Talos Vulnerability Report

TALOS-2020-1076

OS4Ed openSIS course_period_id parameter multiple SQL injection vulnerabilities

AUGUST 31, 2020

CVE NUMBER

CVE-2020-6129, CVE-2020-6130, CVE-2020-6131

Summary

Multiple exploitable SQL injection vulnerabilities exist in the course_period_id parameters used in 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-2020-6129 - CpSessionSet.php

The `course_period_id` parameter in the page `CpSessionSet.php` is vulnerable to SQL injection.

Below is an example request that will trigger the vulnerability:

```
GET /opensis/CpSessionSet.php?title=1&course_period_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 37:

```
 32 if($_REQUEST['title'])
 33 {
 34     if($_REQUEST['course_period_id'])
 35     {
 36     $_SESSION['MassSchedule.php']['course_period_id']=$_REQUEST['course_period_id'];
 37     $gender_res = DBGet(DBQuery('SELECT GENDER_RESTRICTION FROM course_periods WHERE
COURSE_PERIOD_ID='.$_REQUEST['course_period_id']));
 38     $_SESSION['MassSchedule.php']['gender'] = $gender_res[1]['GENDER_RESTRICTION'];
 39 //        $_REQUEST['title'] = str_replace('"', '\"', $_REQUEST['title']);
 40         if ($gender_res[1]['GENDER_RESTRICTION'] != 'N')
 41         $_REQUEST['title']=$_REQUEST['title'].' - Gender : '.($gender_res == 'M' ? 'Male' : 'Female');
 42     }
 43     if($_REQUEST['course_id'])
 44     $_SESSION['MassSchedule.php']['course_id']=$_REQUEST['course_id'];
 45     if($_REQUEST['subject_id'])
 46     $_SESSION['MassSchedule.php']['subject_id']=$_REQUEST['subject_id'];
 47
 48     echo $_REQUEST['title'];
```

CVE-2020-6130 - MassDropSessionSet.php

The `course_period_id` parameter in the page `MassDropSessionSet.php` is vulnerable to SQL injection.

Below is an example request that will trigger the vulnerability:

```
GET /opensis/MassDropSessionSet.php?course_period_id=1[SQLINJECTION]&title=1 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 37:

```
 32 if($_REQUEST['title'])
 33 {
 34     if($_REQUEST['course_period_id'])
 35     {
 36     $_SESSION['MassDrops.php']['course_period_id']=$_REQUEST['course_period_id'];
 37     $gender_res = DBGet(DBQuery('SELECT GENDER_RESTRICTION FROM course_periods WHERE COURSE_PERIOD_ID='.$_REQUEST['course_period_id']));
 38     $_SESSION['MassDrops.php']['gender'] = $gender_res[1]['GENDER_RESTRICTION'];
 39 //        $_REQUEST['title'] = str_replace('"', '\"', $_REQUEST['title']);
 40         if ($gender_res[1]['GENDER_RESTRICTION'] != 'N')
 41         $_REQUEST['title']=$_REQUEST['title'].' - Gender : '.($gender_res == 'M' ? 'Male' : 'Female');
 42     }
 43     if($_REQUEST['course_id'])
 44     $_SESSION['MassDrops.php']['course_id']=$_REQUEST['course_id'];
 45     if($_REQUEST['subject_id'])
 46     $_SESSION['MassDrops.php']['subject_id']=$_REQUEST['subject_id'];
 47
 48     echo $_REQUEST['title'];
```

CVE-2020-6131 - MassScheduleSessionSet.php

The `coure_period_id` parameter in the page `MassScheduleSessionSet.php` is vulnerable to SQL injection.

Below is an example request that will trigger the vulnerability:

```
GET /opensis/MassScheduleSessionSet.php?course_period_id=1[SQLINJECTION]&title=1 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 lines 37 and 38:

```
 34     if($_REQUEST['course_period_id'])
 35     {
 36     $_SESSION['MassSchedule.php']['course_period_id']=$_REQUEST['course_period_id'];
 37     $gender_res = DBGet(DBQuery('SELECT GENDER_RESTRICTION FROM course_periods WHERE COURSE_PERIOD_ID='.$_REQUEST['course_period_id']));
 38     $marking_period= DBGet(DBQuery('SELECT MARKING_PERIOD_ID FROM course_periods WHERE
COURSE_PERIOD_ID='.$_REQUEST['course_period_id']));
 39     if($marking_period[1]['MARKING_PERIOD_ID']==''){
 40         $get_syear_mpid=DBGet(DBQuery('SELECT MARKING_PERIOD_ID FROM school_years WHERE SCHOOL_ID='.UserSchool().' AND
SYEAR='.UserSyear()));
 41         $marking_period=$get_syear_mpid;
 42     }
 43     $get_mp_det=DBGet(DBQuery('SELECT * FROM marking_periods WHERE MARKING_PERIOD_ID='.$marking_period[1]['MARKING_PERIOD_ID']));
 44
 45     $_SESSION['MassSchedule.php']['gender'] = $gender_res[1]['GENDER_RESTRICTION'];
 46 //        $_REQUEST['title'] = str_replace('"', '\"', $_REQUEST['title']);
 47         if ($gender_res[1]['GENDER_RESTRICTION'] != 'N')
 48         $_REQUEST['title']=$_REQUEST['title'].' - Gender : '.($gender_res == 'M' ? 'Male' : 'Female');
 49     }
 50     if($_REQUEST['course_id'])
 51     $_SESSION['MassSchedule.php']['course_id']=$_REQUEST['course_id'];
 52     if($_REQUEST['subject_id'])
 53     $_SESSION['MassSchedule.php']['subject_id']=$_REQUEST['subject_id'];
```

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.