

h2oai / h2o-3 Public

<> Code Issues 2.8k Pull requests 36 Discussions Actions Projects

Commit b9ae2d3



valenad1 authored on Mar 9 · ✖ 82 / 126 · Verified

[GH-16775](#) - Add couple of postgres sql parameters to DEFAULT_JDBC_DISALLOWED_PARAMETERS ([#16776](#))

- * Add postgres sql parameters. Follow-up on [GH-16622](#)
- * Add logs also as user should not need to use them

master (#16776) · jenkins-rel-latest-stable ... gha-master-7302

1 parent [d4c103d](#) commit b9ae2d3

2 files changed

+44 -2 ■■■■■■

Top

- h2o-core/src
 - main/java/water/jdbc
 - SQLManager.java
 - test/java/water/jdbc
 - SQLManagerTest.java

```

...re/src/main/java/water/jdbc/SQLManager.java
@@ -46,8 +46,11 @@ public class SQLManager {
46 46     private static final Pattern JDBC_PARAMETERS_REGEX_PATTERN = Pattern.compile("
    (?i)([a-z0-9_+]\s*=\s*");
47 47

```

```

48 48     private static final List<String> DEFAULT_JDBC_DISALLOWED_PARAMETERS =
        Stream.of(
49     -         "autoDeserialize", "queryInterceptors", "allowLoadLocalInfile",
            "allowMultiQueries", //mysql
50     -         "allowLoadLocalInfileInPath", "allowUrlInLocalInfile",
            "allowPublicKeyRetrieval", //mysql
49     +         "autoDeserialize", "queryInterceptors", "allowLoadLocalInfile",
            "allowMultiQueries", //mysql
50     +         "allowLoadLocalInfileInPath", "allowUrlInLocalInfile",
            "allowPublicKeyRetrieval", //mysql
51     +         "statementInterceptors", //mysql
52     +         "socketFactory", "socketFactoryArg", "sslfactory", "sslfactoryarg",
            //postgresql
53     +         "loggerLevel", "loggerFile", //postgresql -- not dangerous but user
            should not have a reason to use them
51 54         "init", "script", "shutdown" //h2
52 55     ).map(String::toLowerCase).collect(Collectors.toList());
53 56     private static AtomicLong NEXT_TABLE_NUM = new AtomicLong(0);

```



...rc/test/java/water/jdbc/SQLManagerTest.java



```

@@ -246,6 +246,36 @@ public void
testValidateJdbcConnectionStringMysqlMultipleEncodedString() {
246 246     SQLManager.validateJdbcUrl(jdbcConnection);
247 247 }
248 248
249 + @Test
250 + public void testValidateJdbcConnectionStringPostgresqlSocketFactory() {
251 +     exception.expect(IllegalArgumentException.class);
252 +     exception.expectMessage("Potentially dangerous JDBC parameter found:
socketFactory");
253 +
254 +     String jdbcConnection = "jdbc:postgresql://127.0.0.1:5432/test?
socketFactory=org.springframework.context.support.ClassPathXmlApplicationContex
t&socketFactoryArg=http://127.0.0.1:9090/evil.xml";
255 +
256 +     SQLManager.validateJdbcUrl(jdbcConnection);
257 + }
258 +
259 + @Test

```

```

260 + public void testValidateJdbcConnectionStringPostgresqlSslFactory() {
261 +     exception.expect(IllegalArgumentException.class);
262 +     exception.expectMessage("Potentially dangerous JDBC parameter found:
sslfactory");
263 +
264 +     String jdbcConnection = "jdbc:postgresql://127.0.0.1:5432/test?
sslfactory=org.springframework.context.support.ClassPathXmlApplicationContext&s
slfactoryarg=http://127.0.0.1:9090/evil.xml";
265 +
266 +     SQLManager.validateJdbcUrl(jdbcConnection);
267 + }
268 +
269 + @Test
270 + public void testValidateJdbcConnectionStringPostgresqlLogLevel() {
271 +     exception.expect(IllegalArgumentException.class);
272 +     exception.expectMessage("Potentially dangerous JDBC parameter found:
loggerLevel");
273 +
274 +     String jdbcConnection = "jdbc:postgresql://127.0.0.1:5432/test?
loggerLevel=DEBUG&loggerFile=/tmp/pwned.jsp";
275 +
276 +     SQLManager.validateJdbcUrl(jdbcConnection);
277 + }
278 +

```

```

249 279 /**
250 280 * Test fail if any exception is thrown therefore no assert
251 281 */

```



```

@@ -254,4 +284,13 @@ public void testValidateJdbcConnectionStringMysqlPass()
{

```

```

254 284     String jdbcConnection = "jdbc:mysql://127.0.0.1:3306/mydb?
allowedParameter=true";
255 285     SQLManager.validateJdbcUrl(jdbcConnection);
256 286 }

```

```

287 +
288 + /**
289 + * Test fail if any exception is thrown therefore no assert
290 + */
291 + @Test
292 + public void testValidateJdbcConnectionStringPostgresqlPass() {

```

```
293 +     String jdbcConnection = "jdbc:postgresql://127.0.0.1:5432/mydb?  
    ssl=true&sslmode=require";  
294 +     SQLManager.validateJdbcUrl(jdbcConnection);  
295 + }  
257 296 }
```

Comments 0



Please [sign in](#) to comment.