XSS - Cross-Site Scripting

Mitsiu Alejandro Carreño Sarabia

Intro

Cross-Site Scripting (XSS) attacks are a type of injection, in which **malicious scripts are injected** into otherwise benign and trusted websites.

XSS attacks occur when an attacker uses a web application to send malicious code, **generally in the form of a browser side script**, to a different end user.

Intro

Flaws that allow these attacks to succeed are quite widespread and occur **anywhere a web application uses input from a user within the output it generates without validating or encoding it.**

Material adicional:

https://www.youtube.com/watch?v=L5l9lSnNMxg

https://www.youtube.com/watch?v=EoaDgUgS6QA

https://www.youtube.com/watch?v=nTCDQ0UmFgE

HTML/Js

```html
<!DOCTYPE html>
<html lang="en">

<head>
  <meta charset="utf-8" />
  <title>Intro XSS</title>
</head>

<body onload="pageLoaded()">
  <div style="font-size:30px">
    <h1 id="title" style="height:12px">XSS</h1>
    <div style="display:flex; justify-content:left">
      <p>Hola: </p><p id="saludo"></p>
    </div>
    <textarea id="payload" rows="5" placeholder="Ingresa tu nombre" style="width:90%"></textarea>
    <button onclick="updateSaludo()">Click to exec!</button>
  </div>
  <script>
    function updateSaludo(){
      document.getElementById("saludo").innerHTML = document.getElementById("payload").value;
    }
    function pageLoaded(){
      let param = window.location.search
      parameterList = new URLSearchParams(param);
      document.getElementById("saludo").innerHTML = parameterList.get("name")
    }
  </script>
</body>

```

Demo 1 - XSS reflected

¿El mundo real es realmente así?

Sí, pero más complejo….

1-Legalidad, asegurense de aprender en **espacios seguros**, **o tener permiso**.

Docker!

¿Qué es docker?

¿Qué es un contenedor de docker?

¿Que es una imagen de docker?

¿Qué es Dockerfile?

¿Qué es docker-compose?

Docker!

¿Qué es docker?

Docker provides the ability to package and **run an application in a loosely isolated environment called a container.** The isolation and security lets you to **run many containers simultaneously** on a given host. Containers are lightweight and contain everything needed to run the application, so **you don't need to rely on what's installed on the host.**

Material adicional:

https://www.youtube.com/watch?v=Gjnup-PuquQ

https://www.youtube.com/watch?v=cjXI-yxqGTI

https://www.youtube.com/watch?v=eyNBf1sqdBQ

Docker!

¿Qué es un contenedor de docker?

A container is a **sandboxed process running on a host machine** that is isolated from all other processes running on that host machine. (Is a runnable instance of an image.)

¿Que es una imagen de docker?

A running container uses an isolated filesystem. This isolated filesystem is provided by an image, and the image must **contain everything needed to run an application** - all dependencies, configurations, scripts, binaries, etc. The image also contains other configurations for the container, such as environment variables, a default command to run, and other metadata.

Docker!

¿Qué es Dockerfile?

To build the image, you'll need to use a Dockerfile. A Dockerfile is simply a text-based file with no file extension that contains a script of instructions. Docker uses this script to build a container image.

Material adicional:

https://nodejs.org/en/docs/guides/nodejs-docker-webapp

https://www.youtube.com/watch?v=gAkwW2tuIqE

Docker!

Dockerfile

```
FROM node:18

# Create app directory
WORKDIR /usr/src/app

# Install app dependencies
# A wildcard is used to ensure both package.json AND package-lock.json are copied
# where available (npm@5+)
COPY package*.json ./

RUN npm install
# If you are building your code for production
# RUN npm ci --omit=dev

# Bundle app source
COPY . .

EXPOSE 9090
CMD [ "node", "index.js" ]
```

Build image

$ docker build . -t node-web-app

Run the image

$ docker run -p 49160:8080 -d node-web-app

Docker!

¿Qué es docker-compose?

Docker Compose is a tool that helps you **define and share multi-container applications.** With Compose, you can create a YAML file to **define the services and with a single command**, you can spin everything up or tear it all down.

Material adicional:

https://www.baeldung.com/ops/docker-compose

Docker!

docker-compose

```yaml
version: '2'

services:
  mariadb:
    image: mariadb
    volumes:
      - ./database:/var/lib/mysql
    environment:
      - MYSQL_ROOT_PASSWORD=moodle
      - MYSQL_ROOT_USER=root
      - MYSQL_DATABASE=moodle

  moodle:
    image: bitnami/moodle:3.10.4-debian-10-r6
    ports:
      - 8080:8080
      - 8443:8443
    environment:
      - MOODLE_DATABASE_HOST=mariadb
      - MOODLE_DATABASE_USER=root
      - MOODLE_DATABASE_PASSWORD=moodle
      - MOODLE_DATABASE_NAME=moodle
      - PUID=998
      - PGID=100
    volumes:
      - ./moodle:/bitnami/moodle
      - ./moodledata:/bitnami/moodledata
    depends_on:
      - mariadb
    links:
      - mariadb:mariadb
```

Build image

$ docker-compose build

Run the image

$ docker-compose up

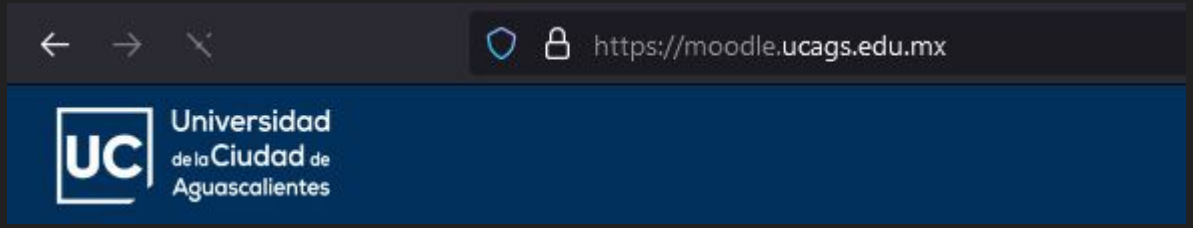
Moodle

Moodle is a free and open-source learning management system written in PHP and distributed under the GNU General Public License. Moodle is used for blended **learning, distance education, flipped classroom and other online learning** projects in schools, universities, workplaces and other sectors.

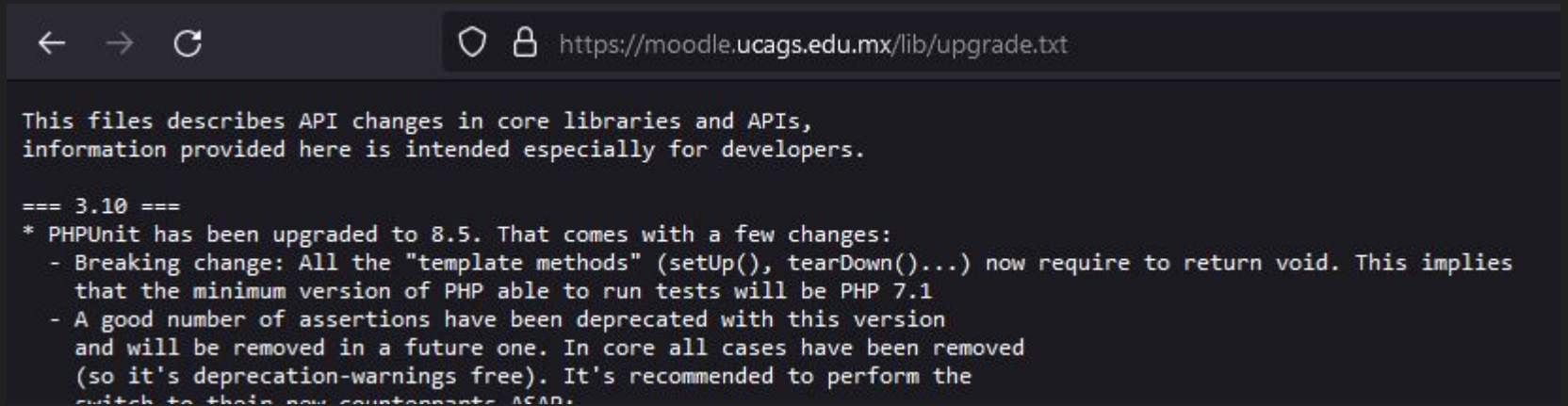
Versión 3.10

Moodle

¿Por qué moodle?



¿Por qué versión 3.10? (https://stackoverflow.com/questions/11548150/getting-moodle-version-info-no-admin-access)



Demo 2 - XSS stored

(Misma red : Compartir IP)
Usuario: <Su nombre minúsculas>
Pass: <Su nombre minúsculas>.Designa0