

## Folla 4.6.- Un pouco de seguridade en PHP

### Session fixation

1. Comproba cos exemplos anteriores *sesion1a.php* e *sesion1b.php* que o identificador de sesión é o mesmo (móstrao por pantalla empregando **`session_id()`** ).
2. Pecha o navegador, ábreo de novo e comproba que cando executas de novo eses scripts o identificador cambia (ou non cambia se tes configurado nos axustes “Abrir ventanas y pestañas anteriores”).
3. Compróbaos tamén conectándote ao servidor apache (*`docker exec -it nomeDocker /bin/bash`*), e vendo como se crea o ficheiro (lembra que era na carpeta /tmp).
4. Imos rexenerar o SID nas sesións novas, como vén nos apuntes. Para iso:
  - Engade en *sesion1b.php* un botón de “Pechar sesión”. Deberá reenviar a *pecheSesion.php* que peche a sesión e reenvíe a *sesion1a.php*.
  - En *sesion1a.php* establece unha variable que se cree a primeira vez que se crea a sesión, e rexenera o Session-ID.
  - Comproba que deste xeito consigues que cambie o identificador da sesión.

### XSS (Cross Site Scripting)

5. Fai un programa que teña un ***login.html*** (pide nome e comentario), que envíe a ***mostra.php***, no que se mostra o comentario. Comproba que se no comentario insertas código javascript (entre etiquetas `<script>` e `</script>`) este código se executa en *mostra.php*.
6. Comproba que se empregas a función ***htmlspecialchars()*** ese código non se executa.
7. Fai un programa que permite almacenar nunha base de datos un usuario, e un comentario de ata 300 caracteres. Fai un programa *mostra.php* que mostre todos os usuarios e comentarios nunha táboa. Introduce 3 usuarios e 3 comentarios.
8. Comproba se se pode almacenar código Javascript dentro do comentario, e o que acontece cando se mostra o comentario na táboa.
9. Corrixe o problema do exercicio anterior, de forma que non se almacenen as etiquetas no comentario, e o teu sitio web non poda sufrir ataques XSS