Paso 1: Crear base de datos y tabla vulnerable

```
CREATE DATABASE truncation_test;
USE truncation_test;

CREATE TABLE users (
  username VARCHAR(13),
  password VARCHAR(100)
);
```

Aquí, `username` está **limitado a 13 caracteres**, que es la clave para el truncamiento.

Paso 2: Insertar un usuario válido

```
INSERT INTO users (username, password) VALUES ('jacob@tornado',
'originalpass');
```

Este es el usuario real.

Paso 3: Un atacante intenta registrar un usuario similar pero más largo

```
INSERT INTO users (username, password) VALUES ('jacob@tornadoX',
'attackerpass');
```

Resultado por defecto:

- MariaDB **trunca automáticamente** `jacob@tornadoX` a `jacob@tornado`.
- Si **no hay restricción `UNIQUE`**, se permite la inserción y hay **dos usuarios distintos con el mismo `username` truncado**.
- Si hay `UNIQUE(username)`, **fallará solo si tienes modo estricto activado**, de lo contrario **lanzará una advertencia, no error.**

Paso 4: Ver los registros

```
SELECT username, password FROM users;
```

Resultado posible:

```
+-----------------+---------------+
| username        | password      |
+-----------------+---------------+
| jacob@tornado   | originalpass  |
| jacob@tornado   | attackerpass  |
+-----------------+---------------+
```

Ahora hay colisión de nombres de usuario. Si la aplicación hace un `SELECT ... LIMIT 1`, puede devolver al atacante.

Cómo evitar el ataque

Opción 1: Activar modo estricto en MariaDB

Verifica el modo SQL actual:

```
SELECT @@sql_mode;
```

Para activar modo estricto (ideal en entorno de pruebas o prod):

```
SET GLOBAL sql_mode = 'STRICT_ALL_TABLES';
```

O en tu archivo de configuración (`/etc/mysql/my.cnf` o `/etc/my.cnf`):

```
[mysqld]
sql_mode=STRICT_ALL_TABLES
```

Con modo estricto activado, al intentar esto:

```
INSERT INTO users (username, password) VALUES ('jacob@tornadoX', 'attackerpass');
```

→ **ERROR 1406 (22001)**: Data too long for column 'username' at row 1

Opción 2: Validar del lado servidor

Antes de insertar, verifica en tu código que `username.length <= 13`.

Opción 3: Agregar restricción de unicidad

```
ALTER TABLE users ADD UNIQUE (username);
```

Esto impide duplicados, pero **sin modo estricto**, aún se puede truncar silenciosamente y colisionar.

Extra: Forzar un ataque con modo no estricto

```
-- Desactiva modo estricto para pruebas
SET SESSION sql_mode = '';
```

Ahora repite:

```
INSERT INTO users (username, password) VALUES
('jacob@tornadoOVERFLOW', 'bypass');
```

→ Se insertará truncado como `jacob@tornado`.

Recomendaciones para MariaDB segura

1. **Activa modo estricto** (`STRICT_ALL_TABLES`).
2. **Usa restricciones `UNIQUE` y longitudes bien pensadas**.
3. **Valida input del lado servidor y no confíes en `maxlength` de HTML.**
4. **Monitorea advertencias de truncamiento con logs.**