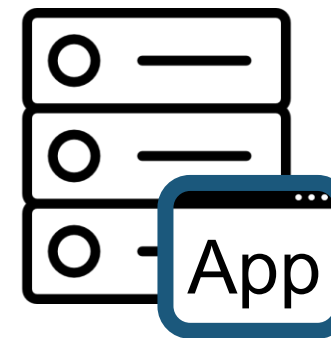
- Runs on a web server (application server)
- Can be implemented in many existing programming languages
 - PHP (Most popular!), Java, Python, Ruby on Rail, JavaScript (Node.js)



Server-side
web application

Server-side Web Application

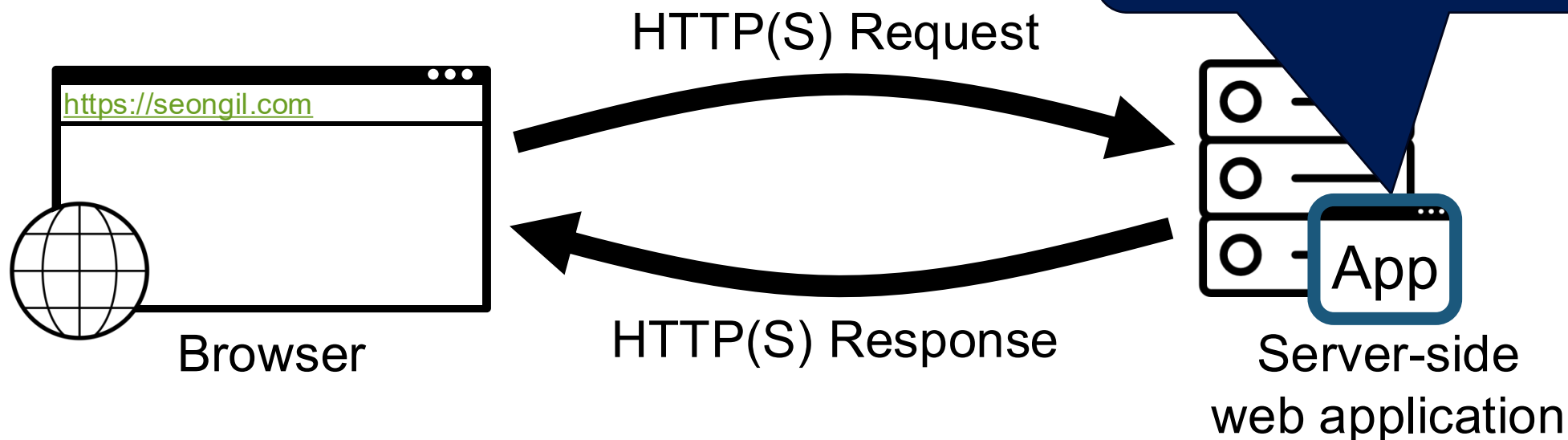
- Runs on a web server (application server)
- Can be implemented in many existing programming languages
 - PHP (Most popular!), Java, Python, Ruby on Rail, JavaScript (Node.js)
- Prepares and outputs results for users
 - Dynamically generated HTML pages
 - Content from many different sources



Server-side
web application

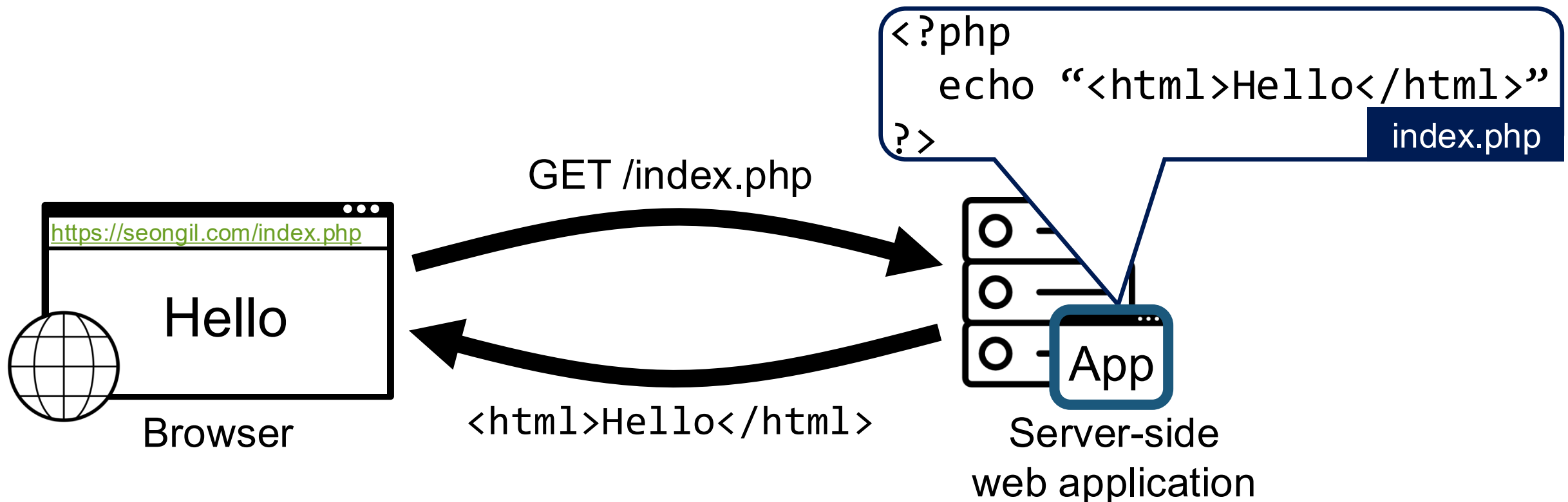
Server-side Web Application

- Runs on a web server (application server)
- Can be implemented in many existing programming languages
 - PHP (Most popular!), Java, Python, Ruby on Rail, JavaScript (Node.js)
- Prepares and outputs results for users
 - Dynamically generated HTML pages
 - Content from many different sources



Hypertext Preprocessor (PHP)

- A general-purpose scripting language that is especially suited to server-side web application development



PHP Variables and APIs: Example



Variable	Description
<code>\$_GET</code>	Handles <u>query-string parameters</u> passed through the URL
<code>\$_POST</code>	Handles <u>data submitted via POST requests</u>
<code>\$_REQUEST</code>	Handles <u>request data</u> from GET, POST, and COOKIE
<code>\$_SERVER</code>	Provides <u>server environment</u> and request information
<code>\$_SESSION</code>	Stores and accesses <u>session data</u> across multiple requests

Do not memorize, just understand!

PHP Variables and APIs: Example



API	Description
<code>echo / print</code>	<u>Outputs strings or values</u> to the HTTP response
<code>include / require</code>	<u>Executes</u> another PHP file
<code>mysqli_query()</code>	<u>Executes an SQL query</u> on a MySQL database
<code>system()</code>	<u>Executes an OS command</u> and outputs the result

Do not memorize, just understand!

Question?