<> Code   Issues   Pull requests   Actions   Projects   Security   Insights

main

xdon / kensite_cms

xdon9 Update kensite_cms   History

1 contributor

83 lines (67 sloc)   2.81 KB   •••

```
1    #[kensite_cms](https://github.com/seeyoui/kensite_cms):
2    #sql injection
3
4    The vulnerability was discovered by downloading the program's source code to local and online depl
5
6    Location:
7    src/main/resources/mapper/mysql/framework/mod/db/DBMapper.xml
8
9    Code:
10
11   Found that the mapper file 'name' 'oldName'parameter is not precompiled
12
13   ```
14   <update id="renameTable" parameterType="com.seeyoui.kensite.framework.mod.table.domain.Table">
15       rename ${oldName} to ${name}
16   </update>
17   ```
18
19   src/main/java/com/seeyoui/kensite/framework/mod/table/service/TableService.java   call dbMapper.ren
20
21   ```
22   public void update(Table table) throws CRUDException{
23       Table tableOld = tableMapper.findOne(table.getId());
24       table.preUpdate();
25       tableMapper.update(table);
26       if(table.getName()!=null && !table.getName().equals(tableOld.getName())) {
27           table.setOldName(tableOld.getName());
28           dbMapper.renameTable(table);
29           tableMapper.updateFk(table);
```

```
30        TableColumn tableColumn = new TableColumn();
31        tableColumn.setOldTableName(tableOld.getName());
32        tableColumn.setTableName(table.getName());
33        tableColumnMapper.rename(tableColumn);
34      }
35    if(table.getComments()!=null && !table.getComments().equals(tableOld.getComments())) {
36        dbMapper.commentTable(table);
37      }
38  }
```

  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 = 54541558253146353342L;

    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..
```

```
79    Edition:
80    Version = all
81
82    Cause the cause :
83    Use the splicing method to splice the parameter'"+name+"' '"+oldname+"' in the sql query statement
```