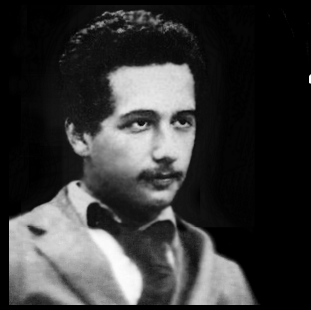
A theory which is not refutable by any conceivable event is non-scientific. Irrefutability is not a virtue of a theory (as people often think) but a vice.

Conjectures and Refutations (1963)
Karl Popper



In a letter from Albert Einstein to Karl Popper:
"[...] and I think (like you, by the way)that theory cannot be fabricated out of the results of observation, but that itcan only be invented."

From "The Logic of Scientific Discovery" (1959)
Karl Popper

Web Cybersecurity – L2

Marco Rocchetto
marco@v-research.it

Mattia Pacchin
mattia@v-research.it

V-ReseArch edu
Research & Development for Cybersecurity Engineering

https://v-research.github.io/edu/

Agenda

Intro to JavaScript & HTML [theory 30m]

Cybersecurity Topic #3 - XSS [theory 15m + lab 1h]
- XSS short intro [15m]
- WebGoat lesson (A7 – Cross-Site Scripting) [1h]

Coffee break [10m]

XSS Game (by Google: https://xss-game.appspot.com/) [1h30m]

"At Google, we know very well how important these bugs are. In fact, Google is so serious about finding and fixing XSS issues that we are paying mercenaries up to $7,500 for dangerous XSS bugs discovered in our most sensitive products."

The VERY IMPORTANT concepts so far

Slide only for
ITS@311Verona
2nd year students
in 2020

1. **L0**
    1. CIA-triad
    2. Weakness (CWE), Vulnerability (CVE), Attack (CAPEC)
    3. Hacker
2. **L1**
    1. What is a SQL injection in theory (definition)
    2. Which types of SQL injections (Boolean-based, Time-based, etc)
    3. Effects of a SQL injection (auth. Bypass, data extraction)
    4. Mitigations of a SQL injection (sanitization function, prepared statement)
    5. *How to write and use SQL injection for both authentication bypass & data extraction*

HTML

- HTML (Hyper Text Markup Language) is the standard markup language for Web pages
- HTML elements are the building blocks of HTML pages
- HTML elements are represented by <> tags

HTML example

```
<!DOCTYPE html>
<html>
    <head>
        <title>
            FLP Project
        </title>
        <script src="jquery-3.3.1.min.js"></script>
        <script type="text/javascript" src="script.js"></script>
        <script src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCuVWoUT48yzX3XVAUCkS6m7b_f0AsicC4&
        <link rel="stylesheet" type="text/css" href="style.css">
        <meta name="viewport" content="width=device-width, initial-scale=1.0">
        <link rel="icon" type="image/ico" href="favicon.ico">
    </head>
    <body>

        <div id="title" class="titleDiv">
            <h2 class="title">
                G GeoCode and Weather Forecast by FLP
            </h2>
        </div>

        <div id="buttonDiv" class="div">
            <input id="button" class="button" type="button" value="Start APP" onclick="getLocation()" />
        </div>
```

<head> - here we call necessary external files and libraries

<body> - here we define all the contents of an HTML document

HTML example

```
        <div id="mapSearchDiv" class="div" style="display: none">
            <input id="mapSearch" class="inputCamp" name="inputText" type="text" size="40" maxlength="
        </div>

        <div id="map" style="width: 80%; height: 700px; margin-left: auto; margin-right: auto"></div>

        <div id="widgetDiv" class="widgetDiv" style="display: none">
            <div class="weatherApp">
                <div class="left">
                    <div class="temperature"><span id="temperature"></span>&deg;C</div>
                </div>
                <div class="right">
                    <img style="width: 50px" id="icon" src="imgs/transparent.png" />
                </div>
                <div class="location">
                    <span id="location"></span>
                </div>
            </div>
        </div>
    </body>
</html>
```

id – it is unique and it specifies a unique HTML element

class – it specifies a class (a group) of HTML elements

style – it defines CSS inline properties

CSS

- CSS (Cascading Style Sheets) describes how HTML elements are to be displayed on screen, paper, or in other media
- CSS saves a lot of work. It can control the layout of multiple web pages all at once
- External stylesheets are stored in .css files
- With an external stylesheet file, you can change the look of an entire website by changing just one file!

CSS example

```css
body {
    background-image: url("wallpaper.png");
    background-size: 100%;
    background-repeat: no-repeat;
}

.title {
    font-family: verdana;
    font-size: 20px;
    color: #4885ed;
}

.text {
    font-family: verdana;
    font-size: 14px;
}

.titleDiv {
    width: 100%;
    height: 30px;
    text-align: center;
}
```

Style valid for <body> element

Style valid for many classes of elements

* #id {} - style valid for the element with that id

JavaScript

- JavaScript is the programming language of the Web
- JavaScript Can Change HTML Content, HTML Attribute Values, HTML Styles (CSS)
- There are Functions and Events
- For example, a function can be called when an event occurs, like when the user clicks a button (events)

JavaScript example

```
// Avvio geolocalizzazione
function getLocation() {
    if (navigator.geolocation) {
        navigator.geolocation.getCurrentPosition(function (loc) {
            var lat = loc.coords.latitude;
            var lng = loc.coords.longitude;
            console.log(loc);
            initMap(lat, lng);
            document.getElementById('mapSearchDiv').style.display = 'block';
            document.getElementById('widgetDiv').style.display = 'block';
            weatherWidget(lat, lng);
        });
    }
}
```

Function name

Function variable

Modifying style of an element identified by id

Calling another function

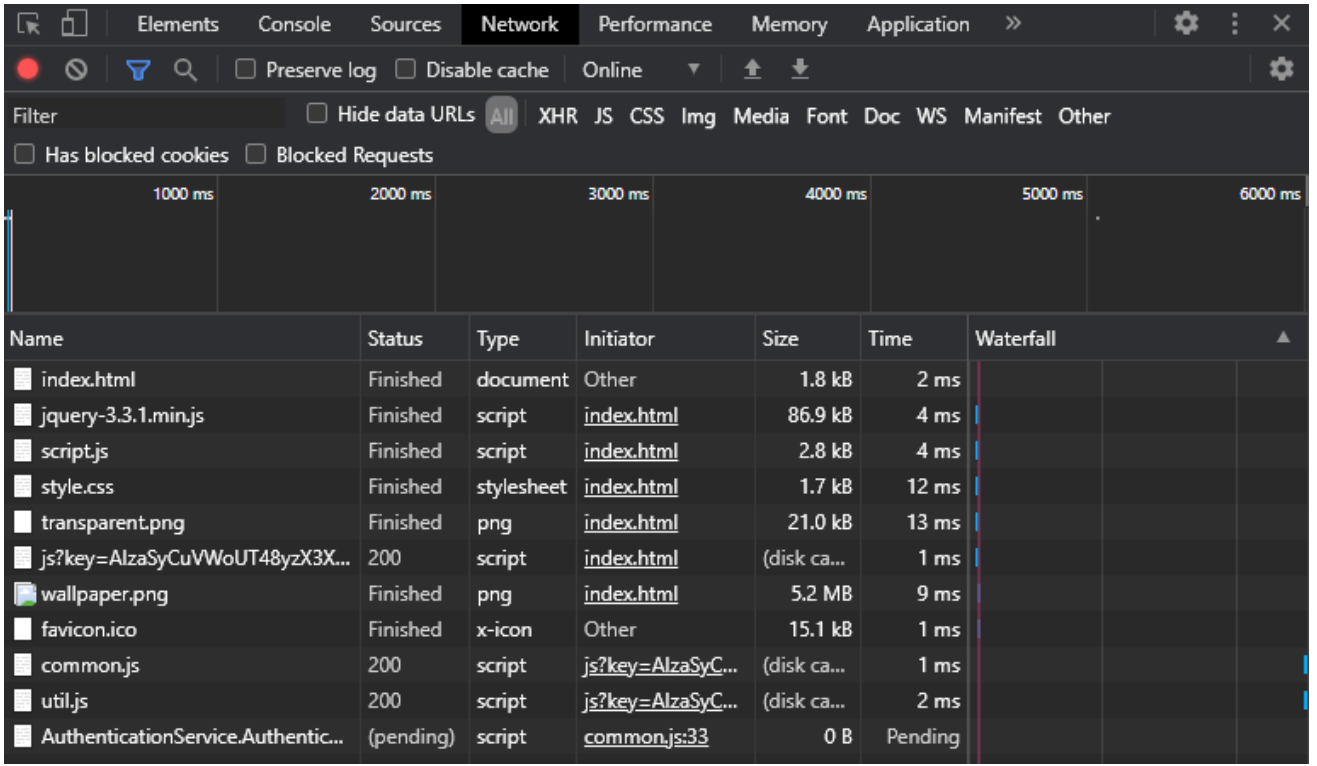
JavaScript example

```
// Richiesta dati meteo per widget
function weatherWidget(lat, lng) {
    var appId = '1b2b8bde5d366e86a2689499c5f69888';
    var temp = document.getElementById('temperature');
    var loc = document.getElementById('location');
    var icon = document.getElementById('icon');

    api_url = 'http://api.openweathermap.org/data/2.5/weather?lat=' +
            lat + '&lon=' + lng + '&units=metric&appid=' + appId + '&lang=it';
    $.ajax({
        url: api_url,
        method: 'GET',
        success: function (data) {
            icon.src = 'imgs/codes/' + data.weather[0].id + '.png';
            loc.innerHTML = data.name;
            temp.innerHTML = data.main.temp;
        }
    });
}
```

ajax call

Element Inspector

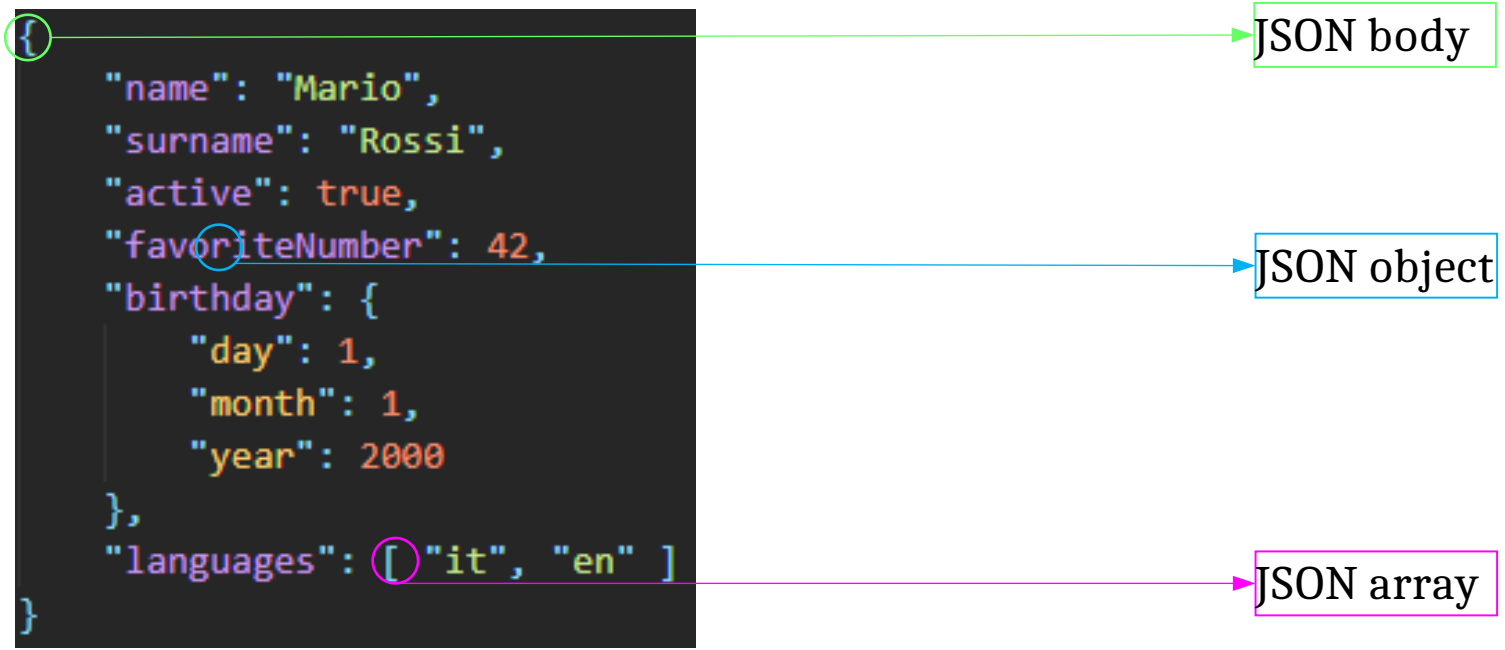


In element inspector we can find everything about our html site

JSON

- JSON (JavaScript Object Notation) is a lightweight data-interchange format
- It is easy for humans to read and write and it is easy for machines to parse and generate
- JSON is built on two structures:
    1. A collection of name/value pairs named object. Ex: {“name”:“Mario”}
    2. An ordered list of values named array. Ex: [“languages”:“en”,“it”]

JSON


{
"name": "Mario",
"surname": "Rossi",
"active": true,
"favoriteNumber": 42,
"birthday": {
"day": 1,
"month": 1,
"year": 2000
},
"languages": ["it", "en"]
}
JSON body
JSON object
JSON array

Agenda

You now also “magically” “know”

ECMAScript

Intro to JavaScript & HTML [theory 30m]

Cybersecurity Topic #3 - XSS [theory 15m + lab 1h]

- XSS short intro [15m]
- WebGoat lesson (A7 – Cross-Site Scripting) [1h]

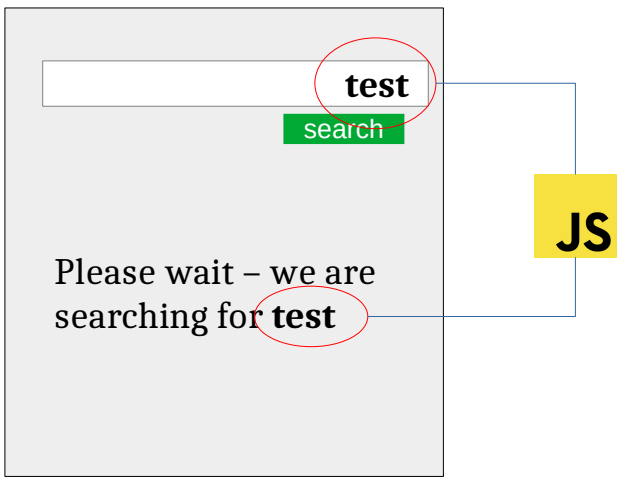
Coffee break [10m]

XSS Game (by Google: https://xss-game.appspot.com/) [1h30m]

“At Google, we know very well how important these bugs are. In fact, Google is so serious about finding and fixing XSS issues that we are paying mercenaries up to $7,500 for dangerous XSS bugs discovered in our most sensitive products.”

XSS or Cross-site Scripting

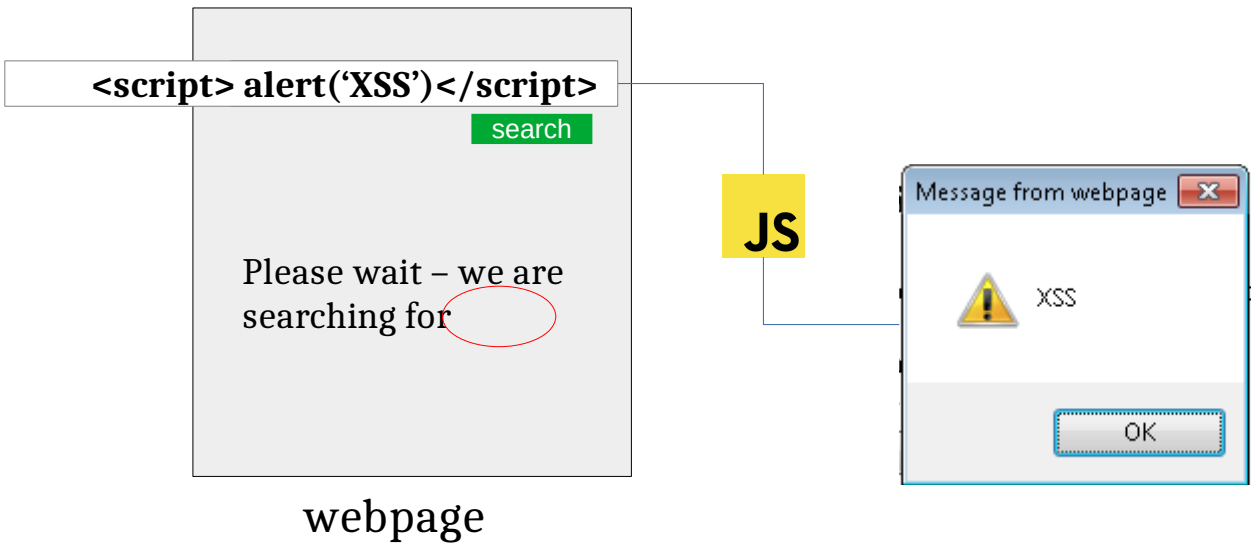
XSS vulnerability enables attackers to
inject client-side scripts
into the web pages
(by taking advantage of another user's session)



webpage

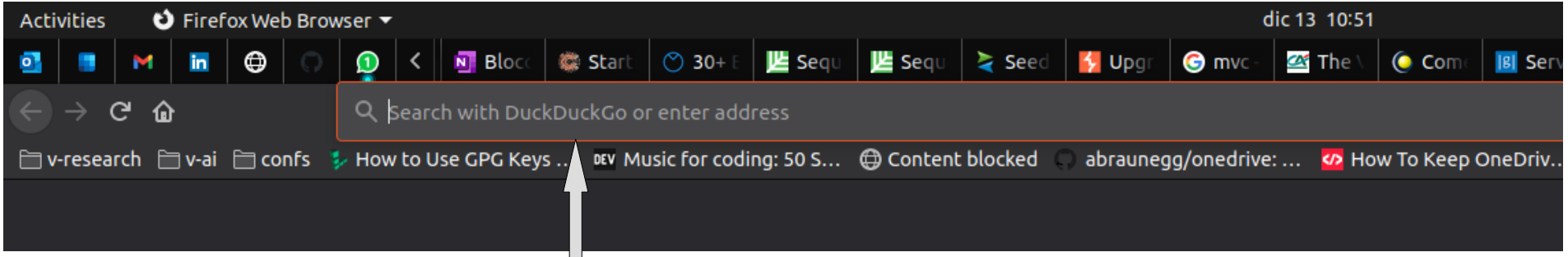
XSS or Cross-site Scripting

XSS vulnerability enables attackers to
inject client-side scripts
into the web pages
(by taking advantage of another user's session)



XSS or Cross-site Scripting

XSS vulnerability enables attackers to
inject client-side scripts
into the web pages
(by taking advantage of another user's session)

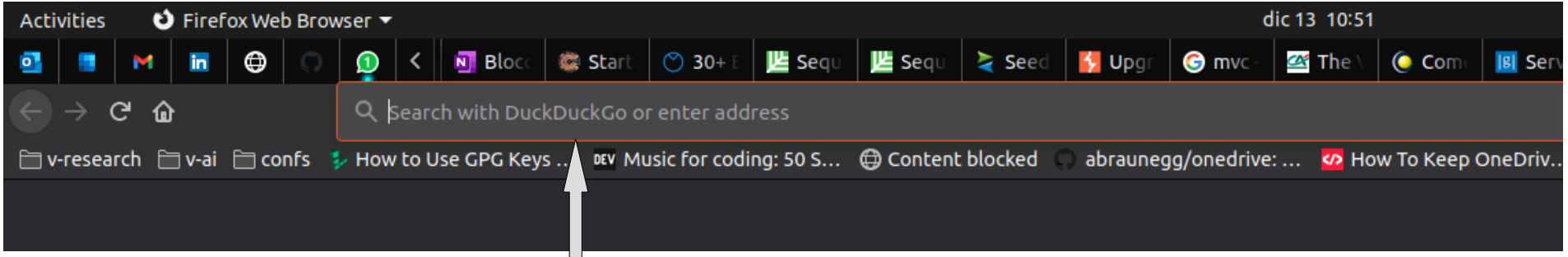


<script> alert('XSS')</script>

Can I use this in a URL?
Can I use it
https://.../[**HERE**]

XSS or Cross-site Scripting

XSS vulnerability enables attackers to
inject client-side scripts
into the web pages
(by taking advantage of another user's session)



<script> alert('XSS')</script>

Can I use this in a URL?
Can I use it
https://.../[**HERE**]

%3Cscript%3Ealert%28%22XSS%22%29%3C%2Fscript%3E

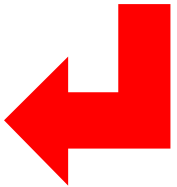
XSS Types

- **Stored XSS (persistent),** where the **malicious client-side code** that the attacker can execute is **saved by the web site and subsequently retrieved**, and executed by a user, every time a certain page is loaded.
- **Relfected XSS (non-persistent),** We refer to Reflected XSS in all the cases where the malicious **client-side code that the attacker can execute is not saved by the web site** but is "reflected" within a web page, meaning that the web site takes an user-controllable input, includes it within a web page (hence the name "reflection") and serves it to the user within the response of a request.
- **DOM-Based XSS**, the **malicious code is never sent to the web server**, meaning that there's no request begin sent and no response being received by the web browser. Both Reflected and Stored XSS requires at least one request being sent and one response being received. Such interaction is carried out by means of the HTTP protocol which involves sending and receiving HTTP requests and responses. DOM-Based XSS, on the other end, does not.
- ~~**Self-XSS** is a form of XSS vulnerability which relies on social engineering in order to trick the victim into executing malicious JavaScript code into their browser.~~

not a new type but an attack vector?

XSS Info & Mitigations

https://owasp.org/www-community/attacks/xss/



How to Avoid Cross-site scripting Vulnerabilities

- XSS (Cross Site Scripting) Prevention Cheat Sheet
- DOM based XSS Prevention Cheat Sheet
- OWASP Development Guide article on Data Validation
- OWASP Development Guide article on Phishing

How to Review Code for Cross-site scripting Vulnerabilities

See the OWASP Code Review Guide.

How to Test for Cross-site scripting Vulnerabilities

See the latest OWASP Testing Guide article on how to test for the various kinds of XSS vulnerabilities.

- Testing_for_Reflected_Cross_site_scripting
- Testing_for_Stored_Cross_site_scripting
- Testing_for_DOM-based_Cross_site_scripting

XSS Info & Mitigations

https://owasp.org/www-community/attacks/xss/

How to Avoid Cross-site scripting Vulnerabilities

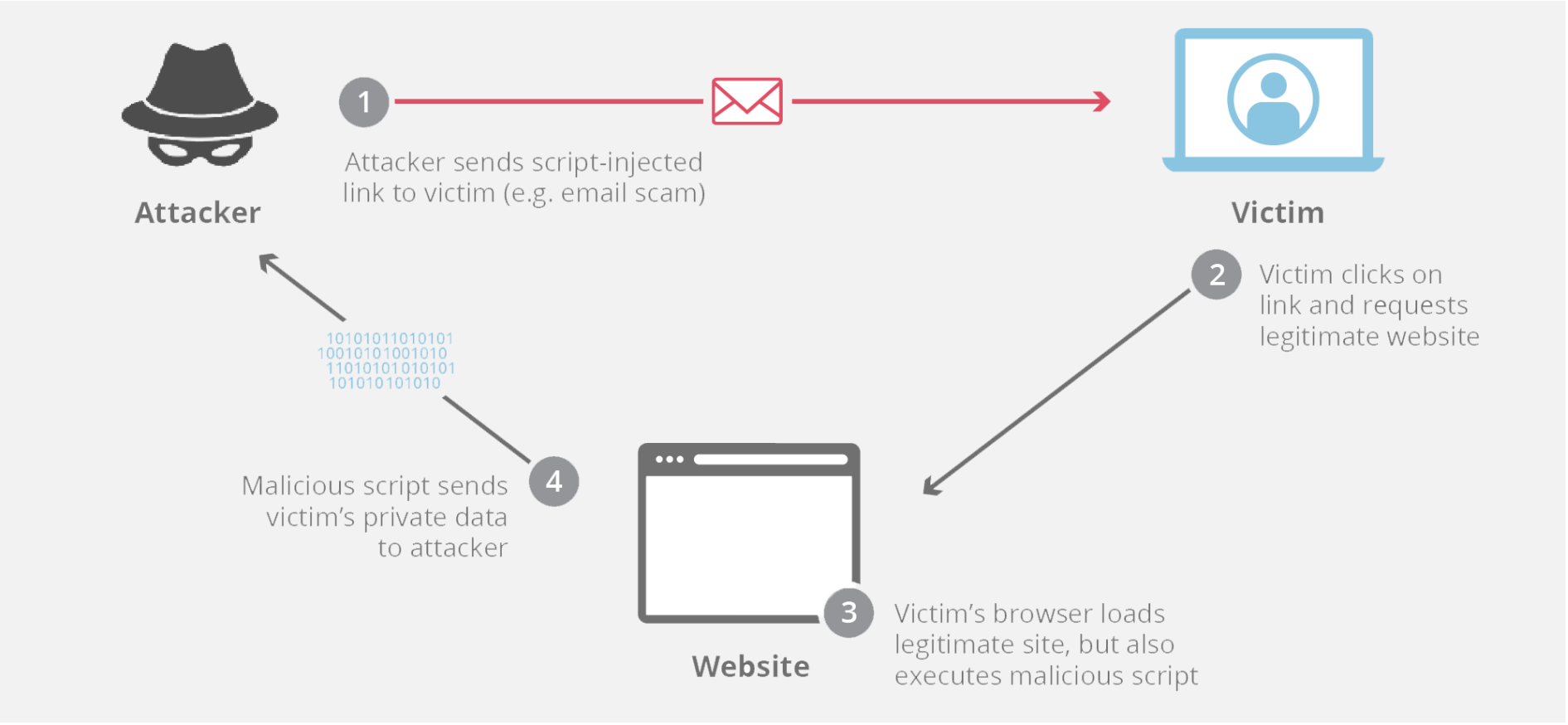
- XSS (Cross Site Scripting) Prevention Cheat Sheet
- DOM based XSS Prevention Cheat Sheet

1. Output Encoding/Escaping
2. Input (HTML) Sanitization
3. Cookies security

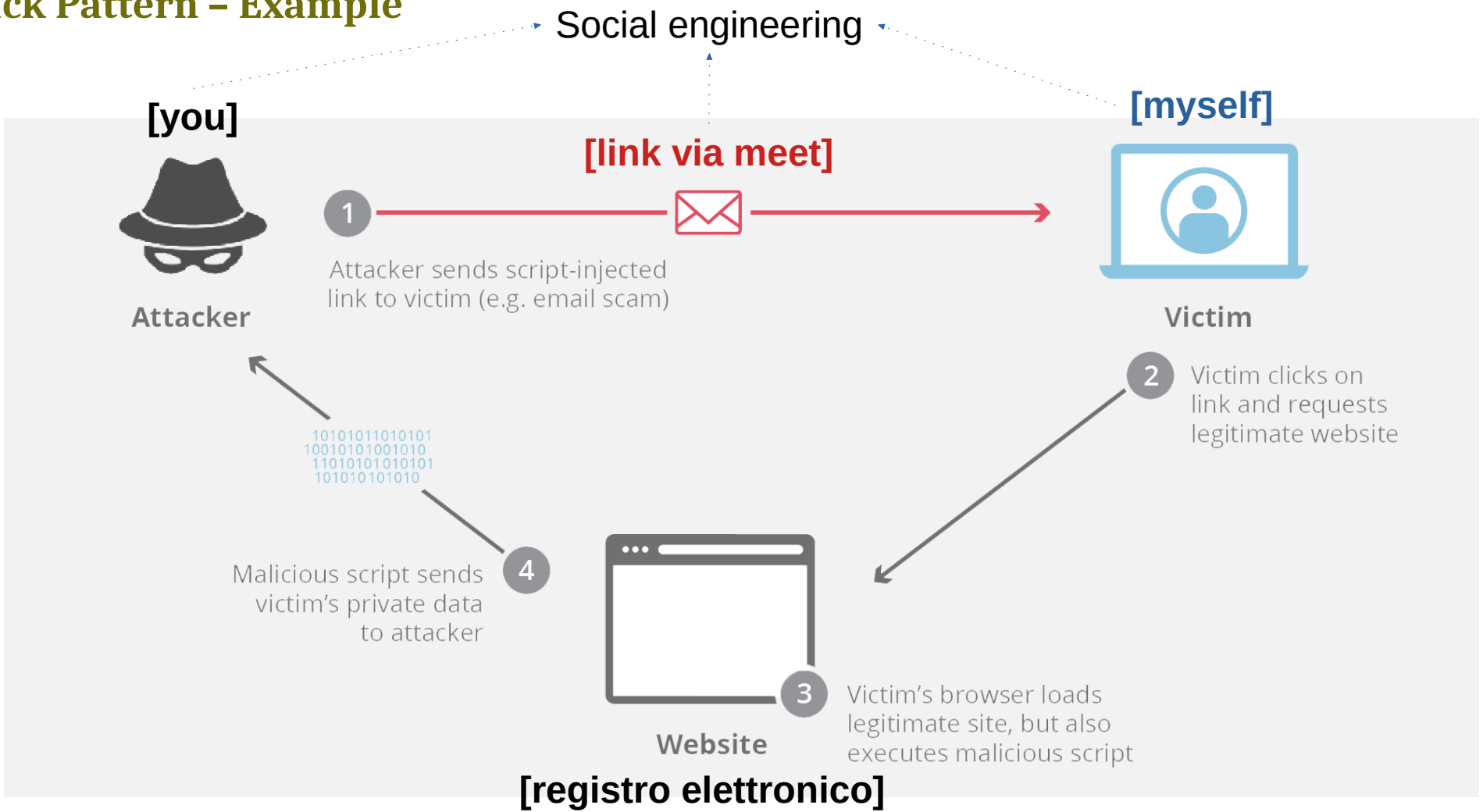
vulnerabilities.

- Testing_for_Reflected_Cross_site_scripting
- Testing_for_Stored_Cross_site_scripting
- Testing_for_DOM-based_Cross_site_scripting

Attack Pattern – Example



Attack Pattern – Example



Agenda

Intro to JavaScript & HTML [theory 30m]

Cybersecurity Topic #3 - XSS [theory 15m + lab 1h]

- XSS short intro [15m]
- WebGoat lesson (A7 – Cross-Site Scripting) [1h]

Coffee break [10m]

XSS Game (by Google: https://xss-game.appspot.com/) [1h30m]

"At Google, we know very well how important these bugs are. In fact, Google is so serious about finding and fixing XSS issues that we are paying mercenaries up to $7,500 for dangerous XSS bugs discovered in our most sensitive products."

Hacking Playground [1h30m]

1

A7 – Cross-Site Scripting
Follow WebGoat 1-9

1 2 3 4 5 6 7 8 9 10 11 12

WEBGOAT

Skip these!

https://xss-game.appspot.com/

3

Warning: You are entering the XSS game area

Google

Hacking Playground [1h30m]

https://stackoverflow.com/questions/10323392/in-javascript-jquery-what-does-e-mean

1

A7 – Cross-Site Scripting
Follow WebGoat 1-9

1 2 3 4 5 6 7 8 9 10 11 12

WEBGOAT

Skip these!

For the courious ones:
* MVC pattern
* JS e as Event var
* %2F = /

https://xss-game.appspot.com/

3

Warning: You are entering the XSS game area

Google

V-ReseΛrch edu