DBProxy 安全模块设计文档

变更说明

日期	版本	变更位置	变更说明	作者
2013-04-23	1.0.1		创建初始文档	

目录

1 目标

1. 可以动态加载配置文件（规则文件）。
2. 可以根据配置的规则阻断某条或者某类 SQL 语句。

2 概要设计

2.1 配置文件

规则配置文件采用 XML 格式。如下：

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
    <rule>
        <ruleType>      </ruleType>
        <ruleContent>   </ruleContent>
        <action>        </action>
    </rule>
</config>
```

规则配置文件名为 SecModRule.xml。在此规则文件中只可以添加某条 SQL 语句或或者某类 SQL 语句的规则。规则的添加需要通过管理员在管理员系统中运行添加规则命令，将添加的规则在写入 SecModRule.xml 中，然后直接更改一个内存中 ReloadConfigFlag 全局标记。通知安全模块规则已发生改变，需要重新加载一次。

在 SecModRule.xml 中可以添加的规则只有两种。一种是单条 SQL 语句，一种是某类 SQL 语句的模板。

说明：

1. 每个<rule> </rule>中放一条规则。
2. ruleType：规则类型分为三种，单条 SQL 语句—SingleSql，SQL 模板—SQLTemplate。关键字—Keyword。
3. ruleContent: 规则的实际内容。
4. action: 匹配到此条规则应该采用的动作，可以替换，阻断，报警等。

举例：

1. 现在想阻断 select * from bigdatatable;这条语句。

管理可以在管理系统中运行添加规则命令。通过提示输入规则：

ruleType: SigleSql

rule: select * from bigdatatale;

action: block。

规则添加命令将管理员的输入经过校验，写入 SecModRule.xml 中，同时更改全局变量 ReloadConfFlag 来通知安全模块规则配置文件发生变化，需要重新加载。这样可以实现动态加载规则。

写入规则后的配置文件变为：

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
    <rule>
        <ruleType>SingleSql</ruleType>
        <ruleContent>select * from bigdatatable</ruleContent>
        <action>block</action>
    </rule>
</config>
```

2. 如果想阻断 select * from student where id = 1;这一类语句，即：

select * from student where id = ?;
同样，管理员在管理系统中运行规则添加命令。通过提示输入：
ruleType: SqlTemplate
rule: select * from student where id = ?
action: block

规则添加命令先将输入的规则经过校验，然后写入 SecModRule.xml 中，同时更改全局变量 ReloadConfFlag 通知安全模块规则配置文件发生变化，虚重新加载。

写入 SqlTemplate 类型的规则后的配置文件：

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
    <rule>
        <ruleType>SingleSql</ruleType>
        <ruleContent>select * from bigdatatable</ruleContent>
        <action>block</action>
    </rule>
    <rule>
        <ruleType>SqlTemplate</ruleType>
        <ruleContent>select * from student where id = ?</ruleContent>
        <action>block</action>
    </rule>
</config>
```

3. 如果想探测 DDL 操作，敏感函数，则建议用正则匹配。
   这样就需要定义另外一个文本配置文件 mysql.conf。如下：

```
# CREATE section lists commands used to create tables/ind
[create]
create table
create index
create database
create procedure
create view

# DROP section lists commands used to drop tables
[drop]
drop table
drop index
drop database
drop view
truncate

# INFO section lists commands used to retrive information
# database structure and other sensitive information.
[info]
^desc
^status
describe
show databases
```

然后我们可以将动作 action 定义在 SecModConf.xml 中：

```
    <rule>
        <ruleType>Keyword</ruleType>
        <ruleContent>drop</ruleContent>
        <action>warning</action>
    </rule>
    <rule>
        <ruleType>Keyword</ruleType>
        <ruleContent>create</ruleContent>
        <action>warning</action>
    </rule>
    <rule>
        <ruleType>Keyword</ruleType>
        <ruleContent>bruteforce functions</ruleContent>
        <action>warning</action>
    </rule>
</config>
```

可以在初始的时候先将 keyword 这一类型规则都写入配置文件中，在管系统也可以通过规则添加命令去添加。

2.2 处理流程

1. 读取规则配置文件，分类建立规则。SingleSql，SqlTemplate，Keyword 三类。
   a. 建立 SingleSql 规则：
      读取 ruleType 为 SingleSql 的 ruleContent，转换成小写，去掉空格，然后经过一次 hash 得到一个 hashid。将此 hashid 和对应的 action 放入结构中存入记录 SingleSql 规则的
      列表中。
   b. 建立 singleTemple 规则：
      读取 ruleType 为 SingleSql 的 ruleContent，转换成小写，去掉空格进行 hash。得到 hashid。
      将此 hashid 和对应的 action 放入结构中存入记录 SqlTemplate 规则的列表中。
   c. Keyword 规则的建立不同于前两种规则建立，我们先需要加载 mysql.conf，生成规则后用 pcre 库提供的函数做匹配。（这个用于 greensql，做的比较好）。
      通过 xml 配置文件中的 ruleContent 与 mysql.conf 中的规则关联采取 action。
2. 三类规则的列表都建立起来以后，等待传入的 sql 语句，将传入的 sql 语句先与 SingleSql 规则匹配，如果匹配到则返回，不用匹配其他两类规则。
   没有匹配到 SingleSql 规则的话，需要将传入的 Sql 模板话，然后做一次 hash。生成的 hashid 与 singleTemplate 规则比较。同样，匹配到的话立即返回，不再继续匹配。前两类都没有匹配到的话就需要做 Keyword 规则的匹配。
3. 安全模块提供的借口打个例如：
   Action mathRule(const char* sql, int& bReloadConf);
   返回应该执行的动作。