La sentencia `'OR 1=1--` es un ejemplo clásico de **inyección SQL**, una vulnerabilidad de seguridad en aplicaciones web que interactúan con bases de datos. Esta técnica se usa para manipular consultas SQL y obtener acceso no autorizado a la base de datos o para alterar su funcionamiento.

Vamos a desglosarla:

1. `'OR`: Este operador lógico "OR" se usa en SQL para combinar dos condiciones. En este caso, lo que viene antes del `OR` será ignorado si la condición después del `OR` es verdadera.
2. `1=1`: Es una expresión que siempre es verdadera, ya que 1 siempre es igual a 1.
3. `--`: Esto es un comentario en SQL. Cualquier cosa que venga después de `--` en la consulta SQL será ignorada por el servidor de la base de datos.

Por ejemplo, si una aplicación web espera recibir una consulta como esta para validar el usuario:

```
SELECT * FROM usuarios WHERE username = 'usuario' AND password = 'contraseña';
```

Si alguien inyecta la siguiente entrada en el campo de **contraseña**:

```
' OR 1=1--
```

La consulta resultante sería:

```
SELECT * FROM usuarios WHERE username = 'usuario' AND password = '' OR 1=1 --';
```

Lo que realmente pasa es que:

- La consulta ignora la parte que valida la contraseña porque la condición 1=1 siempre es verdadera.
- `--` hace que el resto de la consulta sea un comentario y no se procese.

Esto puede dar acceso a la base de datos sin necesidad de una contraseña válida.

Prevención

Para evitar la inyección SQL, es fundamental implementar medidas como:

- **Uso de consultas preparadas** (SQL parametrizado).
- **Validación y saneamiento** de las entradas del usuario.
- Implementación de un **WAF** (firewall de aplicaciones web) que filtre este tipo de ataques.