xdon9 Update kensite_cms History

1 contributor

83 lines (67 sloc) | 2.81 KB

```
#[kensite_cms](https://github.com/seeyoui/kensite_cms):
#sql injection

The vulnerability was discovered by downloading the program's source code to local and online depl

Location:
src/main/resources/mapper/mysql/framework/mod/db/DBMapper.xml

Code:

Found that the mapper file 'name' ’oldName‘parameter is not precompiled

```
<update id="renameTable" parameterType="com.seeyoui.kensite.framework.mod.table.domain.Table">
   rename ${oldName} to ${name}
</update>
```

src/main/java/com/seeyoui/kensite/framework/mod/table/service/TableService.java call dbMapper.ren

```
public void update(Table table) throws CRUDException{
   Table tableOld = tableMapper.findOne(table.getId());
   table.preUpdate();
   tableMapper.update(table);
   if(table.getName()!=null && !table.getName().equals(tableOld.getName())) {
      table.setOldName(tableOld.getName());
      dbMapper.renameTable(table);
      tableMapper.updateFk(table);
```
```

```
      TableColumn tableColumn = new TableColumn();
      tableColumn.setOldTableName(tableOld.getName());
      tableColumn.setTableName(table.getName());
      tableColumnMapper.rename(tableColumn);
   }
   if(table.getComments()!=null && !table.getComments().equals(tableOld.getComments())) {
      dbMapper.commentTable(table);
   }
}
```

  update Interface called tableService.update(table); The table parameter contains name.

```
//@RequiresPermissions("sys:table:update")
@RequestMapping(value = "/update", method=RequestMethod.POST)
@ResponseBody
public String update(HttpSession session,
      HttpServletResponse response, HttpServletRequest request,
      ModelMap modelMap, Table table) throws Exception{
   if (!beanValidator(modelMap, table)){
      RequestResponseUtil.putResponseStr(session, response, request, modelMap, StringConstant.FALS
      return null;
   }
   tableService.update(table);
   RequestResponseUtil.putResponseStr(session, response, request, modelMap, StringConstant.TRUE);
   return null;
}
```

```
public class Table extends DataEntity<Table> {
   private static final long serialVersionUID = 5454155825314635342L;

   private String name;
   private String oldName;
   private String comments;
   private String parentTable;
   private String parentTableFk;
   private String category;
```

Harm:
The attacker only needs an ordinary user to trigger the vulnerability and use the SQL injection vu

Conditions for Execution:

Ordinary users can log in to the background and call sys/table/update to inject SQL..

```
Edition:
Version = all

Cause the cause :
Use the splicing method to splice the parameter'"+name+"' '"+oldname+"' in the sql query statement
```