TASK 6

YOU NEED TO FIND 15 VULNERABLE WEBSITES VULNERABILITIES:

5\* SQL INJECTION VULNERABLE (Error Based)

5\* SQL INJECTION VULNERABLE (Login Bypass)

\* 5 BROKEN ACCESS CONTROL VULNERABLE

Summary:

SQL injection is a common web application vulnerability that occurs when an attacker manipulates user-supplied data to exploit the database. By injecting malicious SQL statements into input fields, attackers can gain unauthorized access, extract sensitive data, modify database content, or even execute arbitrary commands on the server.

This vulnerability arises from improper handling of user inputs by failing to validate or parameterize them before including them into SQL queries. To avoid SQL injection, best practices are using parameterized queries or prepared statements, input validation, and limiting database privileges.

To find these we use google dorks like:

inurl:.php?id=1 site:domain.com

inurl:.php?id=1 site:com

inurl:.php?id=1 site:in

inurl:.php?id=1 site:ac.in etc..

Websites for SQL Injection (Login Based):

1. <https://www.gemportal.co.zw/login.php>

A screenshot of a computer screen

Description automatically generated with low confidence

Here we can use the SQL injection payload:

“=” “OR”

for both username and password fields.



After login:

A close-up of a message

Description automatically generated with low confidence

We have successfully logged in and we can see that an application has been submitted.

1. [http://admissions.hbs.edu.pk/login.phpA screenshot of a login screen

   Description automatically generated with medium confidence](http://admissions.hbs.edu.pk/login.php)

Here we can use the SQL injection payload:

“=” “OR”

for both username and password fields.

A picture containing text, screenshot, font, line

Description automatically generated

After login:

A screenshot of a computer

Description automatically generated with medium confidence

1. <https://career.apl.com.pk/user/login.php?apply=%27Consumer+Sales+Department%27>

A screenshot of a login screen

Description automatically generated

Here we can use the SQL injection payload:

“=” “OR”

for both username and password fields.

After login:

A red and white box with green text

Description automatically generated with low confidence

1. <https://nimls.edu.pk/login.php>

A screenshot of a login box

Description automatically generated with medium confidence

Here we can use the SQL injection payload:

“=” “OR”

for both username and password fields.

After login:

A screenshot of a computer

Description automatically generated with medium confidence

1. <https://www.favy.pk/Auth/login.php>

A screenshot of a login form

Description automatically generated with medium confidence

Here we can use the SQL injection payload:

“=” “OR”

for both username and password fields.

After login: A screenshot of a computer

Description automatically generated with medium confidence

Websites for SQL Injection (Error Based):

To fine websites with SQL injection error based vulnerability we have to add a ‘ at the end of the parameter value in the URL and then we can find a SQL server error reflecting to us in the webpage, This indicated that the page is vulnerable to SQL injection vulnerability.

1. <https://www.jisa.ac.in/gallery.php?id=1>

A picture containing text, screenshot

Description automatically generated

Now we add a ‘ at the end of the URL, we get:



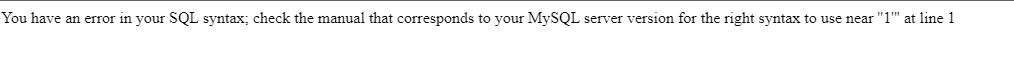
This means that this webpage is vulnerable to SQL injection.

1. <https://www.sbit.ac.in/hs_photos.php?id=1>

A screenshot of a computer

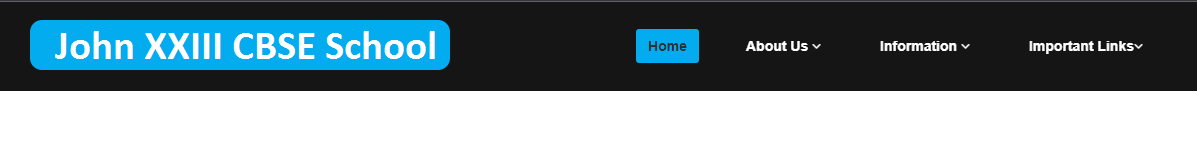
Description automatically generated with low confidence

Now we add a ‘ at the end of the URL, we get:



This means that this webpage is vulnerable to SQL injection.

1. <https://www.johnxxiii.co.in/php/gallary.php?id=1>



Now we add a ‘ at the end of the URL, we get:



This means that this webpage is vulnerable to SQL injection.

1. <https://www.gdcrazole.ac.in/events.php?id=1>

A close-up of a sign

Description automatically generated with medium confidence

Now we add a ‘ at the end of the URL, we get:



This means that this webpage is vulnerable to SQL injection.

1. <https://www.ddpsbijnor.edu.in/ddpsbijnor.php?id=1>

A screenshot of a computer

Description automatically generated with low confidence

Now we add a ‘ at the end of the URL, we get:

A screenshot of a computer

Description automatically generated with low confidence

5 BROKEN ACCESS CONTROL VULNERABLE WEBSITES:

Broken access control refers to accessing the files in a website without proper user validation and authentication.

To find websites with broken access vulnerability, we use some google dorks like:

**intitle:”index of” “admin”**

**intext:”admin login” etc..**

1. <https://www.pakuahatcollege.com/admin/uploads/faculty/>

**A screenshot of a computer

Description automatically generated with low confidence**

**This is the file structure of the given URL, and we can access images and files.**

**A screenshot of a computer

Description automatically generated**

1. <https://www.dbacap.edu.in/admin/assets/images/gallery/>

**A screenshot of a computer

Description automatically generated with medium confidence**

**This is the file structure of the given URL, and we can access images and files.**

**A close up of a flower

Description automatically generated with low confidence**

1. <https://www.cgglobal.com/admin/uploads/>

**A screenshot of a computer

Description automatically generated with medium confidence**

**This is the file structure of the given URL, and we can access images and files.**

A picture containing text, picture frame, rectangle, screenshot

Description automatically generated

1. <https://ftp.dot.state.tx.us/pub/txdot/audit-subcom/>

A screenshot of a computer

Description automatically generated with medium confidence

**This is the file structure of the given URL, and we can access images and files, both audio and video files.**

A picture containing text, screenshot, font, line

Description automatically generated

As we can see here, we have both audio and other files as well.

1. <https://csmss.org/admin/uploads/student/file/>

A screenshot of a document

Description automatically generated with low confidence

**This is the file structure of the given URL, and we can access images and files regarding student details.**

ssA close-up of a college application form

Description automatically generated with low confidence

Conclusion:

To sum up, SQL injection is a critical security flaw that seriously endangers online applications. Attackers can modify SQL queries and obtain unauthorised access to databases when user-supplied data is not adequately verified or sanitised. SQL injection can result in several harmful actions, including server compromise, data theft, and unauthorised data alteration. Developers must be aware of the dangers posed by SQL injection and take precautions such as input validation, parameterized queries, and restricted database rights. Organisations may improve the security of their online apps and shield critical data from unauthorised access by resolving this issue.