Talos Vulnerability Report

TALOS-2020-1077

OS4Ed openSIS id parameter multiple SQL injection vulnerabilities

AUGUST 31, 2020

CVE NUMBER

CVE-2020-6132, CVE-2020-6133, CVE-2020-6134

Summary

Multiple exploitable SQL injection vulnerabilities exist in the ID parameters of OS4Ed openSIS 7.3 pages. A specially crafted HTTP request can lead to SQL injection. An attacker can make an authenticated HTTP request to trigger these vulnerabilities.

Tested Versions

OS4Ed openSIS 7.3

Product URLs

https://opensis.com/

CVSSv3 Score

6.4 - CVSS:3.0/AV:N/AC:L/PR:L/UI:N/S:C/C:L/I:L/A:N

CWE

CWE-89 - Improper Neutralization of Special Elements used in an SQL Command (‘SQL Injection’)

Details

openSIS is a student information system and school management system. It is available in commercial and open-source versions. It allows schools to create schedules and track attendance, grades and transcripts.

CVE-2020-6132 - ChooseCP.php

The `id` parameter in the page `ChooseCP.php` is vulnerable to SQL injection.

Below is an example request that will trigger the vulnerability:

```
GET /opensis/ChooseCP.php?down_id=1&filename=1&name=1&table_name=courses&id=1[SQLINJECTION] HTTP/1.1
Host: [IP]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:74.0) Gecko/20100101 Firefox/74.0
Accept: */*
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
DNT: 1
Connection: close
Referer: http://[IP]/opensis/Modules.php?modname=schoolsetup/Schools.php&modfunc=update
Cookie: miniSidebar=0; PHPSESSID=6chg16qcanbg3adrqlq6sm6fa3
```

The vulnerable code for this parameter is at line 37:

```
 35 if ($_REQUEST['table_name'] != '' && $_REQUEST['table_name'] == 'courses') {
 36
 37     $sql = "SELECT COURSE_ID,c.TITLE, CONCAT_WS(' - ',c.short_name,c.title) AS GRADE_COURSE FROM courses c LEFT JOIN school_gradelevels
sg ON c.grade_level=sg.id WHERE SUBJECT_ID='$_REQUEST[id]' ORDER BY c.TI    TLE";
 38     $QI = DBQuery($sql);
 39     $courses_RET = DBGet($QI);
 40     $html = 'course_modal_cp||';
 41     $html .= '<h6>' . count($courses_RET) . ((count($courses_RET) == 1) ? ' Course was' : ' Courses were') . ' found.</h6>';
 42     $html .= '<table  class="table table-bordered"><thead><tr class="alpha-grey"><th>Course</th></tr></thead>';
 43     $html .= '<tbody>';
 44     foreach ($courses_RET as $val) {
 45
 46         $html .= '<tr><td><a href=javascript:void(0); onclick="grab_coursePeriod(' . $val['COURSE_ID'] .
',\'course_periods\',\'subject_id\')">' . $val['GRADE_COURSE'] . '</a></td></tr>';
 47     }
 48     $html .= '</tbody>';
 49     $html .= '</table>';
 50 }
 51
 52 echo $html;
 53 ?>
```

CVE-2020-6133 - CourseMoreInfo.php

The `id` parameter in the page `CourseMoreInfo.php` is vulnerable to SQL injection.

Below is an example request that will trigger the vulnerability:

```
GET /opensis/CourseMoreInfo.php?id=1[SQLINJECTION] HTTP/1.1
Host: [IP]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:74.0) Gecko/20100101 Firefox/74.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 0
Origin: http://[IP]
DNT: 1
Connection: close
Referer: http://[IP]/opensis/Modules.php?modname=eligibility/Student.php&modfunc=add&start_date=
Cookie: miniSidebar=0; PHPSESSID=6chg16qcanbg3adrqlq6sm6fa3
Upgrade-Insecure-Requests: 1
```

The vulnerable code for this parameter is at line 47:

```
 32  $sql = 'SELECT
 33                                 s.COURSE_ID,s.COURSE_PERIOD_ID,
 34                                 s.MARKING_PERIOD_ID,s.START_DATE,s.END_DATE,s.MODIFIED_DATE,s.MODIFIED_BY,
 35                                 UNIX_TIMESTAMP(s.START_DATE) AS START_EPOCH,UNIX_TIMESTAMP(s.END_DATE) AS END_EPOCH,sp.PERIOD_ID,
 36                                 cpv.PERIOD_ID,s.MARKING_PERIOD_ID as COURSE_MARKING_PERIOD_ID,cp.MARKING_PERIOD_ID as
mpa_id,cp.MP,sp.SORT_ORDER,
 37                                 c.TITLE,cp.COURSE_PERIOD_ID AS PERIOD_PULLDOWN,
 38                                 s.STUDENT_ID,r.TITLE AS ROOM,(SELECT GROUP_CONCAT(cpv.DAYS) FROM course_period_var cpv WHERE
cpv.COURSE_PERIOD_ID=cp.COURSE_PERIOD_ID) as DAYS,SCHEDULER_LOCK,CONCAT(st.LAST_NAM    E, \'' . ' ' . '\' ,st.FIRST_NAME) AS MODIFIED_NAME
 39                                 FROM courses c,course_periods cp,course_period_var cpv,rooms r,school_periods sp,schedule s
 40                                 LEFT JOIN staff st ON s.MODIFIED_BY = st.STAFF_ID
 41                                 WHERE
 42                                 s.COURSE_ID = c.COURSE_ID AND s.COURSE_ID = cp.COURSE_ID
 43                                 AND cp.COURSE_PERIOD_ID=cpv.COURSE_PERIOD_ID
 44                                  AND r.ROOM_ID=cpv.ROOM_ID
 45                                 AND s.COURSE_PERIOD_ID = cp.COURSE_PERIOD_ID
 46                                 AND s.SCHOOL_ID = sp.SCHOOL_ID AND s.SYEAR = c.SYEAR AND sp.PERIOD_ID = cpv.PERIOD_ID
 47                                 AND s.ID=' . $_REQUEST[id] . '  GROUP BY cp.COURSE_PERIOD_ID';
 48
 49         $QI = DBQuery($sql);
```

CVE-2020-6134 - MassDropModal.php

The `id` parameter in the page `MassDropModal.php` is vulnerable to SQL injection.

Below is an example request that will trigger the vulnerability:

```
GET /opensis/MassDropModal.php?table_name=course_periods&id=1[SQLINJECTION] HTTP/1.1
Host: [IP]
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:74.0) Gecko/20100101 Firefox/74.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 0
Origin: http://[IP]
DNT: 1
Connection: close
Referer: http://[IP]/opensis/Modules.php?modname=eligibility/Student.php&modfunc=add&start_date=
Cookie: miniSidebar=0; PHPSESSID=6chg16qcanbg3adrqlq6sm6fa3
Upgrade-Insecure-Requests: 1
```

The vulnerable code for this parameter is at line 35:

```
 33 if ($_REQUEST['table_name'] != '' && $_REQUEST['table_name'] == 'course_periods') {
 34
 35     $sql = "SELECT * FROM course_periods WHERE COURSE_ID='$_REQUEST[id]'AND (marking_period_id IS NOT NULL AND marking_period_id IN(" .
GetAllMP(GetMPTable(GetMP(UserMP(), 'TABLE')), UserMP()) . ") OR marking    _period_id IS NULL AND '" . date('Y-m-d') . "' <= end_date)
ORDER BY TITLE";
 36     $QI = DBQuery($sql);
 37
 38     $coursePeriods_RET = DBGet($QI);
 39     $html = 'cp_modal||';
 40     $html.='<h6>' . count($coursePeriods_RET) . ((count($coursePeriods_RET) == 1) ? ' Period was' : ' Periods were') . ' found.</h6>';
 41     if (count($coursePeriods_RET) > 0) {
 42         $html.='<table c
```

Timeline

2020-06-02 - Vendor Disclosure

2020-08-13 - Vendor provided patch to Talos for testing

2020-08-17 - Talos confirmed patch resolved issue

2020-08-31 - Public Release

CREDIT

Discovered by Yuri Kramarz of Cisco Talos.

VULNERABILITY REPORTS

PREVIOUS REPORT

TALOS-2020-1076

NEXT REPORT

TALOS-2020-1078