FUERZA BRUTA - LOGIN SEGURO

Cosci Franco 78644, Folli Nicolás 78531

Soluciones propuestas:

- Limit-Range
- reCAPTCHA v3

Links

- **Inseguro:** https://sds-inseguro.000webhostapp.com/ o http://localhost/sds-inseguro/

- **Limit-Range (IP)**: https://sds-medio.000webhostapp.com/ o http://localhost/sds-medio/

- **Limit-Range + reCAPTCHAv3**: https://sds-seguro.000webhostapp.com/ o http://localhost/sds-seguro/

LÓGICA LOGIN:

1. Verificar si hay un TIMEOUT vigente.
    a. IP bloqueada.
2. Verificar si Usuario/Mail existe en BD.
    a. Usuario no existe.
    b. Usuario bloqueado para esa IP.
3. Verificar contraseña.
    a. Contraseña incorrecta.
4. Inicio de sesión.

¿Cómo obtener la IP de forma correcta?

Forma incorrecta:

```php
function obtenerDireccionIP()
{
    if (!empty($_SERVER ['HTTP_CLIENT_IP'] ))
      $ip=$_SERVER ['HTTP_CLIENT_IP'];
    elseif (!empty($_SERVER ['HTTP_X_FORWARDED_FOR'] ))
      $ip=$_SERVER ['HTTP_X_FORWARDED_FOR'];
    else
      $ip=$_SERVER ['REMOTE_ADDR'];

    return $ip;
}
```

Forma utilizada:

```php
function obtenerDireccionIP()
{
    $ip = getenv('REMOTE_ADDR', true) ?: getenv('REMOTE_ADDR');
    return $ip;
}
```

- **getenv(): Obtiene el valor de una variable de entorno.**
- **Pasándole ese true al getenv se asegura de que va a obtener el valor de la variable de entorno local (y no global), establecidas por el sistema operativo o putenv.**

¿Cómo realizar el login con Bloqueo IP?

```php
$email = $_POST["email"];
$pass = $_POST["pass"];

$valor = hacerLogin($email, $pass);
switch ($valor) {
    case 0:
        header("Location: acceso_denegado.php");
        break;
    case 1:
        # Correo o contraseña incorrectos
        header("Location: index.php?mensaje=Usuario y/o contraseña incorrectos.");
        break;
    case 2:
        header("Location: index.php?mensaje=Límite de intentos alcanzado. Prueba de nuevo en 3 minutos");
        break;
    case 3:
        iniciarSesionDeUsuario();
        header("Location: usuarios.php");
        break;
}
```

Función hacerLogin($email, $pass)

```
resetearPeticionesAutomaticas(); //Si existen peticiones antiguas guardadas las restartea para evitar contar
```

```
determinarTiempo(); //Controla si se llego a los intentos maximos para una IP y si es asi determina el tiempo de
                    // kick o valida que el tiempo ya paso
```

```
$obtenerCantidadPeticionesIp = obtenerCantidadPeticionesIp();
# Compruebo que un determinado usuario no haya realizado un numero de peticiones mayores a la variable MAXIMOS_INTENTOS_IP
if ($obtenerCantidadPeticionesIp >= MAXIMOS_INTENTOS_IP)
{
    return 0;
}
```

```
else {
    if ($registro == null) {
        # No hay registros que coincidan, y no hay a quién culpar (porque el usuario no existe)
        # Guardo en la BD cada IP con la que se intenta iniciar sesion
        insertarPeticionesIp();
        return 1;
```

Función hacerLogin($email, $pass)

```
else {
    # Sí hay registros, pero no sabemos si ya ha alcanzado el límite de intentos o si la contraseña es correcta
    determinarTiempoXUsuario($registro->id);
    $conteoIntentosFallidos = obtenerConteoIntentosFallidos($registro->id);
    if ($conteoIntentosFallidos >= MAXIMOS_INTENTOS_IPxUSUARIO) {
        # Ha superado el límite por usuario, por ejemplo una determinada IP puede hacer en un determinado tiempo 10 peticiones
        # pero si intenta entrar a un determinado usuario mas de 3 veces (sin conseguirlo), ese usuario se bloquea para esa
        # determinada IP
        insertarPeticionesIp();
        return 2;
    }
```

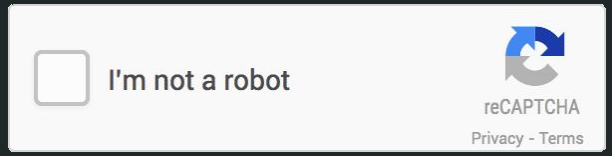
Función hacerLogin($email, $pass)

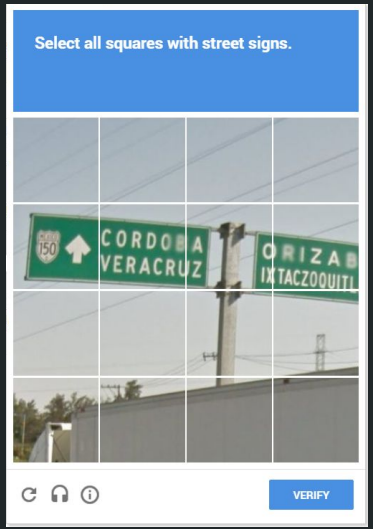
```
else {
    # Extraer la correcta de la base de datos
    $passwordHashada = $registro->pass;
    # Comparar con la proporcionada:
    $passwordCoincidiencia = password_verify($pass, $passwordHashada);
    if ($passwordCoincidiencia) {
        # Todo correcto. Borramos todos los intentos registrados en la BD de un determinado usuario con una IP.
        eliminarIntentos($registro->id);
        //eliminarPeticionesIp(); Si eliminariamos todas las peticiones generariamos una vulnerabilidad ya que
        // el atacante con tener un usuario podria reiniciar el bloqueo de su IP. Por esta razon estas se tendrian que ir
        // eliminando en un determinado tiempo o bien permitir que pueda realizar ciertas cantidad de peticiones en un tiempo
        // determinado
        return 3;
    } else {
        # Agregamos un intento fallido
        agregarIntentoFallido($registro->id);
        insertarPeticionesIp();
        return 1;
    }
}
```

reCAPTCHA

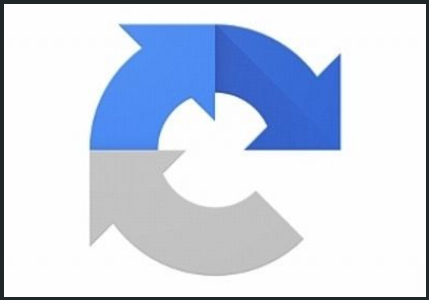
reCAPTCHA V1

reCAPTCHA V2

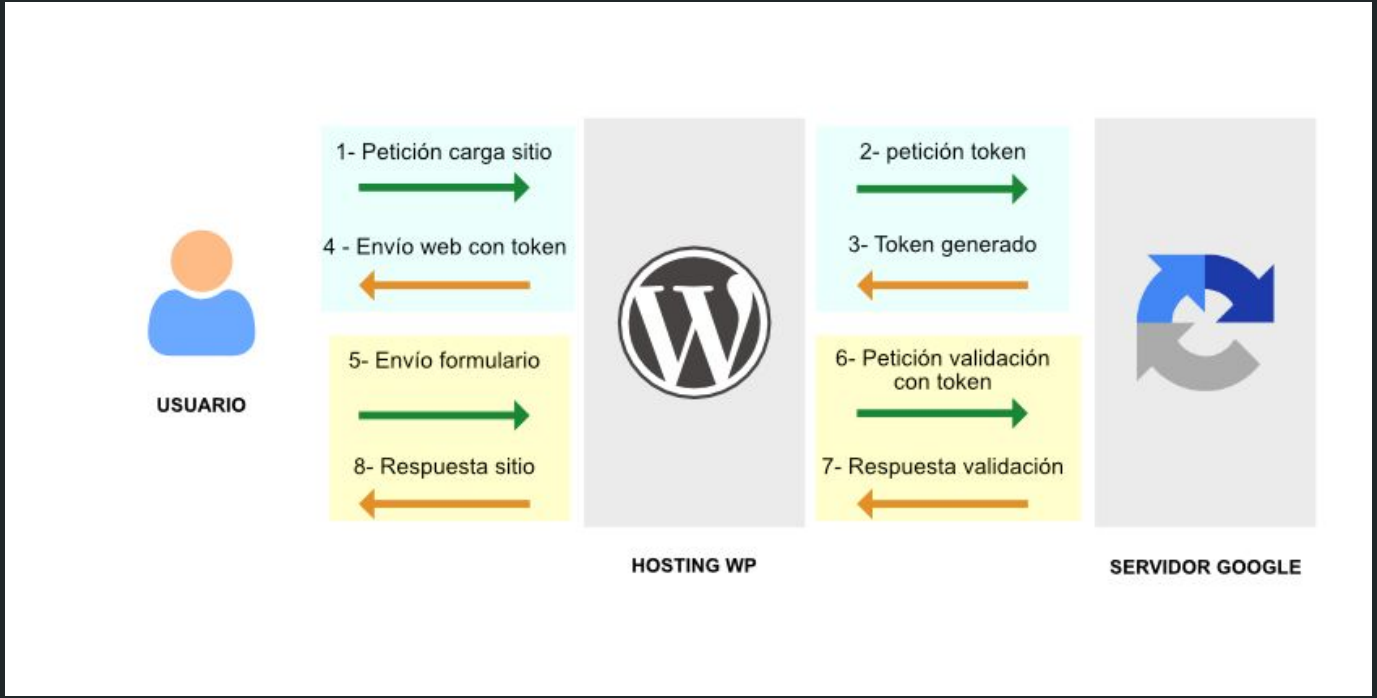




reCAPTCHA V3



Funcionamiento con reCAPTCHA V3



```php
$email = $_POST["email"];
$pass = $_POST["pass"];
$token = $_POST["token"];

$validarCaptcha = solicitudCaptcha($token);
if ($validarCaptcha) {
    $valor = hacerLogin($email, $pass);
    switch ($valor) {
        case 0:
            header("Location: acceso_denegado.php");
            break;
        case 1:
            # Correo o contraseña incorrectos
            header("Location: index.php?mensaje=Usuario y/o contraseña incorrectos.");
            break;
        case 2:
            header("Location: index.php?mensaje=Límite de intentos alcanzado. Prueba de nuevo en 3 minutos");
            break;
        case 3:
            iniciarSesionDeUsuario();
            header("Location: usuarios.php");
            break;
        }
}
else {
    header("Location: index.php?mensaje=Error al validar Captcha.");
}
```


Gracias por su atención